**Test plan**

**1. Introduction**

Lo scopo del documento è quello di pianificare l’attività di test del software PharmaElite.

Il testing ha lo scopo di verificare se esistono differenze tra il comportamento atteso e quello effettivo.

In questa attività si andranno a rilevare eventuali errori presenti all’interno del codice per evitare che il software abbia un comportamento inaspettato quando sarà utilizzato dall’utente finale.

Verranno analizzate le seguenti attività:

Gestore ordine,

Gestore catalogo,

Gestore utente,

Gestore carrello.

Nel documento saranno definiti i test case su cui verranno testate le funzionalità del sistema.

**2. Relationship to other documents**

Il Test Plan ha una stretta relazione con i documenti prodotti finora, dato che il sistema è stato pianificato nelle precedenti documentazioni.

2.1 Relazione con il documento di analisi dei requisiti (RAD)

La relazione riguarda i requisiti funzionali e non funzionali del sistema in quanto in fase di esecuzione di test si terrà conto delle specifiche espresse del RAD.

Il documento a cui si fa riferimento è:

"Inserire nome file "

2.2 Relazione con il System Design Document (SDD)

L’SDD contiene l’architettura del software corrente e proposto e i servizi dei sottosistemi.

Il documento a cui si fa riferimento è:

"Inserire nome file "

3. **System overview**

Come già specificato nel system design document la struttura del nostro sistema si basa su una architettura “three tier” ossia a tre livelli:

* Presentation Layer: Visualizza all’utente l’output e riceve da quest’ultimo l’input.
* Application Layer: Accetta l’input e lo converte in comando
* Data Layer: Si occupa della gestione dei dati

Le caratteristiche che saranno testate per il corretto funzionamento delle funzionalità del sistema saranno:

* Robustezza: la capacità del sistema di reagire ad eventuali input non validi inseriti dall’utente;
* Usabilità: La capacità del sistema di fornire all’utente messaggi di aiuto, in caso di errore, e di notifica, in caso di operazioni eseguite con successo.
* Sicurezza: La capacità del sistema di garantire che le operazioni possono essere effettuate solo dagli utenti con i rispettivi permessi.
* Correttezza: La capacità del sistema di garantire che le operazioni vengono eseguite cosi’ come specificate nei requisiti.

4. **Features to be tested/ not to be tested**

Di seguito saranno elencate le funzionalità del sistema che saranno sottoposte a test, suddivise

per ogni gestore del sistema.

Gestore Ordine:

Aggiungere metodo di pagamento

Aggiungere indirizzo di spedizione

Checkout

Cronologia ordini

Gestore Catalogo:

Aggiungere prodotto al catalogo.

Modificare prodotto del catalogo.

Eliminare prodotto dal catalogo.

Ricercare prodotto.

Gestore utente:

Login

Registrazione

Visualizza pagina utente

Logout

Gestore carrello:

Aggiungere prodotto al carrello

Rimuovere prodotto dal carrello

Modificare quantità prodotto nel carrello

Visualizza carrello

Procedi all'ordine

5.**Pass/Fail criteria**

Per ogni funzionalità i dati di input saranno divisi in classi di equivalenza specificate all’interno dei test case. Un test avrà successo se l’output della funzionalità testata sarà quello atteso, quest’ultimo sarà specificato all’interno del test case.

In caso di failure si procederà alla sua correzione e si effettuerà il test di regressione, rieseguendo tutti i test precedenti.

**6. Approach**

Il testing si comporrà di 3 fasi:

la prima servirà a testare le singole componenti,

poi si passerà a testare le funzionalità delle integrazioni dei vari sottosistemi,

Infine si testerà l’intero

sistema per verificare che esso soddisfi le richieste del committente.

6.1

Per il testing di unità verrà utilizzata la tecnica “Black-Box” in modo da esaminare le funzionalità dell’applicazione ed il comportamento input/output

delle singole componenti senza tener conto della loro struttura interna.

Essendo quasi impossibile

generare tutti i possibili input, verranno create classi d’equivalenza scegliendo per ognuna un test case per ridurre la ridondanza e rendere il test più efficiente.

Gli stati erronei trovati in questa fase, che comporteranno un fallimento del sistema, saranno segnalati agli implementatori al fine di correggerli e riprendere l'attività di testing al più presto.

6.2

Dopo aver sottoposto ogni componente al testing di unità, ed aver corretto gli eventuali errori trovati durante il testing, essi verranno integrati in sottosistemi più grandi che saranno sottoposti ad un test di integrazione.

"Attendere spiegazione professore"

6.3

Lo scopo di questa fase del testing è quello di dimostrare che il sistema soddisfi i requisiti richiesti dal committente e che sia pronto all’uso.

Si testeranno le funzionalità usate maggiormente e che quindi avranno maggior probabilità di fallimento.

Trattandosi di un sistema

web-based verrà utilizzato, per effettuare il testing di sistema, il tool Selenium.

Selenium si occupa di simulare l’interazione con il sistema dal punto di vista dell’utente.

7.**Suspension and Resumption**

7.1

Dopo il raggiungimento di un compromesso tra la qualità del prodotto e i costi dell’attività di testing quest’ultima sarà sospesa. Il testing sarà effettuato per un periodo di tempo tale da non ritardare la consegna e non sforare il budget imposto dal committente.

7.2

Ogniqualvolta che le componenti che genereranno errori o fallimenti, i test case verranno eseguiti nuovamente assicurandosi di aver risolto il problema e non aver introdotto nuovi malfunzionamenti.

**8.Testing materials**

L’hardware necessario per l’attività di test è una qualsiasi macchina su cui poter installare tomcat e MySQL. Il software utilizzato per il test è Junit.