**Software Testing**

Una volta completata la stesura del codice, è necessario effettuare delle prove di testing, atte a verificarne il funzionamento e la qualità, ma anche per rivelare errori/guasti/malfunzionamenti all’interno del sorgente; di conseguenza, si è trattato di test di tipo **manuale** (l’obiettivo è verificare che il software si comporti nel modo corretto, che rispetti i suoi requisiti e porti a termine i propri compiti).

Gli argomenti riguardanti i casi di test sono di natura funzionale (*black-box test*, basati su specifiche di sistema) e di natura strutturale (*white-box*, basati sul programma in sé).

**Fasi Preliminari**

Per poter funzionare è necessario generare il database e le tabelle, contenenti tutte le informazioni relative a personale medico e gli schemi per la creazione di verbali medici/pagine anagrafiche e liste operatorie. Le classi per la creazione di tabelle di Database e Tabelle sono *CreateDB* e *CreateTables*, situati in *GestioneSale\_db\_sqlite*.

In seguito, occorre generare il codice per la gestione dell’intero sistema, attraverso la classe *GenerateCode*, situata in *GestioneSale\_business\_logic*.

Una volta fatto ciò, si può avviare il login per il medico al sistema, attraverso la classe *Login* della cartella *GUI\_PortaleOspedale*.

N.B. Prima di avviare il codice, è necessario che la cartella del database sia vuota: la presenza di un database, generato in precedenza, potrebbe provocare incongruenze nel codice e impedirgli di compiere il suo lavoro; un discorso simile si può dire riguardo le tabelle per la creazione di schede.

**Funzionamento Programma**

|  |  |
| --- | --- |
| Si parte dalla pagina di Login del Sistema: il medico inserisce le sue credenziali (matricola e password) e accede al sistema. |  |
| Una volta entrato nella pagina personale, il medico può eseguire diverse operazioni, principalmente di compilazione:   * *Pagina Anagrafica* * *Lista Operatoria* * Verbale Medico |  |
| Per registrare nuovi pazienti nel sistema dell’Ospedale, il medico deve inserire tutte le informazioni richiesta nella Pagina anagrafica.  Modifiche future alla registrazione del paziente si possono apportare e la cancellazione di una pagina anagrafica elimina, di conseguenza, la lista operatoria a cui il paziente è associato |  |
| In seguito, il medico può compilare le varie liste operatorie, in base ai pazienti coinvolti e alle caratteristiche dell’intervento stesso.  Cancellando una lista operatoria si elimina, di conseguenza, il verbale medico associato |  |
| Infine, il medico può compilare il verbale medico, associato allo specifico intervento |  |

**JUNIT**

Nell’elaborazione dei casi di test abbiamo usato il framework JUNIT (ispirato al eXtreme Programming). I casi di test vengono scritti (nei relativi scenari), eseguiti (per verificare se falliscono) e corretti, per permettere di eseguirli tutti.

I principali test eseguiti riguardano:

* **Login**:
* Test per l’accesso con credenziali corrette;
* Test che verifichi che il sistema non funzioni in caso di *username* e/o *password* invalidi;
* **Ruolo Dipendente Sbagliato:**
* Test al fine di verificare che il sistema riconosca il ruolo di un dipendente dell’ospedale rispetto alla matricola (riconosce se fallisce);
* **Verbale Inesistente:**
* Test per verificare la presenza di un verbale medico rispetto al codice operazione;
* **Salvataggio Non Riuscito**
* Test per informare il sistema che il salvataggio di un nuovo verbale medico/lista operatoria/pagina anagrafica non è riuscito;