**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, i nostri casi di test sono di tipo **manuale** (l’obiettivo è verificare che il software si comporti in 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 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**

|  |  |
| --- | --- |
| Una volta avviato, il programma riporta l’utente alla pagina di Login del Sistema: il medico inserisce le sue credenziali (matricola e password) e accede alla pagina personale. |  |
| Una volta entrato nella propria pagina, il medico può eseguire diverse operazioni, principalmente di compilazione riguardanti:   * *Pagina Anagrafica Paziente* * *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.  L’eliminazione di una lista operatoria comporta, di conseguenza, la cancellazione del verbale medico associato |  |
| Infine, il medico può compilare il verbale medico, associato allo specifico intervento.  Cancellando il verbale medico associato ad una determinata operazione, non si elimina l’operazione suddetta. |  |

**JUNIT**

Nell’elaborazione dei casi di test abbiamo usato il framework JUNIT (ispirato al eXtreme Programming, XP). I casi di test vengono scritti (nei relativi scenari), eseguiti (per verificare se falliscono) e corretti, al fine di poter essere testati tutti senza errori.

I principali test eseguiti riguardano:

* **Login**:
* *testSuccessLogin*: verifica che l’accesso al sistema, con corrette credenziali, non provochi errori;
* *TestInvalidUsername*: riconosce l’acceso al sistema con username invalido;
* *TestInvalidPassword*: riconosce l’accesso al sistema con password sbagliata;
* **Ruolo Dipendente Sbagliato:**
* *RuoloDipendenteSbagliato*: inseriti una matricola dipendente corretta e un ruolo (rispetto all’ospedale), il test verifica che il ruolo dichiarato non corrisponde alla vera posizione del dipendente associato alla matricola;
* **Verbale Inesistente:**
* *VerbaleInesistente*: inserendo un codice operazione inesistente/non archiviato in sistema, il test riconosce che non esiste alcun verbale medico associato;
* **Salvataggio Operazione**
* *SalvataggioOperazioneRiuscito*: test che simula il salvataggio di un’operazione, inserendo i valori corretti, per poi metterla al confronto con i valori in sistema, rispetto al codice operazione. In caso di errore, si elimina la nuova operazione;
* *SalvataggioOperazioneFallito*: simulazione di un salvataggio d’operazione che fallisce per mancanza di dati obbligatori;
* *SalvataggioOperazioneMedicoInesistente*: verifica che il salvataggio di un’operazione è fallito, a causa di un errore nell’inserimento della matricola del medico;

N.B. Queste tipologie di test possono essere applicate anche nel caso di salvataggio di verbale medico

Per il confronto tra vettori di stringhe, abbiamo creato una funzione booleana *confrontoArray* (utilizzata principalmente nei casi di salvataggio operazione).