

Test Plan

- 1. Introduzione*
- 2. Documenti Relati*
- 3. Panoramica del Sistema*
- 4. Caratteristiche testate*

INTRODUZIONE

Lo scopo dell'attività di testing è quello di rilevare gli errori del codice prodotto durante la fase di implementazione.

L'obiettivo è evitare che questi si presentino in fase di utilizzo da parte dell'utente finale, perciò diremo che un test ha successo se ha identificato degli errori!

Lo scopo del presente documento è quello di descrivere e pianificare l'attività di testing dell'applicazione SoUL.

In esso verranno applicati test:

- Unitari su le operazioni fondamentali del server e alcune librerie del framework javascript
- Usabilità effettuati dal tester
- Usabilità effettuati da una serie di utenti.

DOCUMENTI RELATI:

Per le funzionalità da testare abbiamo fatto riferimento al RAD in particolare ai Diagrammi di caso d'uso. Abbiamo utilizzato anche l'SDD per controllare se le funzionalità sono uniformi all'architettura.

PANORAMICA DEL SISTEMA

Il seguente sistema verrà suddiviso in tre sottosistemi e in ognuno di essi i componenti saranno suddivisi in base ai vari tipi di gestione sviluppati

1.Sottosistema Model: GestioneDati

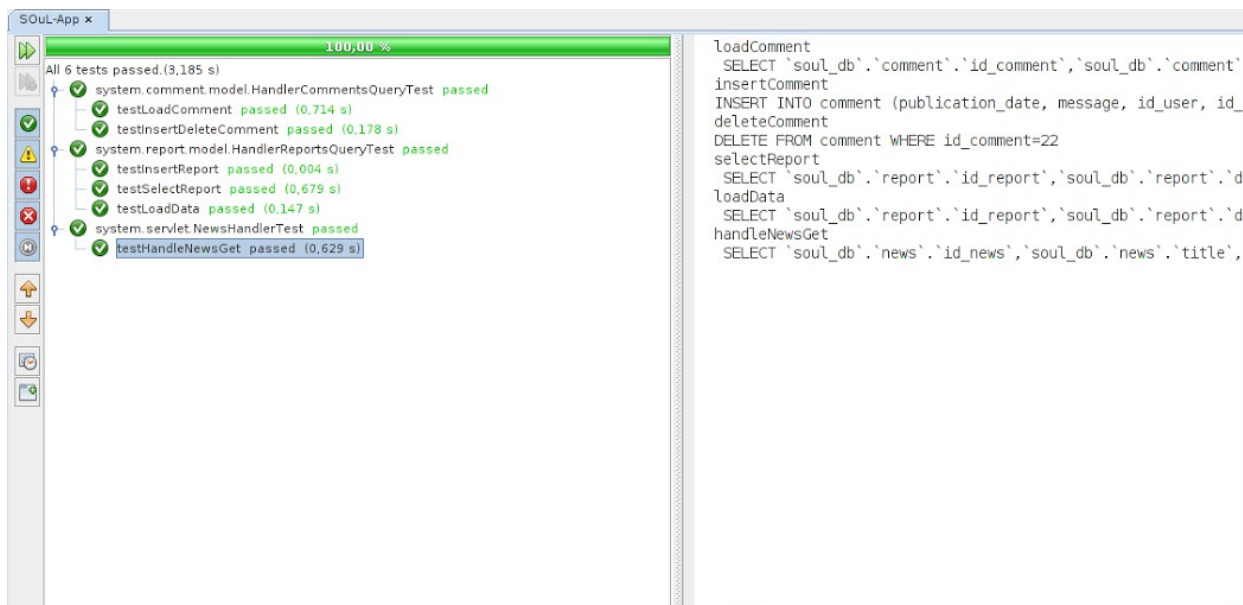
2.Sottosistema Controller: GestioneComunicazioneTraModel-View

3.Sottosistema View: GestioneInterfaccia

FUNZIONALITA TESTATE

Le funzioni che sono state testate sono(utilizzando JUnit):

- Inserimento, cancellazione di un commento
- Caricamento dei commenti
- Caricamento delle news
- Inserimento, cancellazione, selezione segnalazione



```
/**
 * Test of insertComment/deleteComment method, of class
 * HandlerCommentsQuery.
 */
@Test
public void testInsertDeleteComment() {
    System.out.println("insertComment");
    String comment = "ciao come stai?";
    String idReport = "1";
    int idUser = 1;
    Date date = new Date();

    HandlerCommentsQuery instance = new HandlerCommentsQuery();
    PreparedStatement result = instance.insertComment(comment, idReport, idUser, new java.sql.Date(date.getTime()));
    assertNotNull(result);

    try {
        ResultSet rs = result.getGeneratedKeys();
        if (rs.next()) {
            int id = rs.getInt(1);
            System.out.println("deleteComment");

            HandlerCommentsQuery i2 = new HandlerCommentsQuery();
            PreparedStatement res = i2.deleteComment(id);
            assertNotNull(res);
        }
    } catch (SQLException ex) {
        Logger.getLogger(HandlerCommentsQueryTest.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

```

public void deleteComment() {
    this.execute(
        "DELETE FROM Commento WHERE idSegnalazione=4; "
        + "UPDATE Segnalazione SET Numero_commenti=0 WHERE idSegnalazione=4;"
    );
}

```

```

/**
 * Test of loadComment method, of class HandlerCommentsQuery.
 */
@Test
public void testLoadComment() {
    System.out.println("loadComment");
    HandlerCommentsQuery handle = new HandlerCommentsQuery();

    String id = "25";
    ArrayList<RecordSet> res = new ArrayList<>(handle.loadData("`comment`.`id_report`=" + id).getRecords());
    assertTrue(!res.isEmpty());
}

```

```

/**
 * Test of handleNewsGet method, of class NewsHandler.
 */
@Test
public void testHandleNewsGet() throws Exception {
    System.out.println("handleNewsGet");
    ControllerMain.I();

    ArrayList<RecordSet> result = NewsHandler.handleNewsGet(null, null, "");
    assertTrue(!result.isEmpty());
}

```

```

public void insertCommento() {
    this.execute(
        "INSERT INTO Commento(idCommento,Data_pubblicazione,Messaggio,idUtente,idSegnalazione)"
        + " VALUES(NULL,"
        + "          NULL,"
        + "          'Cosa sta succedendo?',"
        + "          2,"
        + "          4) ;"
        + ""
        + "UPDATE `soul_db`.`Segnalazione`"
        + " SET `soul_db`.`Segnalazione`.`Numero_commenti` = `soul_db`.`Segnalazione`.`Numero_commenti`+1"
        + " WHERE `soul_db`.`Segnalazione`.`idSegnalazione`=4 ;"
    );
}

```

Test sulla libreria javascript(Utilizzando MochaJS):

passes: 23 failures: 0 duration: 0.21s

100%

Class

constructor

- ✓ should create the instance

vars

- ✓ variable should have correct values

inheritance

- ✓ should be respected the vertical inheritance

abstract constructor

- ✓ abstract class should not be instantiated

final

- ✓ final class cannot be extended
- ✓ final class cannot be dynamically changed

extend abstract

- ✓ abstract class must be extended

instantiate FullCI

- ✓ should be created successfully

private members

- ✓ private members shouldn't be visible

protected members

- ✓ protected members shouldn't be visible outside

restrictive constructors

- ✓ private constructor

destruct

- ✓ should destruct instance without errors

numeric class

- ✓ should instantiate and call methods without errors

template class

- ✓ method of other classes defined before shouldn't be unexpectedly inherited
- ✓ should instantiate and call methods without errors

styled static class

- ✓ shouldn't be instantiated but can call methods without errors

test singleton

- ✓ should create the singleton

test inner classes

- ✓ should instantiate internal classes
- ✓ should create anonymous class

test class wrapper

- ✓ should wrap a native class

test some friendly-style class

- ✓ we shouldn't have any exception in declaration, definition and instantiation

some deeper Trait tests

- ✓ should call trait methods without errors

test typeHint feature

- ✓ should correctly handle type-hinting

Test di Usabilità: Eseguiti dal tester

Test Case	Inserisco	Risposta Attesa	Risposta Ottenuta	Superato	Risolto
Registrazione	Password Diverse	Messaggio di errore	Messaggio di errore	Si	
Registrazione	Indirizzo email non conforme	Messaggio di errore	Messaggio di errore	Si	
Registrazione	Password non inserita	Messaggio di errore	Messaggio di errore	Si	
Registrazione	Nome non inserito	Messaggio di errore	Messaggio di errore	Si	
Registrazione	Cognome non inserito	Messaggio di errore	Messaggio di errore	Si	
Registrazione	Dati utente esistente	Messaggio di errore	Utente creato	No	Si
Login	Dati utente inesistenti	Messaggio di errore	Messaggio di errore	Si	
Login	Password errata	Messaggio di errore	Messaggio di errore	Si	
Modifica dati utente	Non inserisco la password	Messaggio di errore	Dati modificati	No	Si
Modifica dati utente	Modifica nome	Modifica del nome	Nome modificato	Si	
Modifica dati	Modifica	Modifica del	Cognome	Si	

utente	cognome	cognome	modificato		
Modifica dati utente	Modifica data di nascita	Modifica della data di nascita	Data modificata	Si	
Invia segnalazione(Ospite)	Descrizione ,Tipo di emergenza e immagine	Segnalazione inviata	Errore nella segnalazione	No	Si
Invia segnalazione(Utente)	Tipo di emergenza	Segnalazione inviata	Segnalazione inviata	Si	
Invia segnalazione(Utente)	Descrizione ,Tipo di emergenza e immagine	Utente: Nome Cognome	Utente: Ospite	No	Si
Invio commento	Commento vuoto	Segnalazione non inviata	Segnalazione inviata	No	Si
Invio commento	Commento	Segnalazione inviata	Segnalazione inviata	Si	
Lista segnalazioni		Mostra le segnalazioni inviate	Mostra le segnalazioni inviate	Si	
Cancellazione utente		Logout e cancellazione utente	Logout e cancellazione utente	No	Si

Test Di Usabilità: Altri test di usabilità sono stati effettuati su altri utenti e la valutazione dell'applicazione tramite l'ausilio di un questionario. Per il documento si veda l'allegato corrispondente.

In base ad una valutazione effettuata con potenziali utenti, abbiamo apportato le seguenti modifiche:

- 1) Ci è stato fatto presente che le informazioni che vogliamo fornire agli utenti sono un po' troppo verbose, quindi abbiamo deciso di ridurre il testo presente nella pagina di " segnalazione" rendendolo più conciso, mentre abbiamo lasciato le informazioni estese nel momento in cui l'utente utilizza la funzione " ricerca ".
- 2) Nonostante il sistema preveda già un localizzatore automatico, nella presentazione del progetto non abbiamo esplicitato che la posizione fosse stata recuperata automaticamente tramite i sensori del dispositivo. Ciò sarà reso possibile una volta fatta l'implementazione.
- 3) Ci è stato fatto notare che la freccia per tornare indietro è poco visibile, quindi abbiamo deciso di ingrandirla per permettere all'utente una corretta visualizzazione.
- 4) Ci è stato fatto notare anche che la barra delle icone risulta inutile nella pagina iniziale, quindi abbiamo deciso di eliminare le icone veloci dalla " home " e lasciarle solo nelle pagine secondarie.
- 5) Ci è stato fatto notare che l'iconografia attualmente è troppo scarna e che sarebbe

meglio implementare icone stilisticamente più omogenee e rendere le notifiche animate. Poiché eravamo ancora in fase di progettazione, sia le icone che le animazioni non erano ancora definitive, quindi abbiamo deciso di migliorarle nella release finale.

6) Infine, ci è stato fatto notare che le icone presenti nella barra sia superiore che inferiore erano poco visibili benché fossero utili. Per il motivo spiegato nel punto precedente, sia l'estetica che lo styling sono stati migliorati opportunamente in seguito alla fase di implementazione.

Abbiamo studiato le domande da porre ai potenziali utenti utilizzando un modulo elettronico e fornendo loro quattro diversi tipi di risposta e la possibilità, in caso di risposta negativa, di argomentarla. Per far sì che gli utenti rispondessero alle domande, abbiamo fornito loro un sito dove poter testare powerpoint online e, successivamente, compilare il modulo elettronico. In allegato, sarà fornito il modulo presentato agli utenti, il riepilogo delle risposte date con i relativi diagrammi a torta e le argomentazioni date sulle risposte negative.