



Laurea Magistrale in informatica-Università di Salerno
Corso di Ingegneria del Software 2020/2021
Prof.ssa F. Ferrucci – Prof. C. Gravino

Test Plan

Progetto

STRAGAME

Riferimento	
Versione	1.0
Data	06/01/2021
Destinatario	Prof.ssa F. Ferrucci
Presentato da	STRANIVERSITY (Umberto Mauro, Vincenzo Iovino, Mario Maffettone, Carmine Laudato)
Approvato da	



Laurea Magistrale in informatica-Università di Salerno
Corso di *Ingegneria del Software* 2020/2021
Prof.ssa F. Ferrucci – Prof. C. Gravino

Revision History

Data	Versione	Descrizione	Autori
06/01/2021	0.1	Primo, secondo e terzo Capitolo	Vincenzo Iovino Carmine Laudato
07/01/2021	0.2	Quarto, quinto e sesto capitolo	Mario Maffettone Umberto Mauro
10/01/2021	0.9	Test Cases e Test Schedule	Vincenzo Iovino Carmine Laudato Mario Maffettone Umberto Mauro
11/01/2021	1.0	Revisione Finale	Vincenzo Iovino Carmine Laudato Mario Maffettone Umberto Mauro



Sommario

1. Introduzione	4
2. Relazione con altri documenti	4
3. Panoramica del sistema	5
4. Features da testare	5
5. Features da non testare	6
6. Pass/Fail Criteria	6
7. Approccio	7
8. Sospensione e ripristino	9
9. Materiale per il testing	9
10. Test Cases	9
10.1. Gestione Account	9
10.1.1. Registrazione	9
10.1.2. Modifica Password	11
10.2. Gestione Admin	12
10.2.1. Inserimento News	12
10.2.2. Inserimento Prodotto Videoludico	13
10.3. Gestione Utente	14
10.3.1. Segnalazione Bug	14
10.3.3. Creazione Recensione	16
10.3.4. Segnalazione Pubblicazione	19
11. Testing Schedule	20



1. Introduzione

La piattaforma Stragame ha lo scopo di creare uno strumento di supporto ai videogiocatori e agli sviluppatori, che possa fornire un feedback preciso ed affidabile riguardo ai prodotti offerti.

Il sistema offrirà diverse funzioni:

- la creazione di nuove discussioni(topic);
- la creazione di recensioni dei prodotti videoludici presenti;
- l'inserimento, da parte dell'admin, di nuovi prodotti videoludici;
- la segnalazione dei bug riguardanti i titoli offerti;
- la pubblicazione di aggiornamenti e notizie da parte della casa produttrice;
- la visualizzazione delle discussioni e la possibilità di filtrarle in base ad un prodotto videoludico;
- la visualizzazione delle recensioni e la possibilità di filtrarle in base ad un prodotto videoludico;
- la visualizzazione, per l'admin, dei bug segnalati dagli utenti e la possibilità di filtrarli sia per prodotto videoludico, sia per categoria di Bug;
- la possibilità di segnalare discussioni e recensioni per vari motivi;
- la visualizzazione, per l'admin, delle discussioni e recensioni segnalate;
- la visualizzazione, per l'admin, degli utenti che hanno almeno una pubblicazione segnalata;
- la possibilità, per l'admin, di eliminare utenti e di eliminare le pubblicazioni segnalate.

Le attività di testing sono pianificate per le seguenti gestioni:

- Gestione Account;
- Gestione Utente;
- Gestione Admin.

2. Relazione con altri documenti

Per la corretta individuazione dei casi di test si fa riferimento ai seguenti documenti:

- **Relazione con il Requirements Analysis Document (RAD):** Tale relazione si fonda sul fatto che i test case verranno selezionati sulla base dei requisiti funzionali e non funzionali espressi nel documento sopracitato.



- **Relazione con il System Design Document (SDD):** Tale relazione comprende la suddivisione in sottosistemi effettuata in fase di system design e che dovrà essere preservata durante la fase di pianificazione del testing.
- **Relazione con l'Object Design Document (ODD):** Tale relazione riguarda le dipendenze tra gli oggetti individuate in fase di object design. Tali dipendenze dovranno essere considerate e saranno oggetto di test.

3. Panoramica del sistema

Il software proposto è un sito web che fa uso del pattern architetturale MVC.

Per l'implementazione della piattaforma proposta si farà uso dei seguenti linguaggi:

- Java
- HTML5
- CSS3
- Javascript

Infine, si effettuerà il deploy del sistema su Apache Tomcat.

4. Features da testare

Nell'ambito del testing di sistema si andranno a verificare le principali funzionalità per ciascuna gestione funzionale individuata.

Le funzionalità selezionate da testare sono:

- Gestione Account
 - Registrazione
 - Cambio Password
- Gestione Utente
 - Segnalazione Bug
 - Creare Topic
 - Creare Recensione
 - Segnalazione di una pubblicazione
- Gestione Admin
 - Inserimento News
 - Inserimento Prodotto Videoludico



5. Features da non testare

Le funzionalità escluse dal testing riguardano i requisiti funzionali di bassa e media priorità che, nella prima release rilasciata, non sono stati ancora implementati oppure i requisiti per cui non è necessaria la creazione di test case specifici, perché non richiedono particolari input dall'utente, per esempio la visualizzazione delle liste oppure i dettagli di una pubblicazione.

Le funzionalità da non testare sono le seguenti:

- Gestione Utente
 - Visualizza prodotti videoludici
 - Visualizza news
 - Visualizza pubblicazioni (discussioni o recensioni)
 - Ricerca Pubblicazione
- Gestione Admin
 - Rimozione Pubblicazione
 - Rimozione utente
 - Approvare una segnalazione
 - Visualizzare le pubblicazioni segnalate
 - Visualizzare gli utenti che hanno almeno una pubblicazione segnalata
- Gestione Autenticazione
 - Logout
 - Login

6. Pass/Fail Criteria

Un case test avrà successo (pass) se la sua esecuzione porterà ad un risultato uguale a quello atteso.

Viceversa, sarà considerato fallito (fail) se il risultato ottenuto sarà differente da quello atteso.

Tutto il testing sarà considerato se tutti i seguenti vincoli verranno rispettati:

- Testare tutti i requisiti da testare indicati precedentemente;
- Effettuare test di regressione ogni qualvolta verranno effettuate le modifiche alle componenti del sistema;
- Ottenere un branch coverage non inferiore al 75%.



7. Approccio

Il processo di testing si suddivide in tre parti:

- Testing di sistema
- Testing di integrazione
- Testing di unità

Ciascuna parte verrà progettata nell'ordine mostrato ma eseguito in ordine inverso.

L'individuazione dei casi di test di sistema e di integrazione avverrà prima dell'effettiva implementazione del codice; il loro affinamento avverrà soltanto dopo l'implementazione.

Differentemente, i casi di test di unità verranno progettati durante la fase di implementazione.

L'esecuzione dei casi di test avverrà durante l'implementazione del sistema stesso, al fine di avere un riscontro immediato in caso di errori commessi.

Ogni volta che verrà introdotta una nuova funzionalità, verranno effettuati i test case di unità relativi ad essa ed eventualmente anche i test di integrazione.

Se verranno effettuate delle modifiche su una componente, si rieseguiranno i test di unità e di integrazione relativi alla componente modificata e a tutte le componenti coinvolte dalle modifiche effettuate. Inoltre, si effettuerà il test di regressione, rieseguendo tutti i test case relativi alle componenti apparentemente non impattate dalla modifica.

L'approccio utilizzato per l'individuazione dei test frame sarà il category partition, quindi, al fine di minimizzare il numero di test case, gli input stessi saranno partizionati in classi di equivalenza. Per ciò che concerne la definizione del risultato atteso si userà un oracolo "umano".



Testing di sistema

Tale tipologia di testing verrà effettuata tramite Katalon Recorder, che utilizza la libreria Selenium, che si integra facilmente con quasi tutti i tipi di browser e permette di eseguire automatizzare le interazioni dell'utente con il sistema, permettendoci di effettuare i casi di test di sistema.

Testing di integrazione

In questa fase le singole unità vengono combinate e testate come gruppo. Per poter effettuare l'integration test è stata scelta la strategia bottom-up, in quanto consente di poter iniziare l'attività di testing non appena il primo modulo è stato specificato. Questo approccio richiede la costruzione di driver per simulare l'ambiente chiamante. In generale però, può portare alla problematica che i moduli possano essere codificati senza avere una chiara idea di come dovranno essere connessi ad altre parti del sistema.

La riusabilità del codice è uno dei principali benefici dell'approccio bottom-up. Nonostante questa strategia di testing di integrazione abbia alcune limitazioni, risulta essere la più semplice e naturale forma con cui eseguire questo tipo di testing.

L'esecuzione del test d'integrazione avverrà Step-by-Step, le componenti che verranno testate per prime sono quelle relative al Model, successivamente le componenti del Control e infine quelle del View.

Testing di unità

Per una corretta esecuzione del testing di unità si andranno a testare tutti i metodi definiti all'interno delle classi del sistema, eccezione fatta per le interfacce e per i metodi getter e setter auto-generati.

La definizione dei casi di test avverrà in maniera funzionale, si utilizzerà un approccio Black-Box. Similmente al testing di integrazione, si utilizzerà JUnit per la stesura dei casi di test necessari. Si definirà una classe di test relativa a ciascuna classe sorgente sottoposta a testing di unità, che conterrà i casi di test relativi ad essa.



Nel caso in cui l'esecuzione di tutti i casi di test non porterà ad una Branch Coverage del 75%, si andranno a definire nuovi casi di test utilizzando un approccio strutturale (White-Box).

8. Sospensione e ripristino

Si andranno a definire i criteri di sospensione del testing.

Inoltre, si specificano i criteri per riprendere le attività di testing in caso di sospensione.

Criteri di sospensione

La fase di testing verrà interrotta quando verranno raggiunti i risultati attesi in accordo con i tempi e i costi allocati alle attività di testing.

Criteri di ripresa

Le attività di testing riprenderanno in seguito a delle modifiche che potrebbero introdurre nuovi errori all'interno del sistema.

9. Materiale per il testing

L'hardware necessario per l'attività di testing è un pc non necessariamente avente connessione ad Internet, in quanto il sistema verrà testato in locale.

10. Test Cases

10.1. Gestione Account

10.1.1. Registrazione

Parametro: Nome
Formato: [A-Za-z] {2,25}

Categoria	Scelte
Lunghezza lu	1. Lunghezza <=1 Lunghezza >=26[errore] 2. Lunghezza = 4 [property lunghezzaLNok]
Formato fu	1. Non rispetta il formato [if lunghezzaLNok][errore] 2. Rispetta il formato [if lunghezzaLNok] [property formatoFNok]

Parametro: Cognome
Formato: [A-Za-z] {2,25}

Categoria	Scelte
Lunghezza lu	1. Lunghezza <=1 Lunghezza >=26[errore] 2. Lunghezza = 4 [property lunghezzaLNok]



Formato fu	1. Non rispetta il formato [if lunghezzaLNok][errore] 2. Rispetta il formato [if lunghezzaLNok] [property formatoFNok]
-------------------	---

Parametro: Username
Formato: [a-zA-Z0-9] {5,25}

Categoria	Scelte
Lunghezza lu	1. Lunghezza <=4 Lunghezza >=26[errore] 2. Lunghezza = 9 [property lunghezzaLUok]
Formato fu	1. Non rispetta il formato [if lunghezzaLUok][errore] 2. Rispetta il formato [if lunghezzaLUok] [property formatoFUok]

Parametro: Password
Formato: [a-zA-Z0-9] {8,46}

Categoria	Scelte
Lunghezza lu	1. Lunghezza <=7 Lunghezza >=46[errore] 2. Lunghezza = 9 [property lunghezzaLUok]
Formato fu	1. Non rispetta il formato [if lunghezzaLUok][errore] 2. Rispetta il formato [if lunghezzaLUok] [property formatoFUok]

Parametro: email
Formato: [A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}

Categorie	Scelte
Lunghezza le	1. Lunghezza <=5 [errore] 2. Lunghezza >= 6 [property lunghezzaLEok]
Formato fe	1. Non rispetta il formato [if lunghezzaLEok][errore] 2. Rispetta il formato [if lunghezzaLEok] [property formatoFEok]

Codice	Combinazione	Esito
TC_2.1.1	ln1	Errore



TC_2.1.2	ln2.fn1	Errore
TC_2.1.3	ln2.fn2.lc1	Errore
TC_2.1.4	ln2.fn2.lc2.fc1	Errore
TC_2.1.5	ln2.fn2.lc2.fc2.lu1	Errore
TC_2.1.6	ln2.fn2.lc2.fc2.lu2.fu1	Errore
TC_2.1.7	ln2.fn2.lc2.fc2.lu2.fu2.lp1	Errore
TC_2.1.8	ln2.fn2.lc2.fc2.lu2.fu2.lp2.fp1	Errore
TC_2.1.9	ln2.fn2.lc2.fc2.lu2.fu2.lp2.fp2.le1	Errore
TC_2.1.10	ln2.fn2.lc2.fc2.lu2.fu2.lp2.fp2.le2.fe1	Errore
TC_2.1.11	ln2.fn2.lc2.fc2.lu2.fu2.lp2.fp2.le2.fe2	Successo

10.1.2. Modifica Password

Parametro: oldpassword
Formato: [a-zA-Z0-9]{8,45}

Categorie	Scelte
Lunghezza lop	1. Lunghezza <=7 Lunghezza >=46[errore] 2. Lunghezza = 10 [property lunghezzaLOPok]
Formato fop	1. Non rispetta il formato [if lunghezzaLOPok][errore] 2. Rispetta il formato [if lunghezzaLOPok] [property formatoFOPok]

Parametro: newpassword
Formato: [a-zA-Z0-9]{8,45}

Categorie	Scelte
Lunghezza lnp	1. Lunghezza <=7 Lunghezza >=46[errore] 2. Lunghezza = 10 [property lunghezzaLNPok]
Formato fnp	1. Non rispetta il formato [if lunghezzaLNPok][errore] 2. Rispetta il formato [if lunghezzaLNPok] [property formatoFNPok]



Codice	Combinazione	Esito
TC_2.2.1	lop1	Errore
TC_2.2.2	lop2.fop1	Errore
TC_2.2.3	lop2.fop2.lnp1	Errore
TC_2.2.4	lop2.fop2.lnp2.fnp1	Errore
TC_2.2.5	lop2.fop2.lnp2.fnp2	Successo

10.2 Gestione Admin

10.2.1 Inserimento News

Parametro: Titolo

Formato: [a-zA-Z0-9] {5,45}

Categorie	Scelte
Lunghezza lt	1. Lunghezza <=4 Lunghezza >=46[errore] 2. Lunghezza = 9 [property lunghezzaLTok]
Formato ft	1. Non rispetta il formato [if lunghezzaLTok][errore] 2. Rispetta il formato [if lunghezzaLTok] [property formatoFTok]

Parametro: Descrizione

Formato: .{1,750}

Categorie	Scelte
Lunghezza ld	1. Lunghezza ==0 Lunghezza >=751[errore] 2. Lunghezza = 20 [property lunghezzaLDok]

Codice	Combinazione	Esito
TC_3.1.1	lt1	Errore
TC_3.1.2	lt2.ft1	Errore
TC_3.1.3	lt2.ft2.ld1	Errore
TC_3.1.4	lt2.ft2.ld2	Successo



10.2.2 Inserimento Prodotto Videoludico

Parametro: Titolo

Formato: [a-zA-Z0-9] {2,45}

Categorie	Scelte
Lunghezza lt	1. Lunghezza <=1 Lunghezza >=46[errore] 2. Lunghezza = 9 [property lunghezzaLTok]
Formato ft	1. Non rispetta il formato [if lunghezzaLTok][errore] 2. Rispetta il formato [if lunghezzaLTok] [property formatoFTok]

Parametro: Genere

Formato: [a-zA-Z0-9] {3,45}

Categorie	Scelte
Lunghezza lg	1. Lunghezza <=2 Lunghezza >=46[errore] 2. Lunghezza = 15 [property lunghezzaLGok]
Formato fg	1. Non rispetta il formato [if lunghezzaLGok][errore] 2. Rispetta il formato [if lunghezzaLGok] [property formatoFGok]

Parametro: Descrizione

Formato: .{1,1000}

Categorie	Scelte
Lunghezza ld	1. Lunghezza ==0 Lunghezza >=1001[errore] 2. Lunghezza = 15 [property lunghezzaLDok]

Codice	Combinazione	Esito
TC_3.2.1	lt1	Errore
TC_3.2.2	lt2.ft1	Errore
TC_3.2.3	lt2.ft2.lg1	Errore
TC_3.2.4	lt2.ft2.lg2.fg1	Errore
TC_3.2.5	lt2.ft2.lg2.fg2.ld1	Errore
TC_3.2.6	lt2.ft2.lg2.fg2.ld2	Successo



10.3 Gestione Utente

10.3.1 Segnalazione Bug

Parametro: Titolo

Formato: [a-zA-Z0-9] {5,45}

Categorie	Scelte
Lunghezza lt	1. Lunghezza <=4 Lunghezza >=45[errore] 2. Lunghezza = 7 [property lunghezzaLTok]
Formato ft	1. Non rispetta il formato [if lunghezzaLTok][errore] 2. Rispetta il formato [if lunghezzaLTok] [property formatoFTok]

Parametro: Categoria

Formato: [A-Za-z] {2,45}

Categorie	Scelte
Lunghezza lc	1. Lunghezza <=1 Lunghezza >=46[errore] 2. Lunghezza = 3[property lunghezzaLCok]
Formato fc	1. Non rispetta il formato [if lunghezzaLCok][errore] 2. Rispetta il formato [if lunghezzaLCok] [property formatoFCok]

Parametro: Videogioco

Formato: [a-zA-Z0-9] {2,45}

Categorie	Scelte
Lunghezza lv	1. Lunghezza <=1 Lunghezza >=46[errore] 2. Lunghezza = 3[property lunghezzaLVok]
Formato fv	1. Non rispetta il formato [if lunghezzaLVok][errore] 2. Rispetta il formato [if lunghezzaLVok] [property formatoFVok]



Parametro: Descrizione

Formato: {1,300}

Categorie	Scelte
Lunghezza Id	1. Lunghezza ==0 Lunghezza >=301 [errore] 2. Lunghezza = 15 [property lunghezzaLDok]

Codice	Combinazione	Esito
TC_4.1.1	lt1	Errore
TC_4.1.2	lt2.ft1	Errore
TC_4.1.3	lt2.ft2.lc1	Errore
TC_4.1.4	lt2.ft2.lc2.fc1	Errore
TC_4.1.5	lt2.ft2.lc2.fc2.lv1	Errore
TC_4.1.6	lt2.ft2.lc2.fc2.lv2.fv1	Errore
TC_4.1.7	lt2.ft2.lc2.fc2.lv2.fv2.ld1	Errore
TC_4.1.8	lt2.ft2.lc2.fc2.lv2.fv2.ld2	Successo

10.3.2 Creazione Topic

Parametro: Titolo

Formato: [a-zA-Z0-9] {5,45}

Categorie	Scelte
Lunghezza lt	1. Lunghezza <=4 Lunghezza >=46 [errore] 2. Lunghezza = 5 [property lunghezzaLTok]
Formato ft	1. Non rispetta il formato [if lunghezzaLTok] [errore] 2. Rispetta il formato [if lunghezzaLTok] [property formatoFTok]

Parametro: Videogioco

Formato: [a-zA-Z0-9] {2,45}

Categorie	Scelte
Lunghezza lv	1. Lunghezza <=1 Lunghezza >=46 [errore] 2. Lunghezza = 3 [property lunghezzaLVok]



Formato fv	1. Non rispetta il formato [if lunghezzaLVok][errore] 2. Rispetta il formato [if lunghezzaLVok] [property formatoFVok]
-------------------	---

Parametro: Descrizione

Formato: .{1,2000}

Categorie	Scelte
Lunghezza ld	1. Lunghezza ==0 Lunghezza >=2001 [errore] 2. Lunghezza = 30 [property lunghezzaLDok]
Formato fd	1. Non rispetta il formato [if lunghezzaLDok][errore] 2. Rispetta il formato [if lunghezzaLDok] [property formatoFDok]

Codice	Combinazione	Esito
TC_4.2.1	lt1	Errore
TC_4.2.2	lt2.ft1	Errore
TC_4.2.3	lt2.ft2.lv1	Errore
TC_4.2.4	lt2.ft2.lv2.fv1	Errore
TC_4.2.5	lt2.ft2.lv2.fv2.ld1	Errore
TC_4.2.6	lt2.ft2.lv2.fv2.ld2	Successo

10.3.3 Creazione Recensione

Parametro: Titolo

Formato: [a-zA-Z0-9]{5,45}

Categorie	Scelte
Lunghezza lt	1. Lunghezza <=4 Lunghezza >=46[errore] 2. Lunghezza = 5 [property lunghezzaLTok]
Formato ft	1. Non rispetta il formato [if lunghezzaLTok][errore] 2. Rispetta il formato [if lunghezzaLTok] [property formatoFTok]



Parametro: Videogioco
Formato: [a-zA-Z0-9]{2,45}

Categorie	Scelte
Lunghezza lv	1. Lunghezza <=1 Lunghezza >=46[errore] 2. Lunghezza = 3[property lunghezzaLVok]
Formato fv	1. Non rispetta il formato [if lunghezzaLVok][errore] 2. Rispetta il formato [if lunghezzaLVok] [property formatoFVok]

Parametro: Descrizione
Formato: .{1,2000}

Categorie	Scelte
Lunghezza ld	1. Lunghezza ==0 Lunghezza >=2001[errore] 2. Lunghezza = 30 [property lunghezzaLDok]

Parametro: Gameplay
Formato: [1-9] | 10

Categorie	Scelte
Valore Numerico vng	1. Lunghezza <=0 Lunghezza >=11[errore] 2. Lunghezza = 10 [property lunghezzaLDok]

Parametro: Grafica
Formato: [1-9] | 10

Categorie	Scelte
Valore Numerico vngr	1. Lunghezza <=0 Lunghezza >=11[errore] 2. Lunghezza = 10 [property lunghezzaLDok]

Parametro: Trama
Formato: [1-9] | 10

Categorie	Scelte
Valore Numerico vntr	1. Lunghezza <=0 Lunghezza >=11[errore] 2. Lunghezza = 10 [property lunghezzaLDok]



Parametro: VotoComplessivo

Formato: [1-9] | 10

Categorie	Scelte
Valore Numerico vnvc	1. Lunghezza <=0 Lunghezza >=11 [errore] 2. Lunghezza = 10 [property lunghezzaLDok]

Codice	Combinazione	Esito
TC_4.3.1	lt1	Errore
TC_4.3.2	lt2.ft1	Errore
TC_4.3.3	lt2.ft2.lv1	Errore
TC_4.3.4	lt2.ft2.lv2.fv1	Errore
TC_4.3.5	lt2.ft2.lv2.fv2.ld1	Errore
TC_4.3.6	lt2.ft2.lv2.fv2.ld2.vng1	Errore
TC_4.3.7	lt2.ft2.lv2.fv2.ld2.vng2.vngr1	Errore
TC_4.3.8	lt2.ft2.lv2.fv2.ld2.vng2.vngr2.vntr1	Errore
TC_4.3.9	lt2.ft2.lv2.fv2.ld2.vng2.vngr2.vntr2.vnvc1	Errore
TC_4.3.10	lt2.ft2.lv2.fv2.ld2.vng2.vngr2.vntr2.vnvc1.vnvc2	Successo



10.3.4 Segnalazione Pubblicazione

Parametro: Testo

Formato: .{1,300}

Categorie	Scelte
Lunghezza lt	1. Lunghezza ==0 Lunghezza >=301[errore] 2. Lunghezza = 20 [property lunghezzaLTok]

Parametro: Categoria

Formato: [A-Za-z] {3,45}

Categorie	Scelte
Lunghezza lc	1. Lunghezza <=2 Lunghezza >=46[errore] 2. Lunghezza = 3[property lunghezzaLCok]
Formato fc	1. Non rispetta il formato [if lunghezzaLCok][errore] 2. Rispetta il formato [if lunghezzaLCok] [property formatoFCok]

Codice	Combinazione	Esito
TC_4.4.1	lt1	Errore
TC_4.4.2	lt2.lc1	Errore
TC_4.4.3	lt2.lc2.fc1	Errore
TC_4.4.4	lt2.lc2.fc2	Successo



11. Testing Schedule

Il testing verrà effettuato parallelamente allo sviluppo del sistema, al fine di individuare e correggere errori nel momento in cui essi verranno introdotti.

L'attività di testing è prerogativa di tutti i membri del team, i quali dovranno occuparsi della generazione e dell'esecuzione dei casi di test relativi alle funzionalità che hanno implementato.

Tale attività è fondamentale nello sviluppo di un sistema software in quanto la sua mancanza può portare al completo fallimento del sistema. Data tale premessa ne risulta fondamentale la schedulazione e la successiva documentazione.