**Test 3**

Nome e cognome: Julian Sprugasci

Classe: I3AC

Tempo a disposizione: 90 minuti (a casa)

# Tema

Controllo dei link per un blog

# Indicazioni

Siete amministratori di un blog dove gli utenti pubblicano i loro articoli. Oltre ai loro testi sono presenti parecchi collegamenti a siti esterni, molti dei quali sono però invalidi.

Siccome molti commenti riguardano proprio questi collegamenti non funzionanti, avete deciso di scrivere un programma per controllare automaticamente la validità di ogni collegamento, e modificarne la descrizione se non funzionano.

Es :

<a href="https://miosito.net/pageid">Bello questo!</a>

Deve essere trasformato in

<a href="https://miosito.net/pageid" class="text-danger">[invalid] Bello questo!</a>

Mentre per le immagini cambiare il percorso di src con ".\images\invalid.png" (file già presente).

Il programma deve essere eseguito ogni notte dopo l'esecuzione dei backup giornalieri e, per prevenire gli errori, il programma viene eseguito ogni volta che un utente inserisce un collegamento. Per questo motivo il tempo di esecuzione deve essere inferiore ai 3 secondi.

Per velocizzare la scansione ad ogni inserimento di un collegamento, questo ottiene un id (salvato anche come <a id=[idgenerato] ) che viene salvato in una tabella dei collegamenti (l'utente non ha comunque la possibilità di specificare degli id personali).

L'ID del link è un GUID generato dal db.

La struttura della tabella è la seguente: id\_link, url, id\_post (fk), utente (fk)

Tutte le correzioni sono da salvare in una tabella dedicata, con i campi: id\_correzione, id\_link (fk), url\_precedente, url\_nuovo, data\_di\_modifica, azione.

Il campo azione definisce quale genere di correzione è stata fatta (correzione/ripristino), infatti il programma oltre a verificare link non validi verifica anche se questi ultimi sono tornati ad essere validi (es: sito target torna online).

Preparare anche una pagina di rapporto nel sito, dedicata all'amministratore, per visualizzare tutte le correzioni, con funzioni di ricerca per data, utente, url e azione. Il rapporto deve essere in forma tabellare ed esportabile come csv.

# Compito

Elaborare una tabella dei requisiti secondo lo schema visto durante il corso, con almeno i seguenti campi: ID, Nome, Descrizione, Priorità.

Disegnare a scelta un diagramma d'uso (use case) o di flusso.

Identificare i possibili problemi che possono crearsi con questa soluzione.

# Criteri di valutazione

* Identificazione di tutti i requisiti
* Leggibilità e rispetto dei criteri per i requisiti
* Completezza del diagramma
* Leggibilità del diagramma
* Identificazione di almeno un problema

# Tabella dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Dovrà essere presente un sito web nel quale gli utenti possano caricare gli articoli. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Web |
| **Sotto Requisiti** | |
| **001** | Il sito dovrà essere adatto a tutti i tipi di utente (daltonismo, …) |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Il programma dovrà avere accesso e poter leggere tutti i link del sito Web. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Software |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Il programma dovrà essere in grado di correggere i link non validi e salvati nella tabella dedicata. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Software |
| **Sotto Requisiti** | |
| **001** | Dovrà essere avviato automaticamente ad ogni inserimento di un collegamento |
| **002** | Il tempo di esecuzione del programma dovrà essere inferiore a 3 sec. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Il programma dovrà essere in grado di aggiungere uno stile al link tramite la classe **text-danger**. |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Categoria** | Software |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Il programma dovrà essere in grado di poter aggiungere un parametro **[invalid]** prima del testo di qualsiasi link errato. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Software |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Il programma dovrà essere in grado di cambiare il percorso **src** di un’immagine di un link non funzionante con ".\images\invalid.png". |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Software |
| **Sotto Requisiti** | |
| **001** | L’immagine **invalid.png** dovrà già essere presente. |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Il programma dovrà essere eseguito ogni notte dopo i backup giornalieri. |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Categoria** | Software |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Il programma, ad ogni inserimento di un collegamento ottiene un id come **<a id=[id\_generato].** |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Software |
| **Sotto Requisiti** | |
| **001** | L’id dovrà essere salvato nella tabella dei collegamenti. |
| **002** | L’utente deve avere la possibilità di specificare degli id personali |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Dovrà essere presente un database. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Hardware |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Il database dovrà avere una struttura ben definita. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Struttura |
| **Sotto Requisiti** | |
| **001** | La struttura sarà la seguente: id\_link, url, id\_post (fk), utente (fk) |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Tutte le correzioni dovranno essere salvati in una tabella dedicata |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Hardware |
| **Sotto Requisiti** | |
| **001** | La struttura sarà la seguente: id\_correzione, id\_link (fk), url\_precedente, url\_nuovo, data\_di\_modifica, azione. |
| **002** | Il campo azione definisce quale azione è stata fatta (correzione/ripristino) |

|  |  |
| --- | --- |
| **ID: REQ-012** | |
| **Nome** | Il programma dovrà verificare se i link sono tornati ad essere validi |
| **Priorità** | 2 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-013** | |
| **Nome** | Preparare una pagina di rapporto nel sito, dedicata all’amministratore. |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Sotto Requisiti** | |
| **001** | Dovrà comprendere le correzioni |
| **002** | Dovrà comprendere funzioni di ricerca per data, utente, url e azione. |

|  |  |
| --- | --- |
| **ID: REQ-014** | |
| **Nome** | Il rapporto deve essere in forma tabellare |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Sotto Requisiti** | |
| **001** | Dovrà essere esportabile in csv. |

# Use case

programma

scheduler

correzione link

backup giornaliero

esegue

DB

pagina report

sito

visualizza

admin

Tabella collegamenti

Tabella correzione

id

sito

aggiunta articolo

correzione link

errato

valido

scannerizza

commento

aggiunge

user

# Problema soluzione

Uno dei problemi più grandi potrebbero essere le ridondanze degli id all’interno del database. Alcuni utenti potrebbero inserire gli stessi id per il collegamento, quindi consiglio di mettere l’id come **unique** per evitare questo problema. Un altro problema è quello delle prestazioni che difficilmente rimane sotto il termine stabilito di 3 secondi se si dovessero verificare molte richieste nello stesso momento. Come ultimo problema potrebbe essere che un link potrebbe essere considerato errato per sbaglio e rimarrebbe tale fino al prossimo controllo impedendo all’utente di accederci. La correzione dei link deve per forza essere eseguita prima del backup perché sennò si rischia di archiviare dei dati errati.