**Test 3**

Nome e cognome: André Da Silva

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

### Specifica dei requisiti:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Nome | Descrizione | Priorità |
| Req-1 | Linguaggio di programmazione | Il programma deve essere scritto in PHP. | 1 |
| Req-2 | Scansione link | Il programma scansiona i link del blog e si assicura che essi siano validi. | 1 |
| Req-3 | Modifica link | Se i link non sono validi, il programma cambia l’url con un’immagine indicante che il link non è valido. | 2 |
| Req-4 | Verifica Link | Il programma verifica se i link sono tornati ad essere validi | 1 |
| Req-5 | Momento di esecuzione | Il programma deve essere eseguito dopo i backup giornalieri e dopo l’inserimento di un link da parte di un utente. | 1 |
| Req-6 | Tempo di esecuzione | L’esecuzione del software deve durare meno di 3 secondi. | 1 |
| Req-7 | DBMS | È necessario un DBMS per la creazione di un database per il salvataggio dei link creati. | 1 |
| Req-8 | MYSQL | Preferibilmente, il DBMS deve essere MySQL. | 2 |
| Req-9 | Tabella collegamenti | È necessario creare una tabella in cui salvare i link inseriti dagli utenti con il seguente schema: link\_id, url, id\_post(fk), user(fk). | 1 |
| Req-10 | Tabella correzioni | È necessario creare una seconda tabella, collegata alla prima, in cui salvare tutte le correzioni effettuate sui link con il seguente schema: id\_correzione, id\_link (fk), url\_precedente, url\_nuovo, data\_di\_modifica, azione. | 1 |
| Req-11 | Algoritmo GUID | È necessario sviluppare un algoritmo che permetta di generare degli GUID per l’id di ogni collegamento da inserire nella tabella. | 2 |
| Req-12 | Pagina rapporto | Occorre creare una pagina nel sito per gli amministratori, al fine di visualizzare ogni correzione effettuata dal software. | 1 |
| Req-13 | Funzione di ricerca | La pagina di rapporto deve avere una funzione che permetta di cercare una correzione di un link in base alla data di correzione, l’utente che ha creato il link, url e azione (correzione/ripristino). | 2 |
| Req-14 | Formato pagina | La pagina di rapporto nel sito deve essere in formato tabellare | 2 |
| Req-15 | Esportazione correzioni | È necessaria la creazione di una funzione che permetta di salvare i dati della pagina di rapporto in formato csv. | 1 |

### Use Case:

### Problemi:

* Dato che tutti gli utenti possono pubblicare link, è possibile che essi pubblichino gli stessi identici causando cosi una ridondanza di dati nel database ed andando così a sprecare spazio inutilmente.
* Se è presente una aggiunta quasi costante di link nel blog da parte di molti utenti (ad esempio in un solo giorno), è possibile che ci sia un sovraccarico causato dal tempo di verifica del programma nel convalidare i link. Ciò può ancora peggiore se il tempo di esecuzione specificato nei requisiti non viene soddisfatto.