**Test 3**

Nome e cognome: Kushtrim Rushi

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.

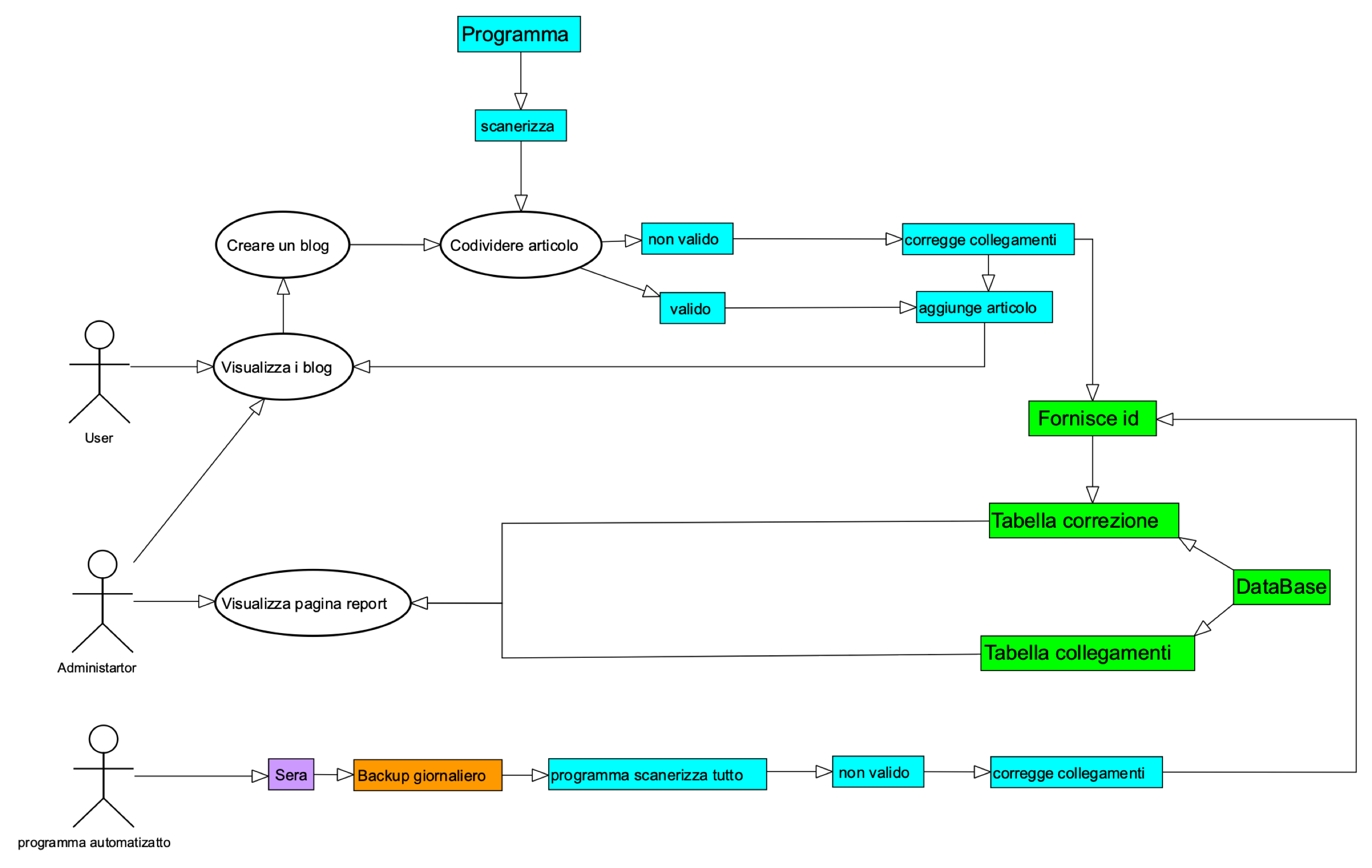
# Criteri di valutazione

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

# Requisiti

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Nome** | **Descrizione** | **Priorità** |
| Req – 01 | Programma controllo collegamenti | Il programma permette di controllare la validità dei collegamenti esterni, automaticamente. | 1.0 |
| Req – 02 | Correzione collegamenti | Una volta che il programma ha notificato un collegamento non valido lo stesso programma lo corregge come definito, cambiando anche la descrizione. | 1.0 |
| Req – 03 | Controllo delle immagini | Lo stesso programma controlla il percorso delle immagini, controlla che i percorsi siano validi. | 2.0 |
| Req – 04 | Correzione percorso immagini | Una volta che il programma notifica un’immagine con un percorso non valido, lo stesso programma cambia l’immagine con quella di non validità (.\images\invalid.png). | 2.0 |
| Req – 05 | Backup giornaliero | Ogni notte bisogna fare un backup sull’intera giornata. | 1.0 |
| Req – 06 | Programma automatizzato | Il programma viene eseguito ogni notte automaticamente, viene eseguito subito dopo il backup giornaliero. | 1.0 |
| Req – 07 | Programma avviato dopo una modifica di un utente | Quando un utente modifica/aggiunge qualcosa, il programma viene avvito solo per la modifica dell’utente. | 2.0 |
| Req – 08 | Generazione id collegamenti | Quando un utente aggiunge un collegamento questo riceve un id generato dal database. L’utente non può in alcun modo modificare questo id. L’id è univoco per link (collegamento). | 1.0 |
| Req – 09 | Database dei link | Ci deve essere un db che permette di salvare tutti i collegamenti. | 1.0 |
| Req – 10 | Tabella collegamenti | Permette di salvare tutti i collegamenti che ci sono all’interno del blog.  Tabella: id\_link, url, id\_post (fk), utente (fk)  (fk) = foreign key | 1.0 |
| Req – 11 | Tabella modifica di un link | Si salvano tutte le modifiche che si sono fatte nel blog di un link.  Tabella: id\_correzione, id\_link (fk), url\_precedente, url\_nuovo, data\_di\_modifica, azione. | 1.0 |
| Req – 12 | Controllo collegamenti validi | Il programma controlla la non validità dei link, ma controlla anche che questi ultimi ritornano ad essere validi. | 1.0 |
| Req – 13 | Pagina di rapporto | Una pagina esclusiva all’amministratore dove vengono riportati tutte le correzioni fatte da un utente. | 2.0 |
| Req – 14 | Visualizzazione rapporti (tabellare) | Nella pagina dei rapporti ci dovrà essere una tabella dove saranno riportati le correzioni. | 3.0 |
| Req – 15 | Esportazione in csv | La tabella dove sono riportati le correzioni potrà essere esportata in un file csv. | 4.0 |

# Use case



# Problemi

Il problema più grande è la gestione di multiutenti, è possibile avere nel database dei collegamenti duplicati dato che più utenti potrebbero aggiungere lo stesso collegamento. Un altro problema riguardante il multiutente è nel caso che tanti utenti aggiungano allo stesso momento più blog, equivarrebbe a dire una diminuzione delle prestazioni che porterebbe a non poter assicurare il tempo di esecuzione di controllo di un post in meno di 3 secondi.