

Trillium

Analisi progetto

Indice

Scopo del documento	1
Obiettivi	1
Requisiti funzionali	2
Generali	2
Utente non autenticato	2
Utente autenticato	2
Requisiti non funzionali	3
Prestazioni	3
Accessibilità	3
Sicurezza	3
Privacy	4
Operatività	4
Compatibilità	4
Design frontend	4
Accesso alla dashboard - desktop	4
Accesso alla dashboard - mobile	5
Sezione di ricerca - mobile	6
Design backend	7

Scopo del documento

Il presente documento analizza e descrive i requisiti di sistema del progetto Trillium in linguaggio naturale.

In particolare vengono:

- Definiti gli obiettivi.
- Elencati i requisiti funzionali e non funzionali.
- Riportati i requisiti e il design di Front-End e Back-End.

Obiettivi

Il progetto ha come obiettivo la realizzazione di un'applicazione web compatibile con la maggior parte dei dispositivi desktop e mobile recenti che permette di caricare e condividere con gli altri utenti dei contenuti testuali o multimediali come video e foto.

Nello specifico, l'applicazione deve:

- Rendere possibile la pubblicazione di post con contenuti multimediali quali foto, video e messaggi, definendo chi può visualizzarli quindi possono essere visualizzabili da tutti (pubblici) o visualizzabili solo dai propri amici (privati).
- Fornire al singolo utente una pagina personale dove poter visualizzare, modificare le proprie pubblicazioni e visualizzare gli utenti seguiti e i follower.
- Rendere possibile agli utenti mettere "Mi piace" o "Non mi piace" ai post pubblicati dagli altri utenti e commentarli.
- Fornire la possibilità di seguire un utente visualizzando i suoi post pubblici in home.
- Fornire all'utente la possibilità di richiedere/accettare l'amicizia permettendo di vedere/far visualizzare i post privati.

Requisiti funzionalit

Requisiti generali

RF 1. Tipo di applicazione

- L'applicazione deve essere web-based e accessibile da diversi dispositivi quali computer, tablet e smartphone.

RF 2. Ruoli utente

- Il sistema deve risultare accessibile in due modi:
 - **Utente non autenticato:** L'utente che accede alla piattaforma per la prima volta oppure che vuole solamente "dare un'occhiata".
 - **Utente autenticato:** L'utente che ha deciso di registrarsi per contribuire alla community caricando nuovi contenuti.

RF 3. Metodi di autenticazione

- L'applicazione deve permettere all'utente di registrarsi e successivamente autenticarsi tramite username e password oppure Google, Facebook.

RF 4. Ricerca rapida

- L'applicazione memorizza i tag più cercati dagli utenti e li suggerisce per una ricerca rapida.

Utente non autenticato

RF 5. Visibilità dei post

- Deve poter visualizzare solamente i post pubblici pubblicati dagli altri utenti.

RF 6. Ricerca avanzata

- Può cercare tramite tag o nome altri utenti e visualizzare i loro contenuti pubblici.

Utente autenticato

RF 7. Funzionalità di base

- Deve poter utilizzare le stesse funzionalità di un utente non autenticato.

RF 8. Creazione dei post

- Deve poter creare e pubblicare dei post “Trill” e definire i tag, una eventuale descrizione, un titolo e quali utenti possono visualizzare i post, questi possono essere:
 - Pubblici: chiunque può visualizzarli.
 - Privati: solo gli amici possono visualizzarli.

RF 9. Logout

- Deve poter uscire dal proprio account.

RF 10. Rimozione dei post

- Deve poter eliminare i propri post caricati precedentemente.

RF 11. Post Feedback

- Può mettere “mi piace” ai post degli altri utenti e visualizzare quelli messi dagli altri ai propri contenuti.

RF 12. Controllo email

- Durante l’inserimento della mail da parte dell’utente, deve essere controllata la sua validità. La validità viene controllata nel seguente modo:
 - Ci deve essere almeno un carattere prima della chiocciola
 - Una chiocciola
 - Almeno un carattere dopo la chiocciola
 - Un punto
 - Almeno un altro carattere dopo il punto

Esempio email valida: cristian.scapin@google.com

Esempi email non valida:

- @google.com
- cristian.scapin@go

→ cristian@.com

Requisiti non funzionali

RNF 1. Tempi di ricerca

I tempi di attesa per la ricerca tramite tag devono essere inferiore ai 4 secondi.

RNF 2. Prestazioni vecchi dispositivi

- Le immagini devono essere compresse per garantire adeguate prestazioni anche nei dispositivi più vecchi.

RNF 3. Tempi di caricamento

- I post devono essere caricati in blocchi e non tutti contemporaneamente per evitare lunghi tempi di attesa una volta che gli utenti cresceranno.

Accessibilità

RNF 4. Design applicazione

- L'applicazione deve avere un design con icone semplici e diversificate per garantire una lettura chiara e immediata delle informazioni.

RNF 5. Leggibilità del testo

- Il testo deve risultare leggibile anche da chi utilizza dispositivi mobile.

RNF 6. Accessibilità

- L'applicazione deve risultare utilizzabile anche da eventuali utenti ipovedenti o daltonici.

RNF 7. Supporto multilingua

- L'applicazione deve essere utilizzabile sia in Italiano che in Inglese.

Sicurezza

RNF 8. Password sicure

- Le password devono rispettare gli standard di sicurezza, i vincoli sono i seguenti:

- Da 8 a 24 caratteri
- Almeno una lettera minuscola
- Almeno una lettera maiuscola
- 1 carattere speciale tra i seguenti: !"#\$%&'()*+,-.\/:;<=>?@[^_{}~

RNF 9. Hashing password

- Il salvataggio delle password deve garantire la sicurezza di tutti gli account in casi di data breach, tramite hashing con salt and pepper.

RNF 10. Chiamate criptate

- Le chiamate alle varie API devono essere criptate tramite protocollo HTTPS.

RNF 11. Sicurezza accesso

- L'accesso all'utente verrà bloccato nel caso commetta 3 errori consecutivi durante la fase di login. Per lo sblocco dovrà essere inviata una mail all'account per il recupero con successivo inserimento di nuove credenziali.

RNF 12. Sicurezza logout

- Il logout dell'utente deve essere eseguito rispettando la sua privacy e sicurezza: tutti i dati salvati nel browser dovranno essere eliminati.

Privacy

RNF 13. Norme di sicurezza

- Devono essere rispettate le norme presenti il GDPR.

RNF 14. Sicurezza sui dati

- I dati raccolti dall'applicazione non dovranno essere condivisi a terzi.

RNF 15. Visibilità dei dati

- L'utente potrà modificare la visibilità dei dati.

RNF 16. Diritto all'oblio

- L'utente può in qualsiasi momento cancellare tutte le informazioni personali presenti sulla piattaforma.

Operatività

RNF 17. Orari di operatività

- L'applicazione deve essere disponibile almeno tra le 8:00 e le 22:00 UTC+1.

Compatibilità

RNF 18. Compatibilità tra piattaforme

- L'applicazione deve avere le stesse features tra Desktop, Android e IOS.

Design frontend

Accesso alla dashboard - desktop

- L'utente, sia autenticato che non, all'apertura della webapp vedrà la **dashboard**, schermata principale che permette di avere una overview dei contenuti.
 - Nel caso l'utente abbia fatto l'accesso vedrà i post degli altri utenti che segue.
 - Nel caso di utente non autenticato verranno invece caricati post pubblici della sezione esplora.
- Tramite la barra di ricerca posizionata nell'header è possibile cercare specificatamente dei contenuti.
- Tramite il bottone apposito verrà aperta la schermata di creazione post.
- Tramite la sezione tag è possibile filtrare velocemente i contenuti secondo i tag più usati.



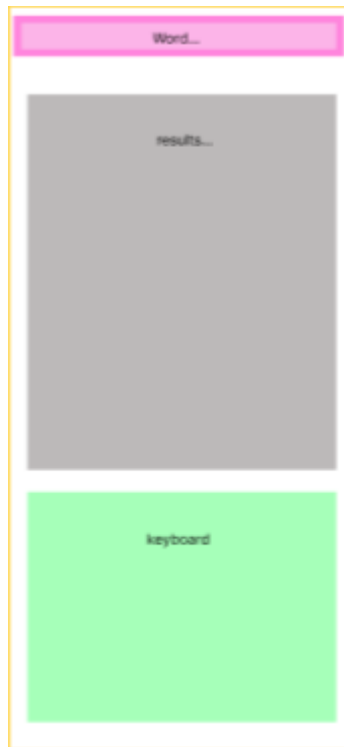
Accesso alla dashboard - mobile

- L'utente, sia autenticato che non, visualizzerà la lista dei contenuti come nella versione desktop.
- I bottoni per le sezioni principali sono stati posizionati sotto per un facile accesso con una mano.
- La barra di ricerca, posizionata in alto, apre una schermata di ricerca dove è possibile inserire testo e selezionare i tag.



Sezione di ricerca - mobile

- Schermata per ricercare utenti e post caricati.
- Possibile inserire tag per filtrare meglio i post.



Design backend

I sistemi esterni con cui trillium dovrà interfacciarsi sono:

1. OAuth per l'autenticazione tramite Google e Facebook
2. Google cloud per l'upload in cloud di media caricati dall'utente, nello specifico video o foto relative ai post e immagini di profilo
3. DiceBear api per la creazione di immagini profilo casuali generate tramite un seed (seme)

