

SquadUp

Mattia Panni
Riccardo Fiorani
Matteo Susca

13 febbraio 2023

Indice

1	Analisi	2
1.1	Requisiti	2
1.2	Descrizione del dominio	3
2	Design	4
2.1	Architettura	4
2.2	Funzionalità richieste	5
2.2.1	Sign-up e Sign-in	5
2.2.2	Feed	7
2.2.3	Posts	8
2.2.4	Commenti	8
2.2.5	Follow	8
2.2.6	Pagine Profilo	9
2.2.7	Notifiche	11
2.3	Funzionalità extra	11
2.3.1	Like di post	11
2.3.2	Map	12
2.3.3	Ultimo accesso utenti	12
2.3.4	Gruppi	12
2.3.5	Ricerca	14
2.3.6	Sicurezza e hashing delle password	14
2.3.7	Utilizzo di chiamate HTTP con Axios	14

Capitolo 1

Analisi

L'obiettivo del progetto è quello di sviluppare un'applicazione web accessibile, responsive e mobile first, che metta a disposizione le funzionalità principali dei social network.

1.1 Requisiti

Progetto assegnato per il corso di *tecnologie web* con i seguenti requisiti

- *Design*: per un massimo di 3 punti.
- *Registrazione/Login*: per un massimo di 3 punti.
- *Feed*: per un massimo di 4 punti.
- *Posts*: per un massimo di 3 punti.
- *Commenti*: per un massimo di 3 punti.
- *Richieste di amicizia/follow*: per un massimo di 3 punti.
- *Pagine utente*: per un massimo di 4 punti.
- *Notifiche*: per un massimo di 5 punti.
- *Extra/effetti wow*: per un massimo di 4 punti.

1.2 Descrizione del dominio

SquadUp è un servizio web di social networking presso il quale è possibile registrarsi come *utenti*, inviare richieste di amicizia (che potranno essere accettate o rifiutate dall'interessato) ad altri utenti, creare *post*, ovvero messaggi visibili da altri utenti del network. Questi possono includere testo e immagini e possono anche essere “piaciuti” e commentati da altri utenti. La visibilità di questi ultimi è limitata agli amici dell'autore del post. Chiunque abbia la possibilità di visionare il post sarà anche in grado di visionare la lista degli utenti che hanno messo like e, tramite un click, accedere al profilo utente di uno di essi. Sarà anche possibile creare degli *eventi*, che possono essere utilizzati per organizzare una vasta gamma di attività, come feste, conferenze, riunioni aziendali o banali uscite fra amici. Gli utenti con particolari privilegi (in particolare il creatore e gli admin) possono invitare altri utenti a partecipare. Ciascun evento avrà una data di creazione, di inizio e fine ed eventualmente un'immagine. Gli eventi potranno essere pubblici o privati, nel primo caso visibili a tutti gli utenti iscritti al social network, altrimenti solo agli amici del creatore. Il social prevede anche la possibilità di creare *gruppi* di utenti che, alla stregua di questi ultimi, potranno eventi e racchiuderanno i post dei propri utenti. Admins e creatore potranno aggiungere, rimuovere membri e renderli amministratori, modificare nome, immagine o descrizione, mentre solo colui che ha creato il gruppo potrà eliminarlo. Sarà anche possibile, tramite una sezione del social, vedere su una mappa la posizione (aggiornata in tempo reale per coloro che sono online o quella avuta durante l'ultima attività sul social per quelli offline) dei propri amici, nonché la propria. E' anche possibile visionare l'ultimo accesso, qualora non fossero online, degli utenti. Il sito fornisce anche un sistema di notifiche generate al verificarsi di una serie di eventi quali

- Richiesta di amicizia (accettabile premendo sulla notifica)
- Like ad un proprio post
- Commento sotto un proprio post
- La pubblicazione di un post da parte di un nostro amico (cliccando sulla notifica sarà possibile visionare il post stesso)

Capitolo 2

Design

In questo capitolo introdurremo nel dettaglio le scelte architetturali e di design che abbiamo deciso di prendere per il progetto.

2.1 Architettura

Per quanto riguarda l'interazione client-server abbiamo deciso di attuare una soluzione ibrida. In particolare per alcune funzionalità in cui fondamentalmente si raccoglievano dati da form da inviare al server, come la registrazione di utenti al social, creazione di post, ecc. si è realizzato tale comportamento attraverso semplici invocazioni a funzioni PHP di back-end. Per altre funzionalità quali

- Aggiunta di like a post
- Aggiornamento dell'ultimo accesso di un utente e conseguentemente dell'interfaccia utente
- Aggiornamento della posizione di un utente
- Invio e richiesta delle notifiche

dove fare invocazioni a funzioni PHP non fosse sufficiente, si è preferito utilizzare strumenti per l'invio asincrono di richieste HTTP al server. In particolare, per via della sua semplicità di utilizzo, abbiamo optato per l'utilizzo della libreria Axios. Il motivo principale è dovuto alla flessibilità fornita da questo strumento, infatti avevamo l'esigenza di avere un aggiornamento server, e conseguentemente dell'UI che non richiedesse il ricaricamento della pagina per inviare i dati al server (come nel caso dell'aggiunta di like ad un post). Un modo rapido per fare ciò è quello di aggiornare *dinamicamente*

l'HTML tramite JavaScript. A livello implementativo avevamo bisogno di ottenere e inviare dati dal e al server. Per fare ciò passavamo attraverso delle API PHP che facevano da tramite fra i vari script JS e il server stesso.

2.2 Funzionalità richieste

Di seguito verranno descritte brevemente le funzionalità dell'app e vengono forniti i mockup utilizzati per avere un riferimento attraverso cui implementare il front-end.

2.2.1 Sign-up e Sign-in

Interazione client-server

Sia sign-up che sign-in sono stati realizzati utilizzando l'approccio citato in precedenza che sfrutta il ricaricamento della pagina per inviare i dati contenuti nei rispettivi form al server. Tale approccio va più che bene in questo caso dal momento che queste pagine fanno un redirect alla home page del sito web.

Mock-up

Di seguito sono riportati i mockup, per semplicità solo di login, sia della versione mobile che desktop.

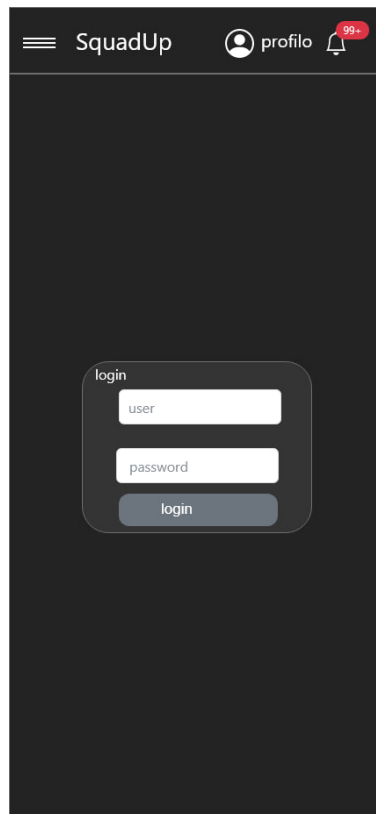


Figura 2.1: Mock up login mobile version.

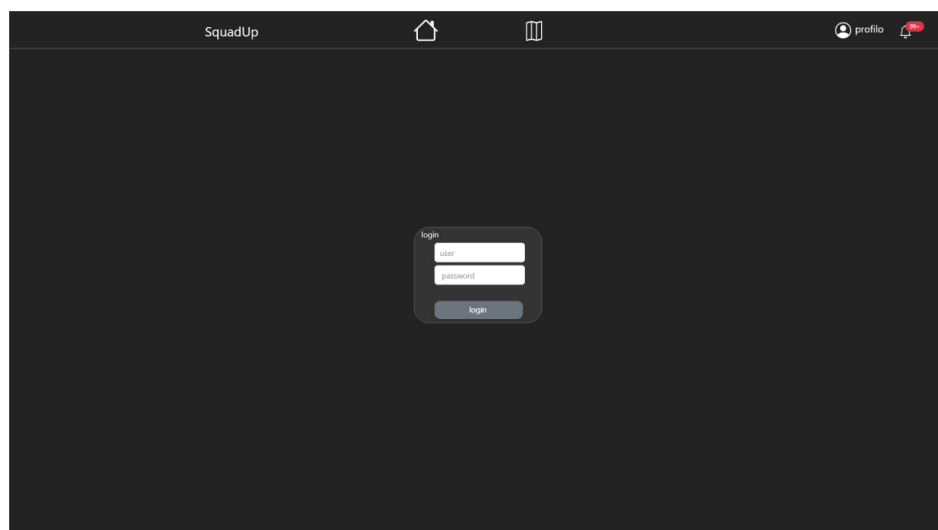


Figura 2.2: Mock up login desktop version.

2.2.2 Feed

Una volta che il sign-in/sign-up sono andati a buon fine, l'utente verrà reindirizzato alla propria home page (o *feed*). La quale è strutturata in tre colonne nella versione desktop e una sola con dei menu hamburger in quella mobile. In essa è possibile vedere i post dei propri amici, mettere like e commentare. Nella colonna sinistra si avrà la possibilità di creare un gruppo cliccando nel bottone apposito che aprirà una finestra modale che consentirà di fare ciò. Nella colonna di destra si potrà vedere una lista dei propri amici (e di ciascuno di essi se online o offline) e una lista dei gruppi di cui si è membri.

Mock-up



Figura 2.3: Mock up feed mobile version.

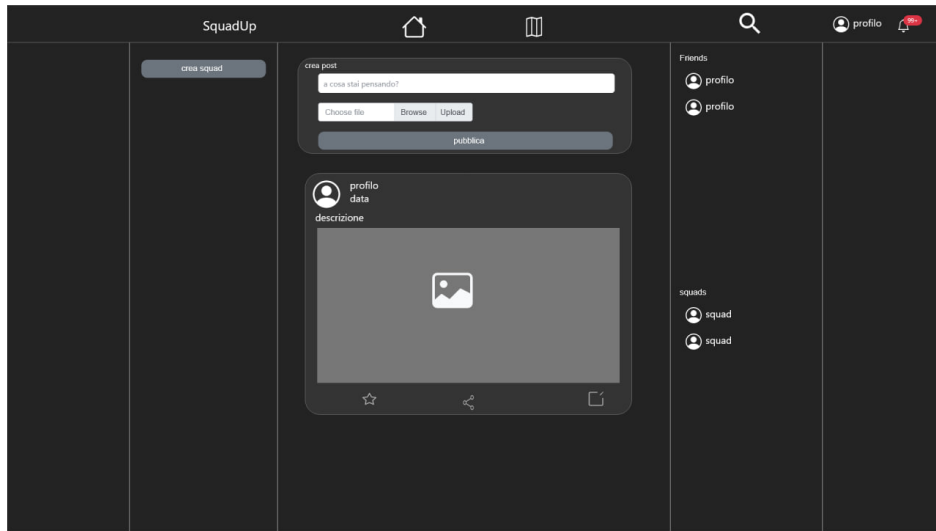


Figura 2.4: Mock up feed desktop version.

2.2.3 Posts

Gli utenti potranno pubblicare dei post che appariranno nella propria pagina utente, in quella degli eventuali gruppi a cui partecipa e nel feed dei propri amici. Ciascun post sarà composto da

- Titolo
- Immagine
- Descrizione

Mock-up

Per i mock-up dei post si veda la struttura mostrata nelle figure 2.3, 2.4.

2.2.4 Commenti

I post possono essere commentati da un qualunque utente che sia in grado di vedere il post in questione.

2.2.5 Follow

Sarà possibile inviare richieste di amicizia agli utenti. Questi ultimi potranno controllare tutte le richieste di amicizia ad essi inviate nell'inbox delle proprie notifiche (che tratteremo in seguito). Da questa area si potrà decidere

se accettare o meno la richiesta. Per inviare una richiesta si dovrà accedere alla pagina dell'utente in questione e cliccare sul bottone "Add friend". Fatto ciò, il bottone verrà disabilitato e comparirà il seguente testo "Pending request". Non appena l'utente accetterà o rifiuterà la richiesta, il bottone tornerà cliccabile e con testo rispettivamente "Remove friend", "Add friend".

2.2.6 Pagine Profilo

Le pagine profilo si presentano, a tre colonne così come la homepage. Sulla destra vi sarà una colonna indicante gli amici dell'utente e i gruppi ai quali egli partecipa. Al centro, invece, verranno mostrati i post dell'utente stesso. Il sistema riconoscerà se ci si trova nella propria pagina e in tal caso sarà presente anche la possibilità di creare nuovi post. Sulla colonna di sinistra, invece, sarà presente un pannello di configurazione qualora ci si trovi nel proprio profilo (il quale consentirà di cambiare foto, nome, descrizione, ecc.), altrimenti dei bottoni per aggiungere/rimuovere l'amico, come già menzionato sopra.

Mock-up



Figura 2.5: Mock up profile mobile version.

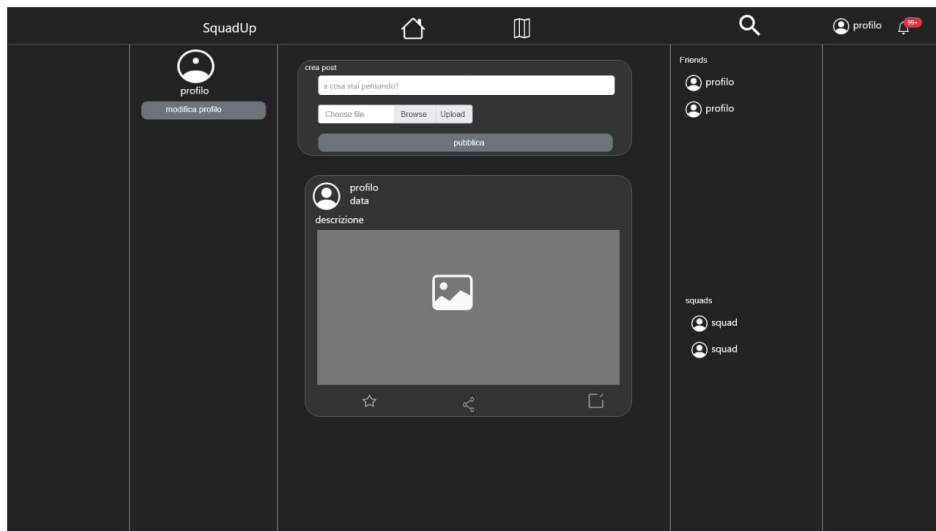


Figura 2.6: Mock up profile desktop version.

2.2.7 Notifiche

Per le notifiche esiste un menù a tendina accessibile attraverso la pressione di un bottone contenente il numero di notifiche non ancora visualizzate. Al suo interno verranno inserite tutte le notifiche che riguardano determinati eventi relativi al nostro profilo, al profilo di un nostro amico o ad un gruppo al quale apparteniamo. In particolare, come già descritto nell'introduzione, le notifiche saranno causate da richieste di amicizia, like a propri post, commenti, ecc. Per visualizzare i dettagli della notifica sarà sufficiente cliccarci, questa pressione, inoltre, la farà scomparire dall'inbox.

2.3 Funzionalità extra

In questa sezione verranno elencate le funzionalità extra:

2.3.1 Like di post

Ogni utente avrà la possibilità di mettere like al post di un qualunque membro di un gruppo a cui appartiene o appartenente ad un suo amico. Al verificarsi di tale evento la pagina si aggiornerà dinamicamente aumentando il contatore dei like e aggiornando la lista utenti che li hanno messi, con il proprio nome. Sarà anche possibile, cliccando nuovamente il bottone, rimuovere l'eventuale like posto in precedenza. Per fare una lista che si aggiornasse dinamicamente e che soprattutto fosse adeguatamente posizionata in tutti i contesti (senza

ad esempio andare in overflow se non ci fosse sufficiente spazio nell'area visibile per mostrarlo), si è deciso di utilizzare il framework Tippy.js, basato su Popper.js. L'accessibilità di questi popup dinamici è garantita dal fatto che sono apribili anche mediante utilizzo di tastiera e non solo mouse.

2.3.2 Map

Sfruttando l'API messa a disposizione da Google Maps e qualche chiamata Axios, siamo riusciti ad ottenere una pagina che contenesse la posizione di tutti i nostri amici, nonché la propria. Questa posizione sarà aggiornata in tempo reale qualora l'utente che stiamo visualizzando sia online in quel momento, altrimenti fornirà l'ultima posizione registrata per quest'ultimo.

2.3.3 Ultimo accesso utenti

All'interno di tutto il sito è possibile vedere lo stato di attività di tutti gli utenti. In particolare nelle colonne relative alla lista amici e membri di gruppi sarà possibile controllare se uno specifico utente è online o meno in quell'istante. Infatti sarà presente un pallino affianco alla foto profilo che assumerà colorazione verde se l'utente è online o grigia altrimenti. Inoltre, accedendo al profilo di un qualunque utente, sarà possibile anche visualizzare il suo ultimo accesso (sottoforma di timestamp) sotto il nome. Per ottenere questo comportamento abbiamo implementato due script js, uno per aggiornare il database con l'ultimo accesso avuto sul sito dell'utente con il quale ho fatto l'accesso, l'altro per ottenere l'ultimo accesso di tutti gli utenti. Il primo script, necessario in tutte le pagine del sito, è stato integrato direttamente nel template HTML base da cui tutti gli altri derivano.

2.3.4 Gruppi

Come menzionato nell'introduzione, il sito offre anche la possibilità di avere dei gruppi di utenti che raccolgono i post di questi ultimi e consentono di creare eventi ai quali poter eventualmente aderire.

Mock-up



Figura 2.7: Mock up squad mobile version.

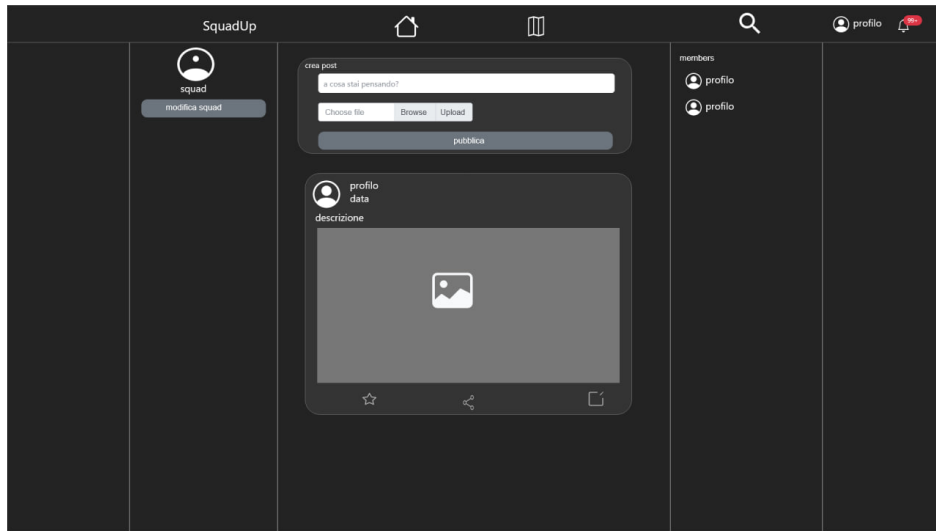


Figura 2.8: Mock up squad desktop version.

2.3.5 Ricerca

Abbiamo implementato un sistema di ricerca di utenti, gruppi ed eventi, che funziona non solamente digitando il nome esatto di uno di essi, bensì anche una versione parziale della stringa, o ad esempio cercando non per username ma per nome o cognome.

2.3.6 Sicurezza e hashing delle password

Dal punto di vista della sicurezza abbiamo optato per strutturare il database in maniera tale da avere una tabella separata, chiamata *login* in cui si mantengono username, password e sale, quest'ultimo è un hash random generato all'interno del PHP. La colonna "username" è una foreign key di un'altra tabella *utente* che mantiene tutte le informazioni relative all'utente, come nome, cognome, mail, data di nascita, ecc. In fase di sign-up sulla tabella login vengono inviati sale e password salata, in fase di login, invece, si controlla che la password inserita, opportunamente hashata con il sale memorizzato nel db, corrisponde alla password precedentemente memorizzata, in caso affermativo il login è andato a buon fine.

2.3.7 Utilizzo di chiamate HTTP con Axios

Come già anticipato, sono state fatte chiamate HTTP a API che interfacciasero UI a server, in modo da riuscire ad aggiornare dinamicamente le pagine HTML senza richiedere il ricaricamento di esse.