

Friend Stuff DB

Autore: Federico Marra 0245408
Corso di laurea: Informatica
Email: marrafederico@proton.me

Indice

| | | |
|----------|---|----------|
| 1 | Introduzione | 2 |
| 1.1 | Problematiche individuate | 2 |
| 1.2 | Target di utenza | 2 |
| 2 | Analisi dei requisiti | 2 |
| 2.1 | Glossario dei termini | 3 |
| 2.2 | Schema Entity-Relationship (ER) | 3 |
| 2.3 | Glossario delle entità | 4 |
| 2.4 | Glossario delle relazioni | 4 |
| 2.5 | Vincoli d'integrità sui dati | 4 |
| 2.5.1 | users | 5 |
| 2.5.2 | user groups | 5 |
| 2.5.3 | events | 5 |
| 2.5.4 | locations | 5 |
| 2.5.5 | expenses | 5 |
| 2.5.6 | chat | 6 |
| 2.5.7 | messages | 6 |
| 2.5.8 | memories | 6 |
| 2.6 | Classi di utenza | 6 |
| 2.7 | Elenco delle operazioni | 6 |

1 Introduzione

Friend Stuff è una piattaforma pensata per semplificare l'organizzazione di eventi, la gestione delle spese comuni e la condivisione dei ricordi all'interno di gruppi di amici.

1.1 Problematiche individuate

L'idea del progetto nasce dall'esigenza di avere un luogo unico dove poter organizzare e gestire le spese legate ad eventi tra persone. Oggi, per pianificare una semplice giornata fuori, ci si ritrova a utilizzare un insieme frammentato di applicazioni:

- app di messaggistica per comunicare,
- calendari condivisi per fissare le date,
- applicazioni finanziarie per gestire le spese,
- servizi di cloud storage per condividere foto e video.

Questo approccio frammentato porta a confusione, informazioni perse e tensioni nei gruppi. Friend Stuff nasce per risolvere proprio queste difficoltà, offrendo un unico spazio digitale, ordinato e trasparente.

1.2 Target di utenza

La piattaforma si rivolge a gruppi che organizzano regolarmente attività insieme e cercano un'alternativa ai social generalisti, mantenendo focus su comunicazione, pianificazione e ricordi condivisi.

2 Analisi dei requisiti

La piattaforma è progettata per offrire a utenti registrati un ambiente centralizzato in cui poter gestire dinamiche sociali e organizzative legate a gruppi ed eventi. Gli utenti, una volta autenticati, hanno la possibilità di instaurare connessioni sociali tramite l'aggiunta di amici e, successivamente, creare gruppi privati composti da altri utenti della piattaforma. All'interno di ciascun gruppo, è possibile organizzare eventi dedicati, a cui partecipano esclusivamente i membri del gruppo stesso. Ogni evento dispone di una chat integrata attraverso la quale i partecipanti possono comunicare in modo diretto. In aggiunta, la piattaforma consente di gestire in maniera centralizzata i contenuti multimediali prodotti durante gli eventi (come foto e video), riducendo la necessità di utilizzare servizi esterni e minimizzando la frammentazione delle informazioni. Gli utenti che prendono parte a un evento possono registrare le spese sostenute e richiedere la suddivisione automatica degli importi. Il sistema calcola le quote individuali da versare o ricevere, semplificando le operazioni di rimborso e contribuendo a evitare incomprensioni tra i partecipanti.

A seguito di una fase di raccolta e analisi dei requisiti, sono state individuate le principali entità che compongono il modello concettuale del sistema:

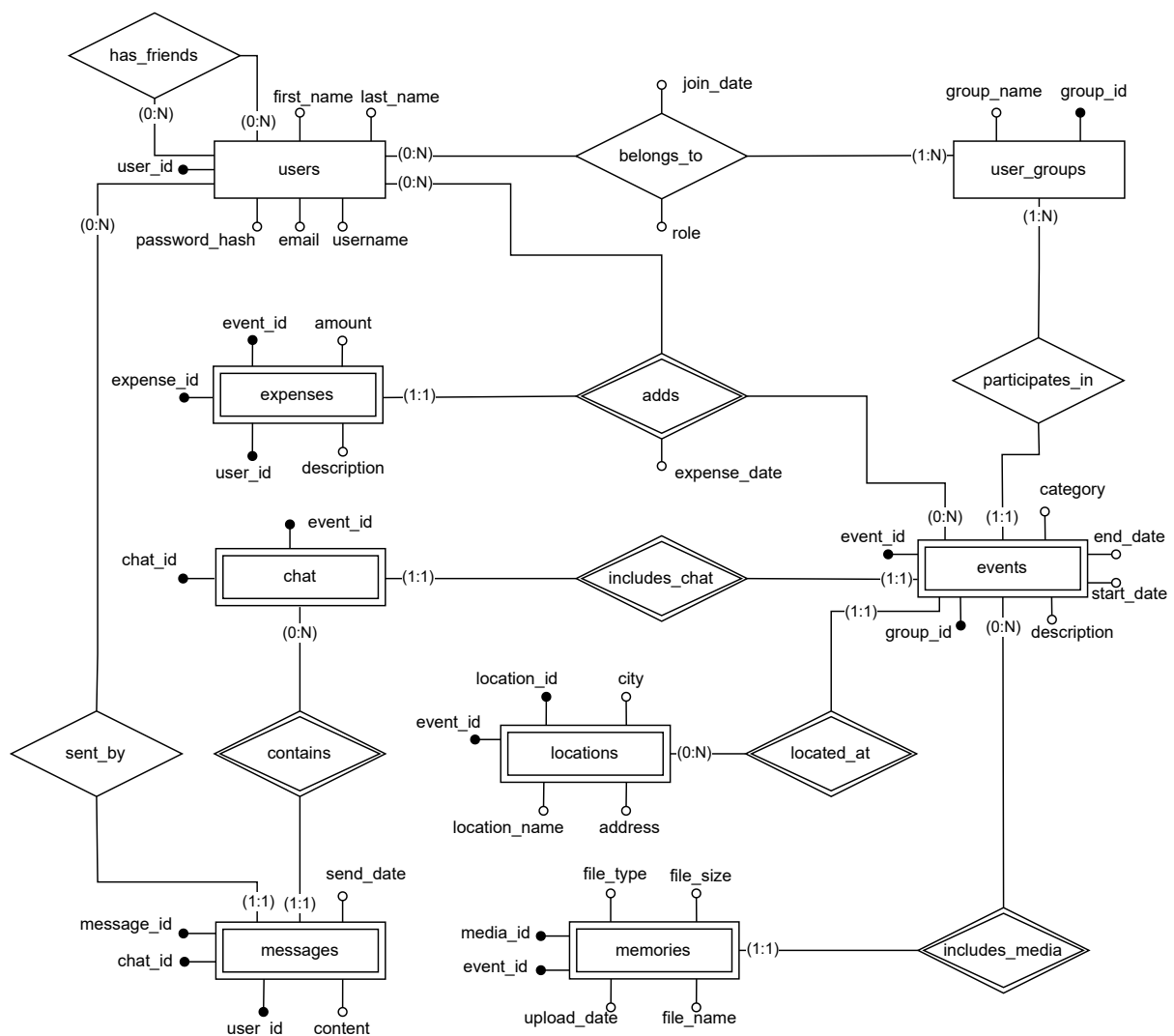
1. users identificata da un codice, si vuole memorizzare username, nome, cognome, email e password hash.
2. user_groups identificata da un codice, ogni gruppo ha uno o più admin, un nome e una data di creazione.
3. events identificata da un codice, caratterizzata da una descrizione, data, luogo e categoria.
4. locations identificata da un codice, caratterizzata dal nome del luogo, la città e l'indirizzo.
5. expenses identificata da un codice. È associata a un evento e a un utente (che effettua la spesa), ed è caratterizzata da un importo e una descrizione.
6. memories identificata da un codice. Permette agli utenti di condividere ricordi relativi a un evento, come immagini o video.
7. chat Ogni evento ha una chat ed è identificata da un codice.
8. messages Rappresenta un singolo messaggio inviato in una chat. È identificato da un codice ed è caratterizzato da un contenuto, una data e un utente.

2.1 Glossario dei termini

| Termine | Descrizione |
|--------------------------|--|
| Utente/Utente registrato | Persona autenticata nel sistema tramite email e password. |
| Gruppo di utenti | Collettivo privato di utenti registrati e autenticati. |
| Evento/Attività | Iniziativa (es. gita, cena, vacanza) organizzata all'interno di un gruppo. |
| Spesa | Transazione economica registrata da un utente durante un evento. |
| Chat | Canale di messaggistica associato a un evento. |
| Ricordo/Media | Contenuto multimediale (foto/video) condiviso in relazione a un evento. |
| Messaggio | Testo inviato da un utente nella chat dell'evento. |

Tabella 1: Glossario dei termini

2.2 Schema Entity-Relationship (ER)



2.3 Glossario delle entità

| Entità | Attributi | Descrizione |
|-------------|--|--|
| users | user_id, email, username, first_name, last_name, password_hash | Identifica un utente registrato sulla piattaforma. |
| user_groups | group_id, group_name | Identifica un gruppo di utenti all'interno della piattaforma. |
| expenses | (expense_id, event_id, user_id), amount, description | Identifica una spesa nel contesto di un evento. |
| events | (event_id, group_id), description, start_date, end_date, category | Descrive l'evento al quale un gruppo di utenti parteciperà. |
| locations | (location_id, event_id), city, locations_name, address | Descrive il luogo in cui si svolge l'evento. |
| memories | (media_id, event_id), file_type, file_size, file_name, upload_date | Identifica i media memorizzati nel database. |
| chat | (chat_id, event_id) | Identifica una chat all'interno di un evento. |
| messages | (message_id, user_id, chat_id), send_date, content | Identifica un messaggio inviato da un utente all'interno di una chat di un evento. |

Tabella 2: Glossario delle entità

2.4 Glossario delle relazioni

| Relazione | Entità coinvolte | Descrizione |
|-----------------|-------------------------|--|
| has_friends | users | Un utente può avere molti amici. Relazione ricorsiva. |
| belongs_to | users, user_groups | Un utente può appartenere a più gruppi. Un gruppo ha almeno un membro. |
| participates_in | users, events | Un gruppo di utenti partecipa a uno o più eventi. Ad un evento può partecipare solo un gruppo. |
| adds | users, expenses, events | Un utente registra nessuna o più spese durante un evento. Un evento ha nessuna o più spese. Una spesa è correlata ad un solo evento ed utente. |
| located_at | events, locations | Un evento si svolgerà in un luogo. In un luogo possono essere stati svolti nessuno o più eventi. |
| includes_media | events, memories | Ogni evento può avere nessuno o più media associati. Un media è correlato ad un evento specifico. |
| includes_chat | events, chat | Ogni evento ha al più una chat associata. Ad un evento è associata al più una chat. |
| contains | chat, messages | Una chat contiene nessuno o molti messaggi. Un messaggio è correlato ad una chat specifica. |
| sent_by | users, messages | Ogni messaggio è correlato ad un utente specifico. Un utente può aver mandato nessuno o molti messaggi. |

Tabella 3: Glossario delle relazioni

2.5 Vincoli d'integrità sui dati

Sono stati identificati i seguenti vincoli d'integrità sui dati per le entità discusse precedentemente.

2.5.1 users

| Attributo | Tipo | Dettagli |
|---------------|----------------------|--|
| user_id | chiave primaria (PK) | INT, AUTO_INCREMENT. |
| email | UNIQUE, NOT NULL | VARCHAR(100). Deve essere un formato email valido. |
| password_hash | NOT NULL | VARCHAR(255). Contiene l'hash della password. |
| username | UNIQUE, NOT NULL | VARCHAR(100) |
| first_name | NOT NULL | VARCHAR(100) |
| last_name | NOT NULL | VARCHAR(100) |

Tabella 4: Vincoli entità users

2.5.2 user groups

| Attributo | Tipo | Dettagli |
|------------|----------------------|----------------------|
| group_id | chiave primaria (PK) | INT, AUTO_INCREMENT. |
| group_name | UNIQUE, NOT NULL | VARCHAR(100) |

Tabella 5: Vincoli entità user_groups

2.5.3 events

| Attributo | Tipo | Dettagli |
|-----------------------|--------------------------------|--|
| (events_id, group_id) | chiave primaria composita (PK) | event_id: INT, AUTO_INCREMENT; group_id: FK verso user_groups. |
| category | NOT NULL | VARCHAR(100) |
| description | NOT NULL | VARCHAR(255) |
| start_date | NOT NULL | DATETIME. Data di inizio dell'evento. |
| end_date | NOT NULL | DATETIME. Data fine evento. La data di fine evento non può essere precedente alla data di inizio evento. |

Tabella 6: Vincoli entità events

2.5.4 locations

| Attributo | Tipo | Dettagli |
|--------------------------|--------------------------------|--|
| (locations_id, event_id) | chiave primaria composita (PK) | locations_id: INT, AUTO_INCREMENT; event_id: FK verso events. |
| locations_name | NOT NULL | VARCHAR(255) |
| city | NOT NULL | VARCHAR(100) |
| address | NOT NULL | VARCHAR(255) |

Tabella 7: Vincoli entità locations

2.5.5 expenses

| Attributo | Tipo | Dettagli |
|----------------------------------|--------------------------------|---|
| (expenses_id, user_id, event_id) | chiave primaria composita (PK) | expense_id: INT, AUTO_INCREMENT; user_id: FK verso users; event_id: FK verso events. |
| amount | NOT NULL | FLOAT |
| description | NOT NULL | VARCHAR(255) |

Tabella 8: Vincoli entità expenses

2.5.6 chat

| Attributo | Tipo | Dettagli |
|---------------------|-------------------------------|---|
| (chat_id, event_id) | chiave primaria composta (PK) | chat_id: INT, AUTO_INCREMENT, event_id: Fk verso events. |

Tabella 9: Vincoli entità chat

2.5.7 messages

| Attributo | Tipo | Dettagli |
|--------------------------------|-------------------------------|--|
| (message_id, chat_id, user_id) | chiave primaria composta (PK) | message_id: INT, AUTO_INCREMENT; chat_id: FK verso chat; user_id: FK verso users. |
| content | NOT NULL | VARCHAR(255) |
| send_date | NOT NULL | TIMESTAMP |

Tabella 10: Vincoli entità messages

2.5.8 memories

| Attributo | Tipo | Dettagli |
|----------------------|-------------------------------|--|
| (media_id, event_id) | chiave primaria composta (PK) | media_id: INT, AUTO_INCREMENT; event_id: FK verso events. |
| file_name | NOT NULL | VARCHAR(255) |
| file_type | NOT NULL | VARCHAR(255) |
| upload_date | NOT NULL | TIMESTAMP |
| file_size | NOT NULL | BIGINT |

Tabella 11: Vincoli entità memories

2.6 Classi di utenza

Durante l'analisi dei requisiti sono state individuate due classi di utenza all'interno di un gruppo:

- **Admin:** un utente con privilegi amministrativi limitati al gruppo che ha creato. Oltre a poter gestire gli utenti del proprio gruppo, l'admin ha anche tutte le funzionalità di un normale membro.
- **Member:** un utente standard del sistema non dispone di permessi amministrativi in un gruppo.

2.7 Elenco delle operazioni

Basandosi sull'analisi dei requisiti svolta finora, sarà possibile svolgere le principali operazioni sui dati per la realizzazione delle seguenti funzioni.

| Funzione | Admin | Descrizione |
|-------------------------|-------|--|
| registration/login | no | Registrazione/autenticazione di un utente sulla piattaforma. |
| add/remove friend | no | Gestione delle amicizie tra utenti. |
| create group | no | Creazione di un gruppo. |
| remove group | sì | Rimozione di un gruppo. |
| create/remove events | sì | Gestione degli eventi all'interno del gruppo. |
| add/remove expense | no | Registrazione o cancellazione di spese personali in un evento. |
| send/delete message | no | Invio o eliminazione di un messaggio nella chat dell'evento. |
| assign admin | sì | Promozione o revoca del ruolo di admin per altri membri. |
| leave group | no | Permette all'utente di uscire da un gruppo. |
| add/remove group member | sì | L'admin può aggiungere o rimuovere utenti dal gruppo. |
| update/remove profile | no | Modifica dei propri dati o cancellazione del profilo. |
| refund friend | no | Rimborso di una quota da parte di utenti. |

Tabella 12: Operazioni sui dati