

# **PROGETTO BASE SPAZIALE**

## **ABSTRACT:**

La base Concordia ha necessità di comunicare con centri di ricerca nel resto del mondo, ma a causa della sua localizzazione la base non ha possibilità di connettersi ad Internet se non in un determinato lasso di tempo ogni giorno. Risulta quindi necessario sviluppare un applicativo che utilizzi tale finestra di tempo per trasferire le informazioni necessarie.

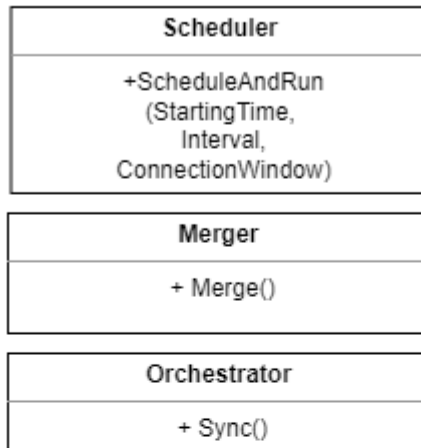
## **REQUISITI:**

- Consentire ai centri di ricerca di aggiungere nuove schede per assegnare nuovi lavori alla stazione.
- Le schede memorizzate nel database della stazione devono contenere i seguenti dati: titolo, descrizione, scadenza (opzionale), etichetta di priorità, ultimo commento aggiunto, assegnatari (uno o molti).
- Allineare il database della stazione e la board Trello, mantenendo le modifiche apportate da entrambi i lati.
- Consentire agli scienziati di aggiungere un commento alle schede, e di spostare le schede da una lista all'altra per indicare il progresso nel lavoro.
- Consentire agli scienziati di visualizzare le schede a loro assegnate, e le schede ordinate in base alla loro priorità.

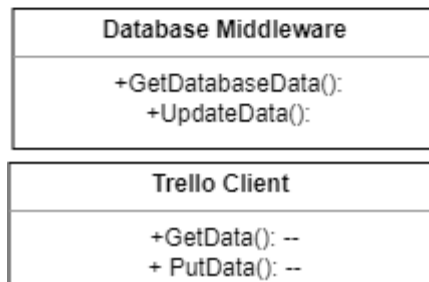
## SOLUZIONE PROPOSTA

Diagramma UML, descrizione visuale dell'architettura e lo scopo di ogni componente / modulo:

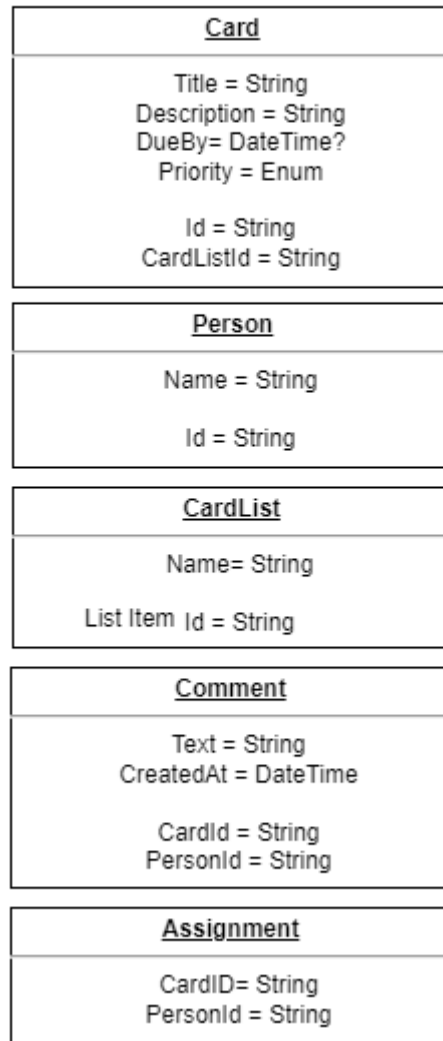
### Architecture

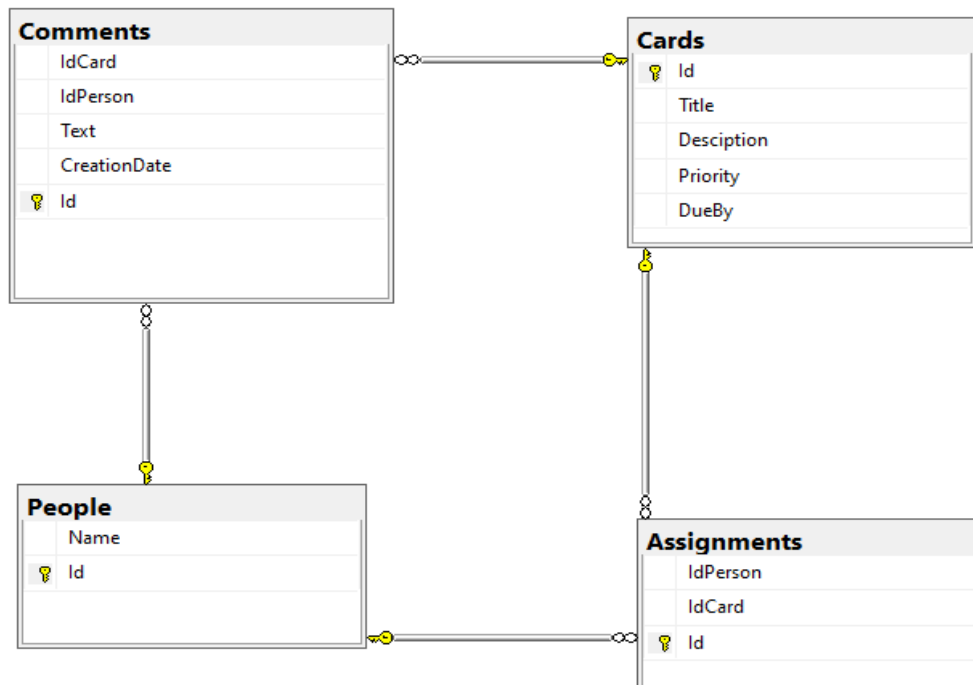
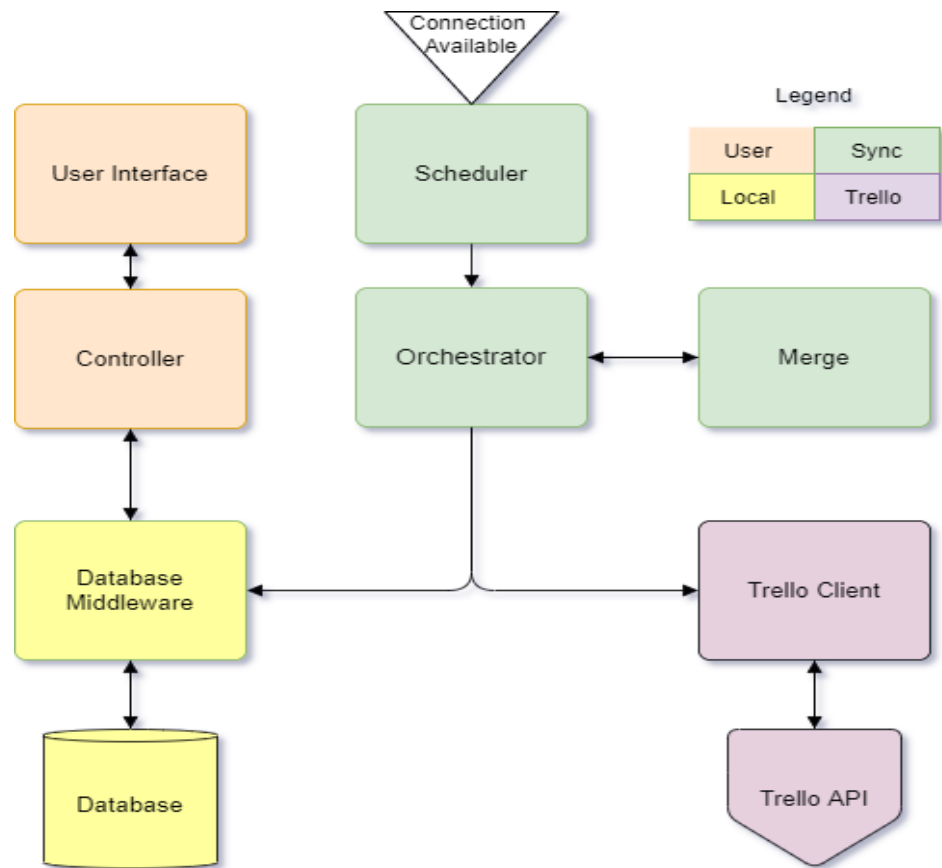


### Input/Output



### Oggetti





### LIMITAZIONI:

La base Concordia può effettuare solo le seguenti modifiche ai dati:

- Creazione di un commento su una scheda
- Spostamento di una scheda da una lista ad un'altra lista

Nessun'altra modifica è consentita: in particolare, le liste non possono essere create localmente dalla base.

I commenti creati dal lato base vengono attribuiti all'account di Trello della base, non ai singoli scienziati.

I commenti creati dal lato base possono, sotto certe circostanze, essere copiati su Trello e in seguito sovrascritti con tale copia: questo comporta una scrittura su database in più di quanto necessario.

Ogni scheda contiene solamente l'ultimo commento ad essere stato aggiunto: se entrambe le parti aggiungono un commento, in seguito alla sincronizzazione rimarrà solamente il commento aggiunto per ultimo.

I centri di ricerca non possono spostare le schede fra varie liste (ad esempio, per indicare che un progetto completato dovrebbe essere riaperto), e quindi devono utilizzare altre vie per comunicare variazioni nello stato delle richieste.

Si assume di non raggiungere mai i limiti delle API di Trello, quindi di non avere mai più di 1000 oggetti di dominio per tipo.

### IPOTESI DI LAVORO:

Team Leader: Igor

Architettura: Igor

Database: Sebastian

Input/output: Giacomo

Tecnologie Utilizzate:

1. Database SQL server
2. L'API di Trello
3. C#
4. NUnit
5. AutoMapper
6. Postman

### TEST:

Come pensate di testare l'applicazione? 10 righe sul vostro sistema di testing che intendete adottare e/o i dati necessari per validare l'implementazione

Unit test + schema per verificare che l'API di trello funzioni come ci si aspetta?

Test:

1. Unit test con NUnit, per coprire la logica di sincronizzazione e le conversioni di AutoMapper
2. Validazione della WebAPI con test di Postman
3. Verifica della comunicazione con Trello

TEMPISTICHE:

Database: 5 ore, 1 persona.

Gateway: 10 ore, 1 persona.

Trello Client: 15 ore, 1 persona.

Master: 5 ore, 1 persona.

Controller: 5 ore, 1 persona.

Scheduler: 10 ore, 1 persona.

Merge: 20 ore, 3 persone.

40 ore per una Proof of Concept con connessioni con Trello e database.