

# ESERCIZIO 3: GRUPPI DI LAVORO (javascript)

Web technologies

# Specifiche:

Si realizzi un'applicazione client server web che modifica le specifiche del progetto html pure come segue:

- L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password", anche a lato client. La registrazione controlla l'unicità dello username.
- Dopo il login, l'intera applicazione è realizzata con un'unica pagina
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- Nella sezione DETTAGLI GRUPPO se l'utente è il creatore del gruppo appare l'icona di un cestino. L'utente può trascinare il nome e cognome di un partecipante sul cestino per cancellarne la partecipazione. A seguito del rilascio sul cestino l'applicazione controlla che sia rispettato il numero minimo di partecipanti. Se il vincolo è rispettato l'operazione è eseguita nella base di dati e un messaggio di conferma viene emesso. Se il vincolo NON è rispettato l'operazione NON è eseguita nella base di dati e un messaggio di diniego viene emesso
- La scelta dall'anagrafica deve essere realizzata con una pagina modale con i bottoni invia e cancella. NB: una pagina modale è una finestra che si apre per dare una qualche scelta o un qualche messaggio all'utente: [https://it.wikipedia.org/wiki/Finestra\\_modale](https://it.wikipedia.org/wiki/Finestra_modale)
- I controlli di correttezza del numero di invitati e del massimo numero di tentativi, con i relativi messaggi di avvertimento, devono essere realizzati anche a lato client.
- Lo stato dell'interazione (numero di tentativi) deve essere memorizzato a lato client.

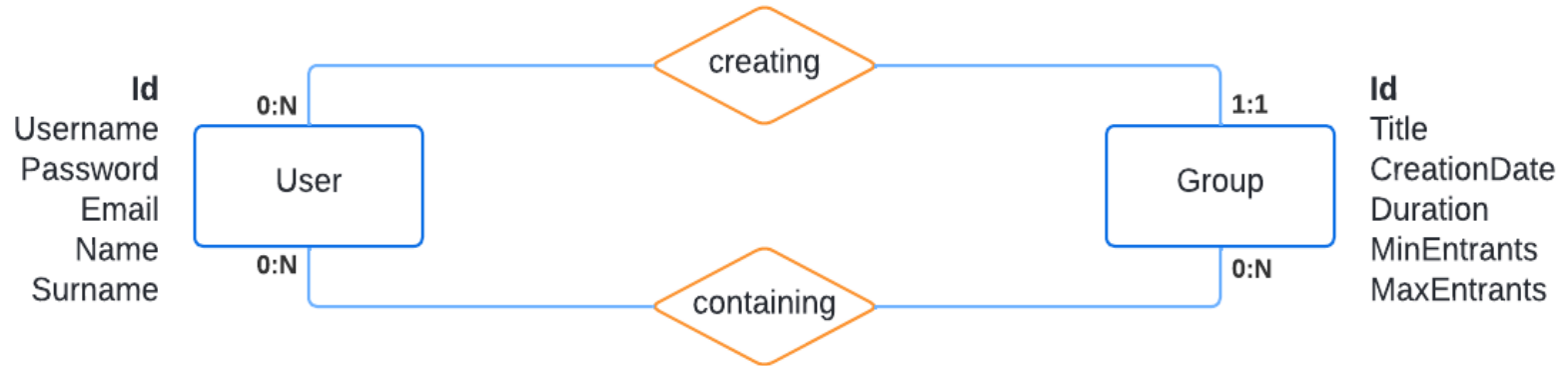
Inoltre, abbiamo assunto che, al login, l'utente sia accolto da un messaggio personalizzato, che ogni dato da compilare nel form sia obbligatorio ai fini della creazione di un nuovo gruppo, che il valore di default per la data di creazione del gruppo sia la data odierna e che, in seguito alla creazione o modifica (cioè eliminazione di un utente) di un gruppo se ne visualizzino i dettagli nella pagina home.

# Analisi dei requisiti: **entità**, **attributi**, **relazioni**

Un'applicazione web consente la gestione di **gruppi di lavoro**. L'applicazione supporta registrazione e login degli **utenti** mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica **dell'indirizzo di email** e l'uguaglianza tra i campi **"password"** e **"ripeti password"**. La registrazione controlla l'unicità dello **username**.

Un gruppo ha un **titolo**, una **data di creazione**, una **durata** dell'attività in giorni e un **numero minimo e massimo di partecipanti**. L'utente fa il login e, se autenticato, accede all'HOME page che mostra l'elenco **dei gruppi creati da lui** e ancora attivi, l'elenco **dei gruppi cui è stato invitato** e ancora attivi, e una form per creare un nuovo gruppo. Quando l'utente seleziona un gruppo compare una pagina DETTAGLI GRUPPO che mostra i dati del gruppo e la lista degli utenti partecipanti identificati da **nome** e **cognome**.

# Database Design



# Local Database

```
CREATE TABLE `User` (  
    `Id` int NOT NULL AUTO_INCREMENT,  
    `Username` varchar(100) NOT NULL,  
    `Password` varchar(100) NOT NULL,  
    `Email` varchar(100) NOT NULL,  
    `Name` varchar(100) NOT NULL,  
    `Surname` varchar(100) NOT NULL,  
    PRIMARY KEY (`Id`)  
)  
  
CREATE TABLE `Entrant` (  
    `GroupId` int NOT NULL,  
    `UserId` int NOT NULL,  
    PRIMARY KEY (`GroupId`, `UserId`),  
    FOREIGN KEY (`GroupId`) REFERENCES `Group` (`Id`)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    FOREIGN KEY (`UserId`) REFERENCES `User` (`Id`)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
)
```

```
CREATE TABLE `Group` (  
    `Id` int NOT NULL AUTO_INCREMENT,  
    `CreatorId` int NOT NULL,  
    `Title` varchar(100) NOT NULL,  
    `CreationDate` date NOT NULL,  
    `Duration` int NOT NULL,  
    `MinEntrants` int NOT NULL,  
    `MaxEntrants` int NOT NULL,  
    PRIMARY KEY (`Id`),  
    FOREIGN KEY (`CreatorId`) REFERENCES `User` (`Id`)  
        ON UPDATE CASCADE  
        ON DELETE NO ACTION  
)
```

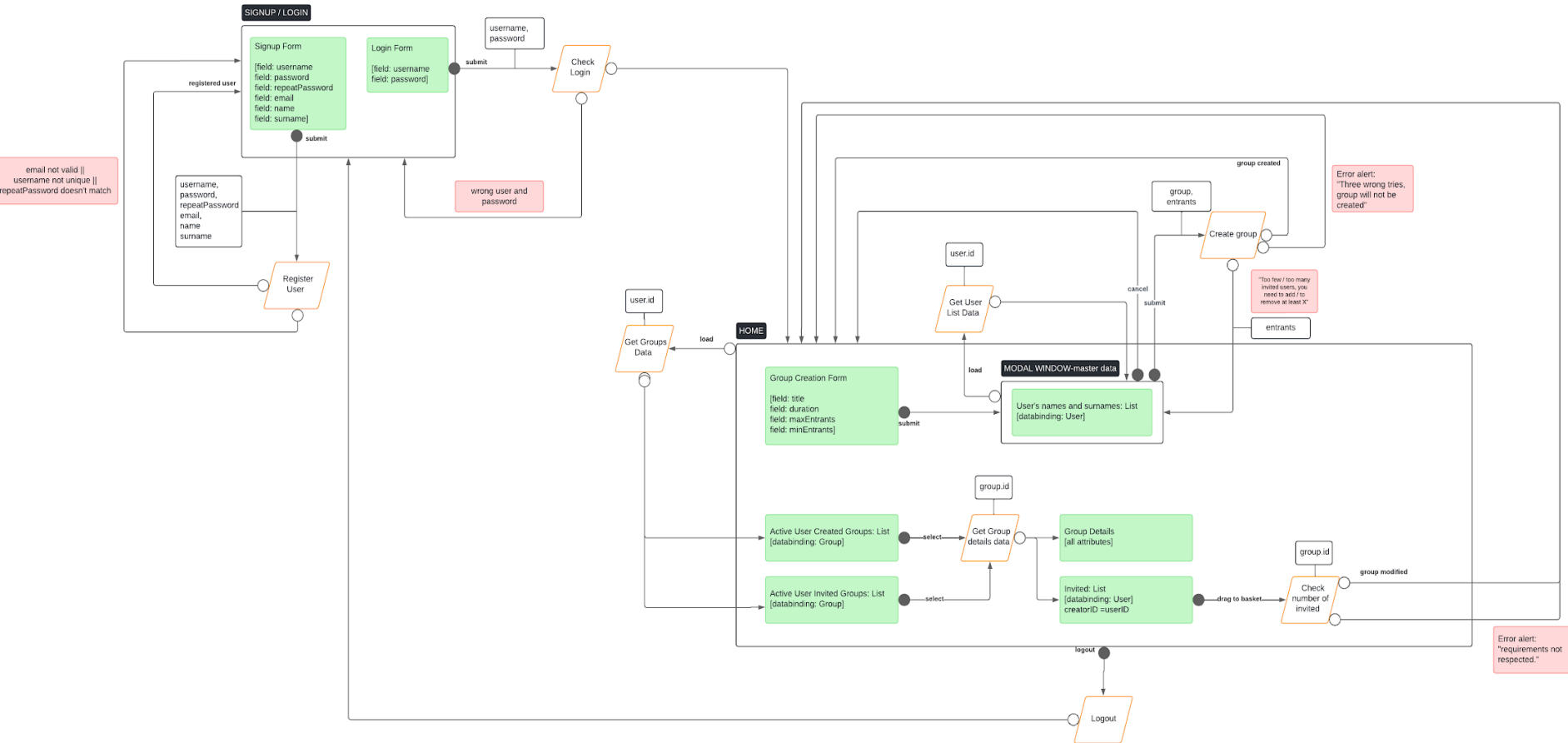
# Analisi funzionale : **pagine (views)**, **componenti delle pagine**, **eventi**, **azioni**

Si realizzi un'applicazione client server web che modifica le specifiche del progetto html pure come segue:

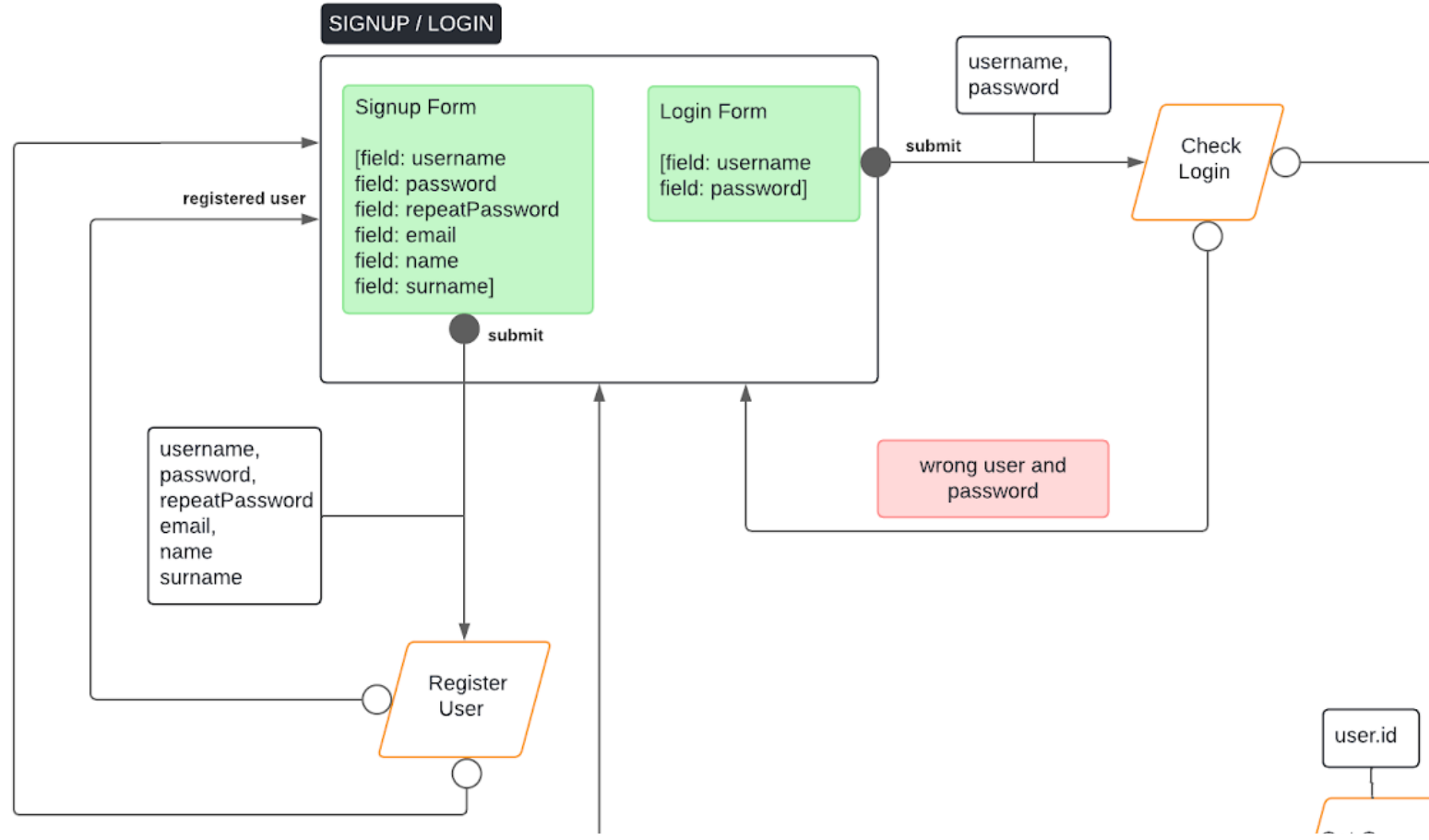
- L'applicazione supporta registrazione e login mediante una **pagina pubblica** con **opportune form**. Quando l'utente **inoltra la form di registrazione**, se i controlli vanno a buon fine, **è memorizzato nella base di dati** e l'utente è rimandato alla pagina pubblica.
- Dopo aver **inoltrato il form di login**, l'intera applicazione è realizzata con **un'unica pagina (pagina HOME)**
- Nella **sezione DETTAGLI GRUPPO** se l'utente è il creatore del gruppo appare **l'icona di un cestino**. **L'utente può trascinare il nome e cognome di un partecipante sul cestino per cancellarne la partecipazione**. A seguito del rilascio sul cestino l'applicazione controlla che sia rispettato il numero minimo di partecipanti. **Se il vincolo è rispettato l'operazione è eseguita nella base di dati** e un **messaggio di conferma** viene emesso. Se il vincolo NON è rispettato l'operazione NON è eseguita nella base di dati e un messaggio di diniego viene emesso
- **La scelta dall'anagrafica** (comprensiva di selezione, invio ed eventuale deselezione) deve essere realizzata con una **pagina modale** con i **bottoni invia** e **cancella**.
- I controlli di correttezza del numero di invitati e del massimo numero di tentativi, con i **relativi messaggi di avvertimento**, devono essere realizzati anche a lato client.
- Lo stato dell'interazione (numero di tentativi) deve essere memorizzato a lato client.

Oltre alle nuove aggiunte di javascript e alle altre componenti già citate nel precedente testo, ricordiamo che anche questa versione avrà nella pagina home **la lista di gruppi cui è stato invitato e quella di gruppi da lui creati** e, **in seguito al click su un determinato gruppo**, verranno mostrati i **dettagli dello stesso**. Inoltre, solo qualora i controlli su correttezza del form, numero di invitati e tentativi diano esito positivo, **il gruppo sarà memorizzato nella base di dati**. Naturalmente è ancora permesso il **logout dell'utente**.

# Application design

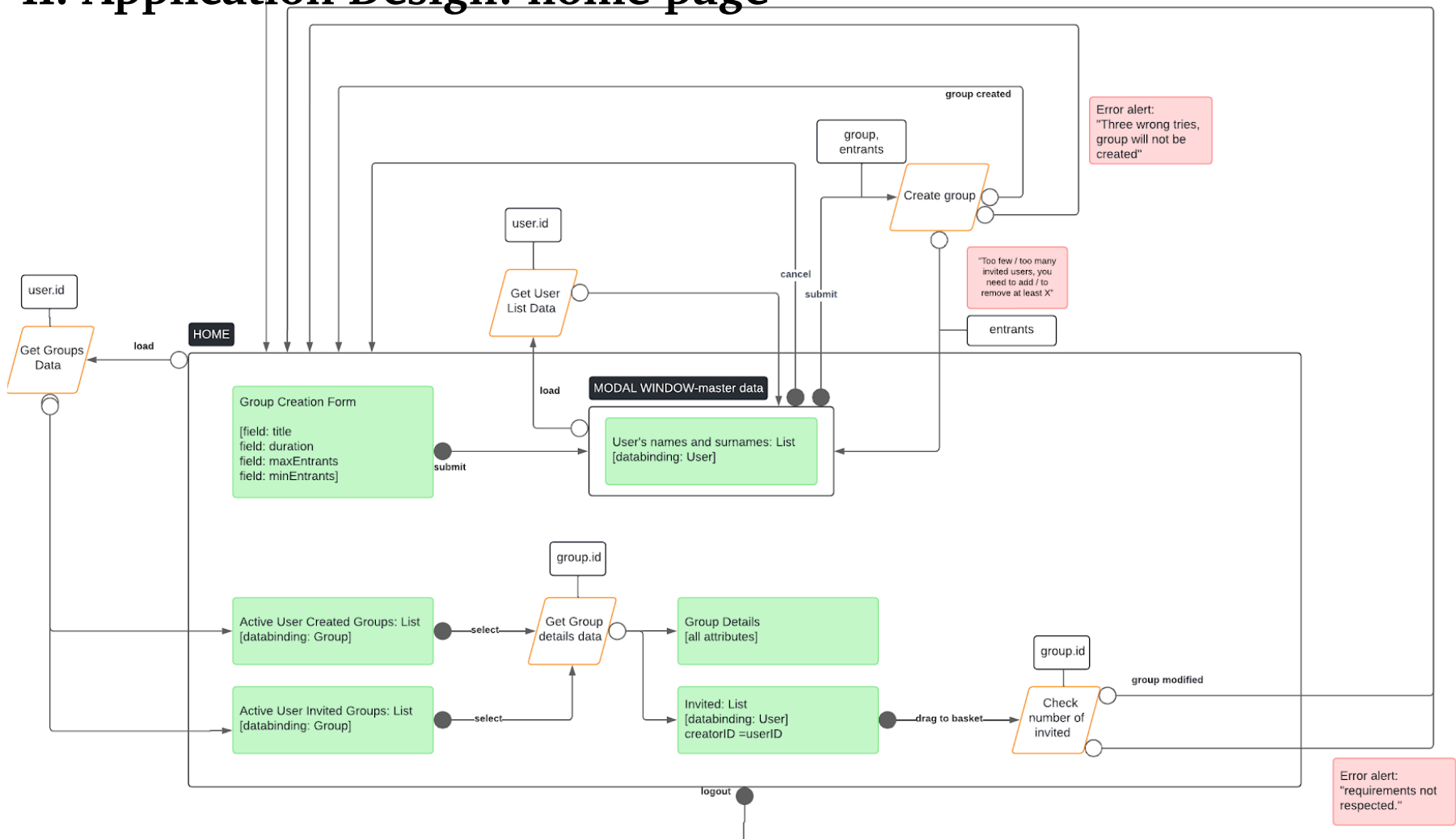


# I. Application Design: signup/login page





# II. Application Design: home page



# Components, server side:

- **Model Objects (Beans)**
  - User
  - Group
- **Data Access Objects (Classes)**
  - **UserDAO**
    - checkCredentials
    - addUser
    - findEntrantsByGroupId
    - findUserById
    - isUsernameUnique
    - findAllUsersOrderedBySurname
  - **GroupDAO**
    - findActiveGroupsByCreatorId
    - findActiveGroupsByEntrantId
    - findGroupById
    - addGroup
    - getGroupFromResult
- **Controllers (Servlet)**
  - CheckLogin
  - RegisterUser
  - GetGroups**Data**
  - GetGroupDetails**Data**
  - GetUserList**Data**
  - CheckInvited
  - **DeleteEntrant**
  - Logout

# Components: Client Side

- **View (Templates)**
  - **Index**
    - Login form  
Gestione del submit e degli errori
    - Registration form  
Gestione del submit e degli errori
  - **Home**
    - CreatedGroups
    - InvitedGroups
    - GroupDetails
    - GroupCreation form
    - Modal Window with master data

## Events & Actions

Client side		Server side	
Evento	Azione	Evento	Azione
index → login form → submit	Controllo dati	POST username password	Controllo credenziali
index → sign up form → submit	Controllo dati	POST username password	Controllo credenziali con unicità username
Home page → load	Aggiorna view con dati elenco	GET (nessun parametro)	Estrazione gruppi dell'utente
Home → elenco gruppi → seleziona gruppo	Aggiorna view con dettagli del gruppo	GET (id gruppo)	Estrazione dati gruppo Estrazione dati partecipanti
Home → Form di Creazione gruppo → cancel	Aggiorna view		
Home → Form di Creazione gruppo → submit	Controllo dati e aggiornamento view con apertura finestra anagrafica	GET (nessun parametro)	Estrazione dati utenti
Home → Create group → submit	Controllo dati e aggiornamento numero di tentativi	POST (group data)	Controllo dati ed aggiunta gruppo
Home → Dettagli gruppo → drag to basket	Controllo dati e aggiornamento view	POST (group e user to delete id)	Controllo dati ed eventuale modifica gruppo
Logout		GET	Terminazione della sessione

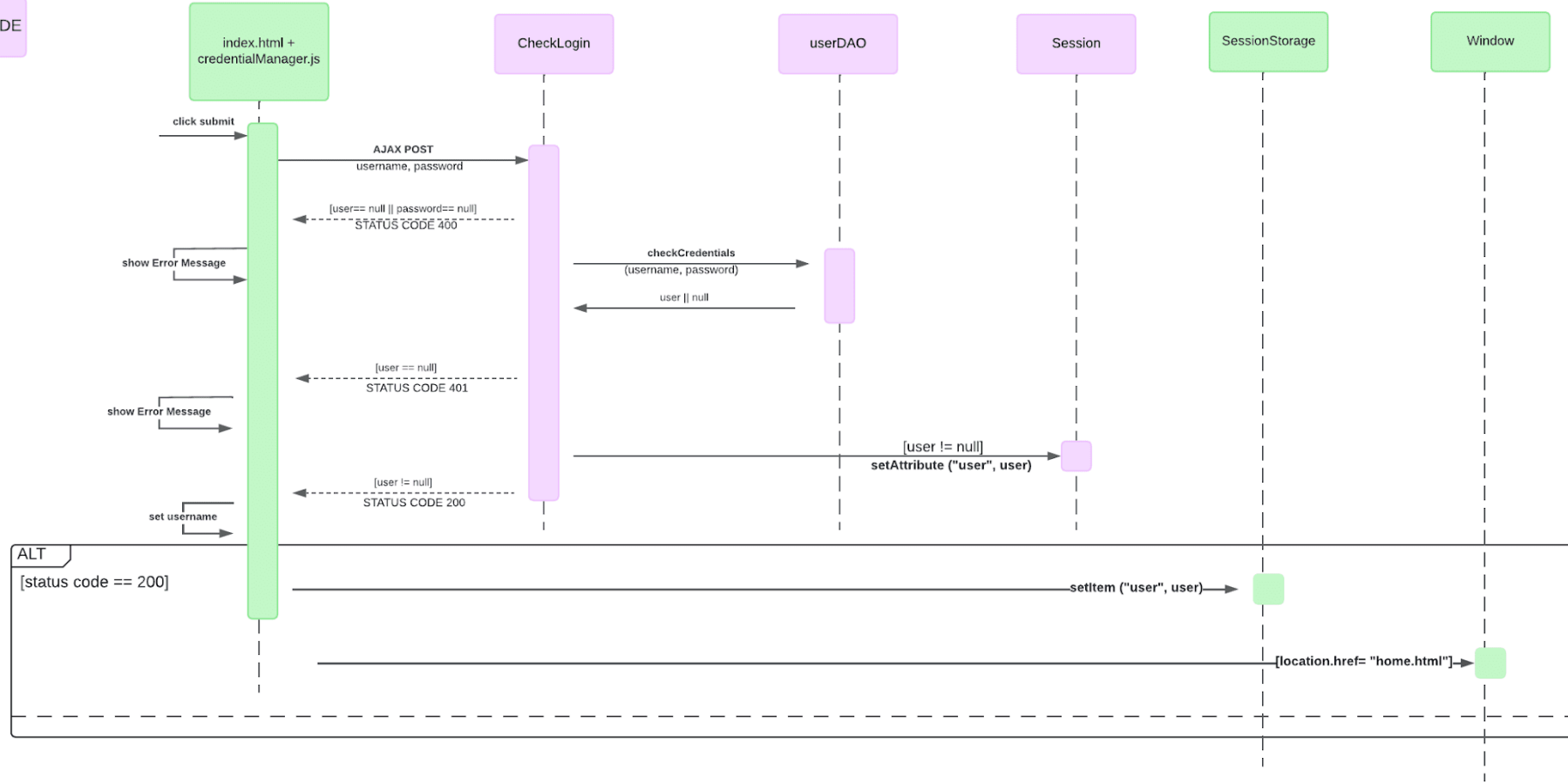
# Controller & Event Handler

Client side		Server side	
Evento	Controllore	Evento	Controllore
index login form submit	Function makeCall	POST username password	CheckLogin
index sign up form submit	Function makeCall	POST username password	RegisterUser
Home page load	Function PageOrchestrator	GET (nessun parametro)	GetGroupListData
Home elenco gruppi seleziona gruppo	GroupDetails.show	GET (id gruppo)	GetGroupDetailsData
Home Form di Creazione gruppo cancel	PageOrchestrator.refresh		
Home Form di Creazione gruppo submit	anagrafica.show	GET (nessun parametro)	GetUserListData
Home Create group submit	Function makeCall PageOrchestrator.refresh	POST (group data)	CreateGroup
Home Dettagli gruppo drag to basket	Function makeCall PageOrchestrator.refresh	POST (group e user to delete id)	CheckNumberOfInvited
Logout		GET	Logout

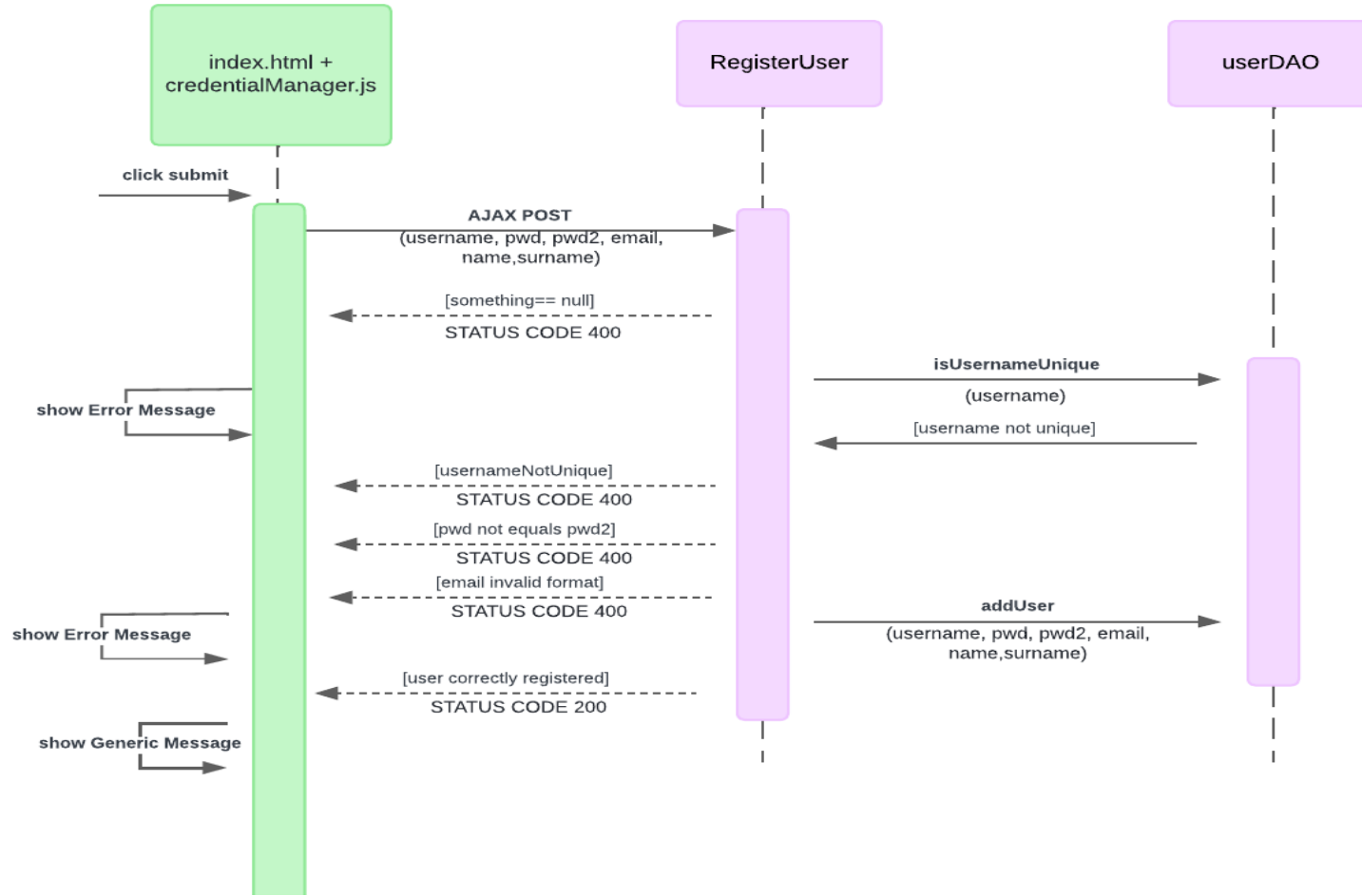
# LOGIN

CLIENT SIDE

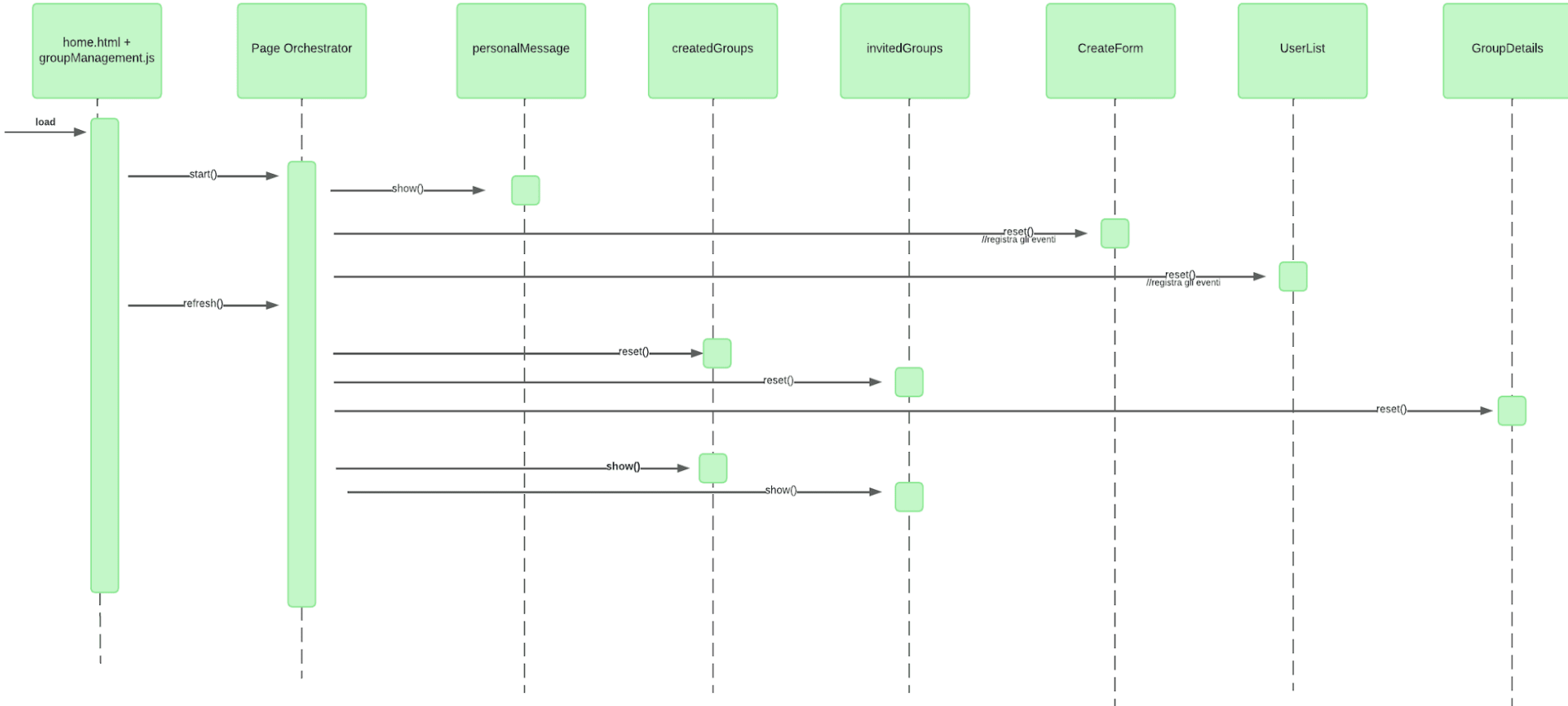
SERVER SIDE



# SIGN UP

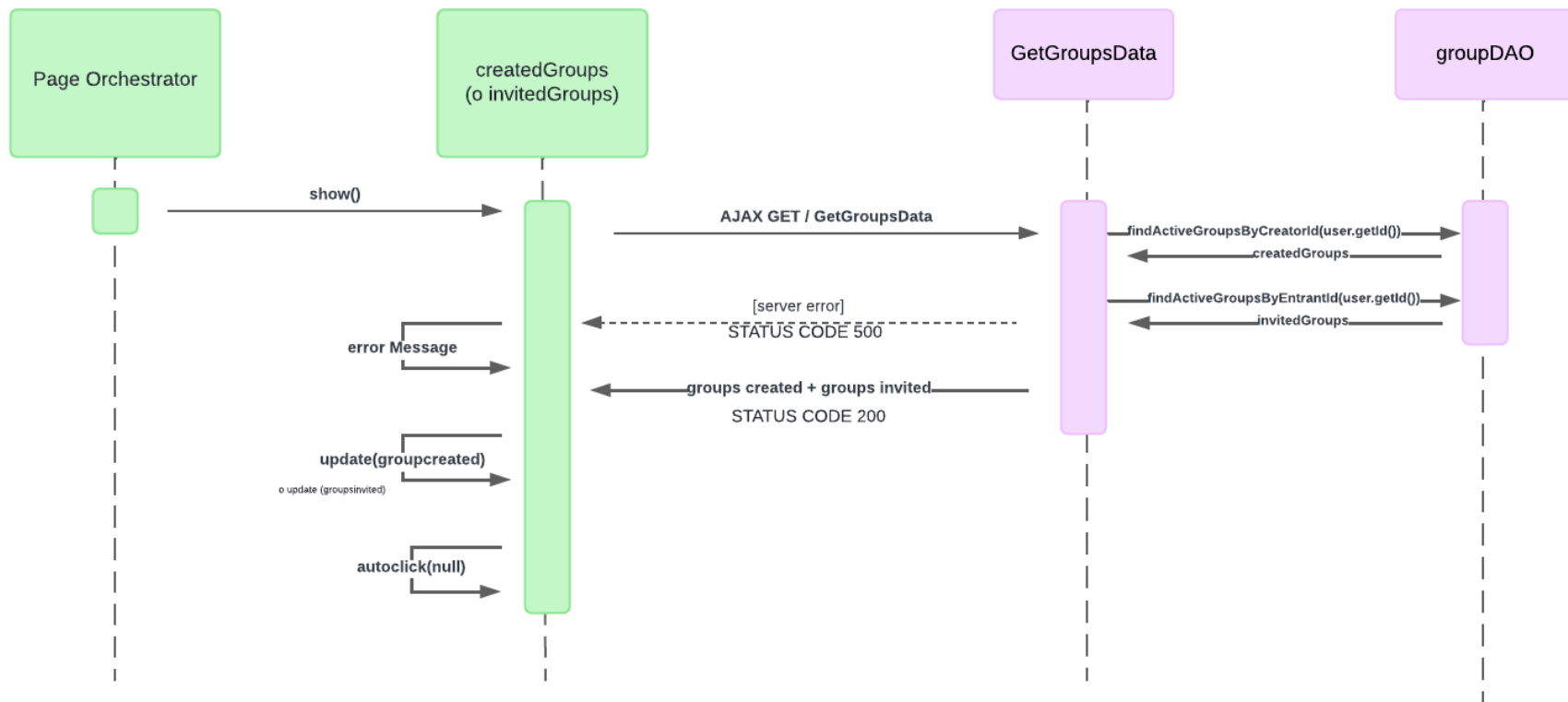


# CARICAMENTO HOME PAGE

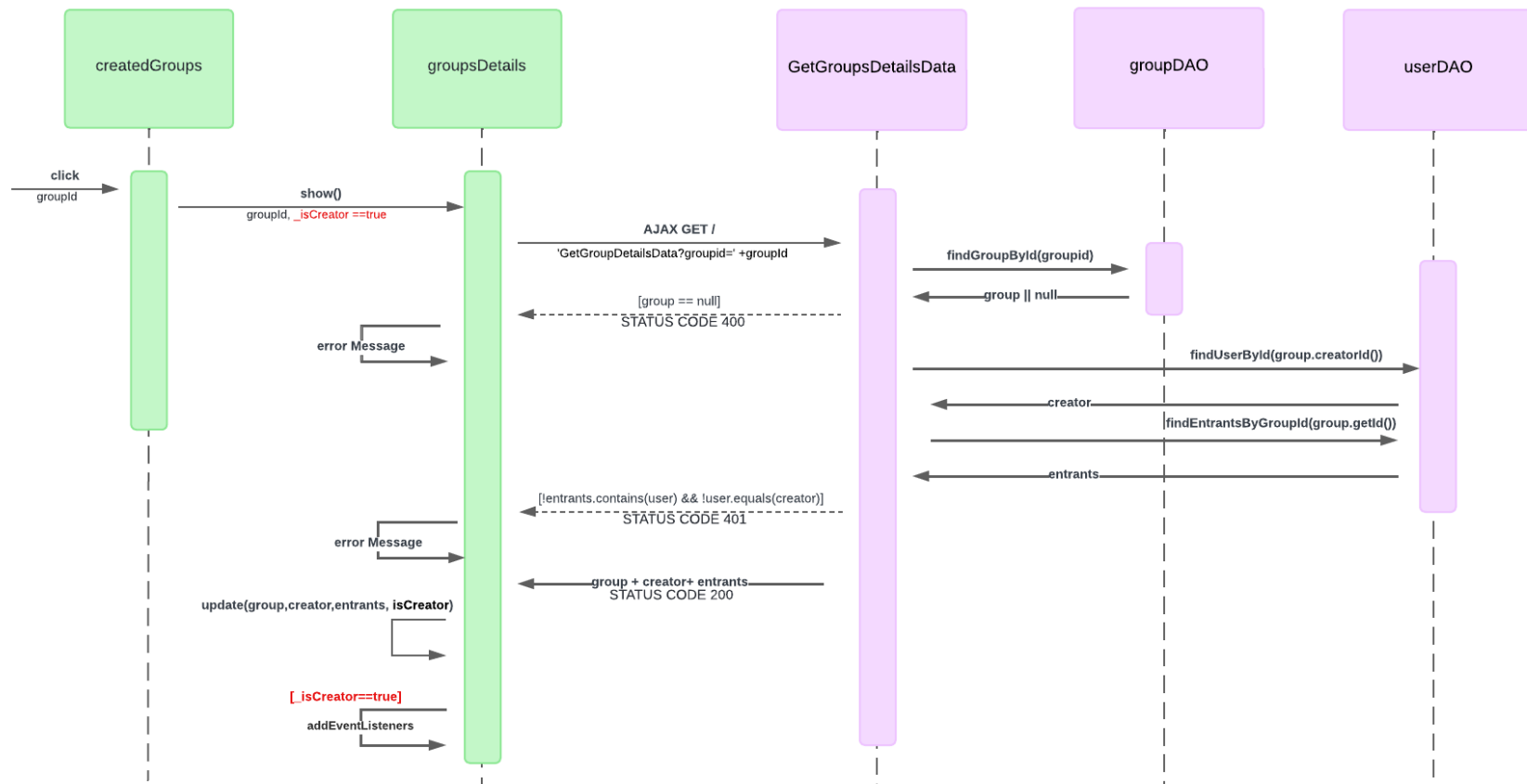




## in dettaglio: show() su createdGroups o invitedGroups

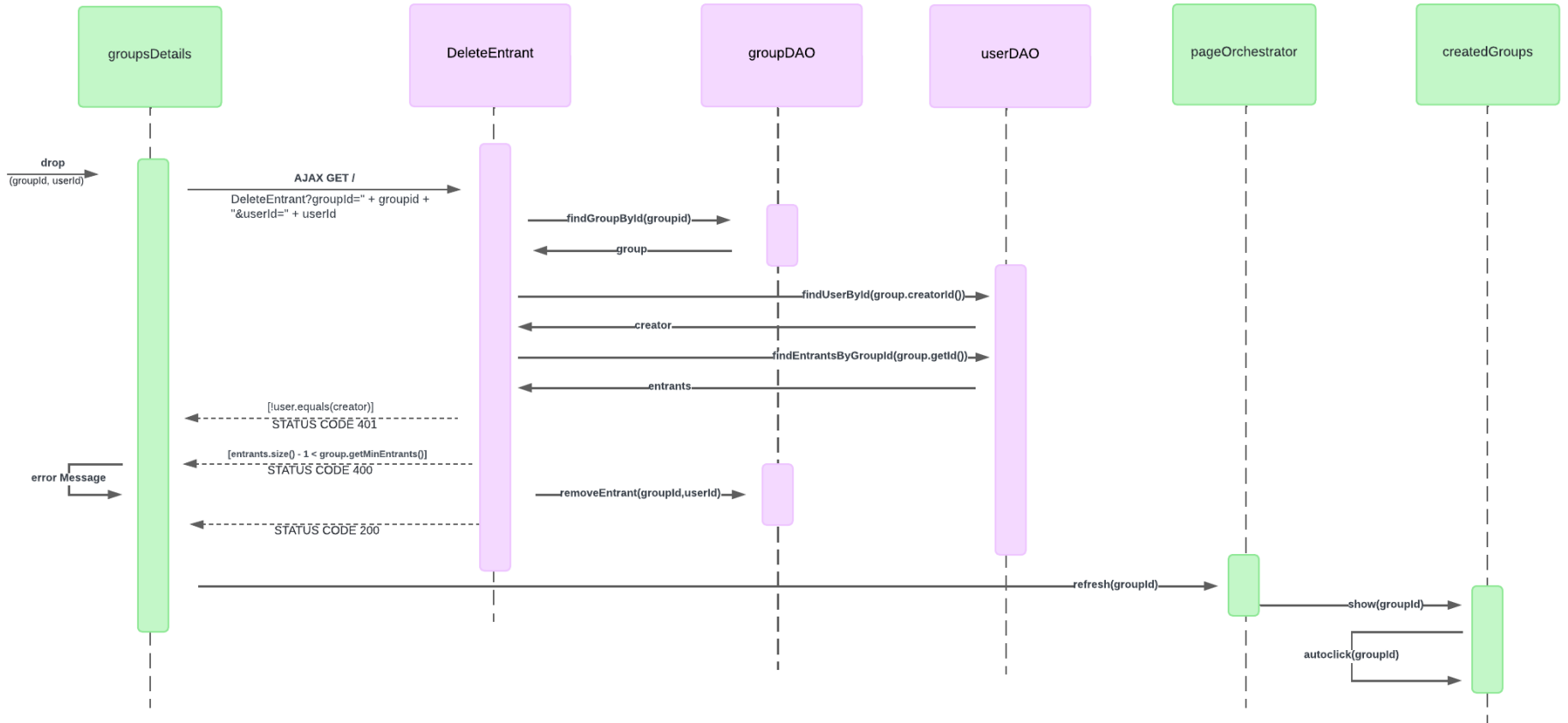


# SELEZIONE DI UN GRUPPO



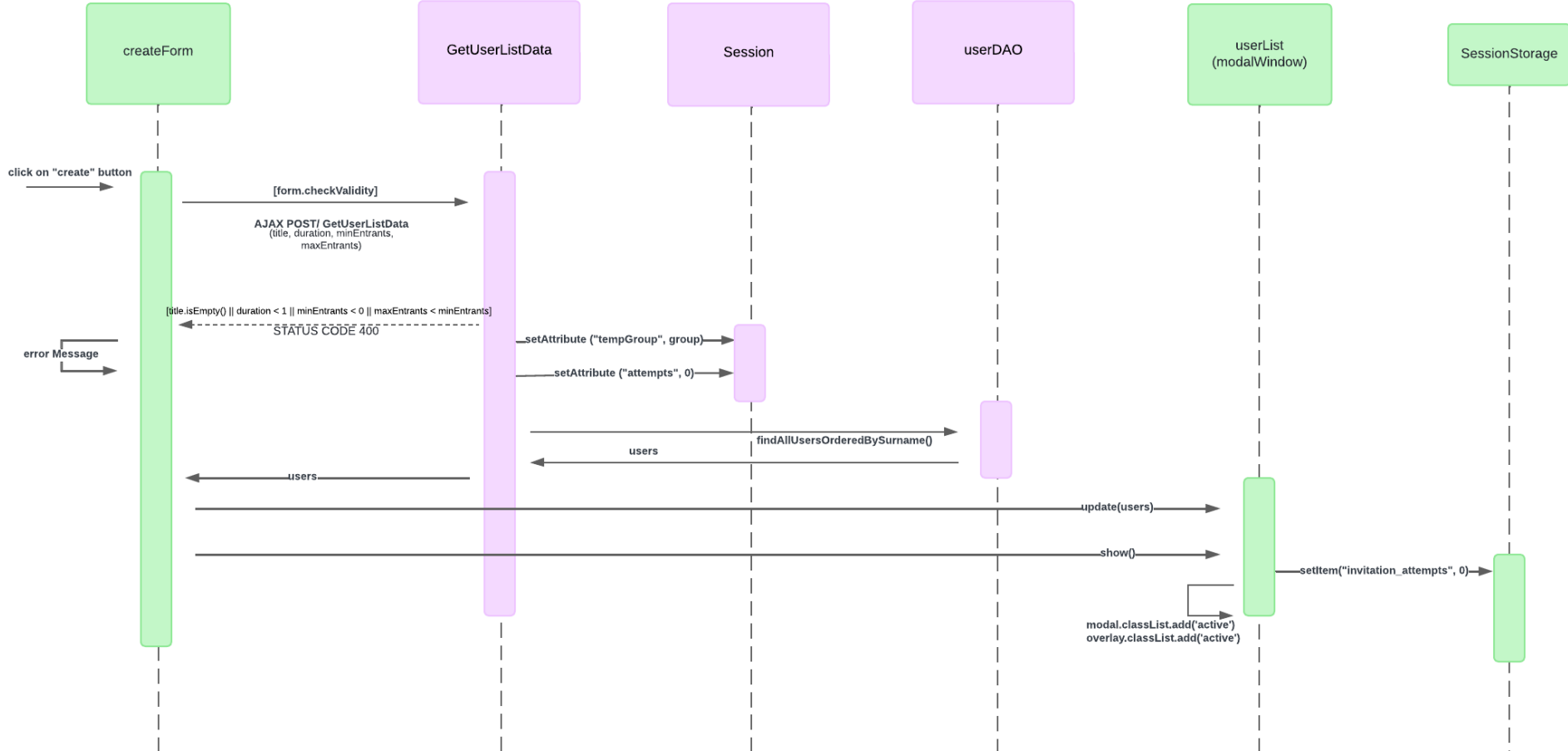
L'unica differenza, nel caso la selezione avvenga su un gruppo di InvitedGroups, è che il booleano **\_isCreator** è inizializzato a false.

# ELIMINAZIONE DI UN UTENTE

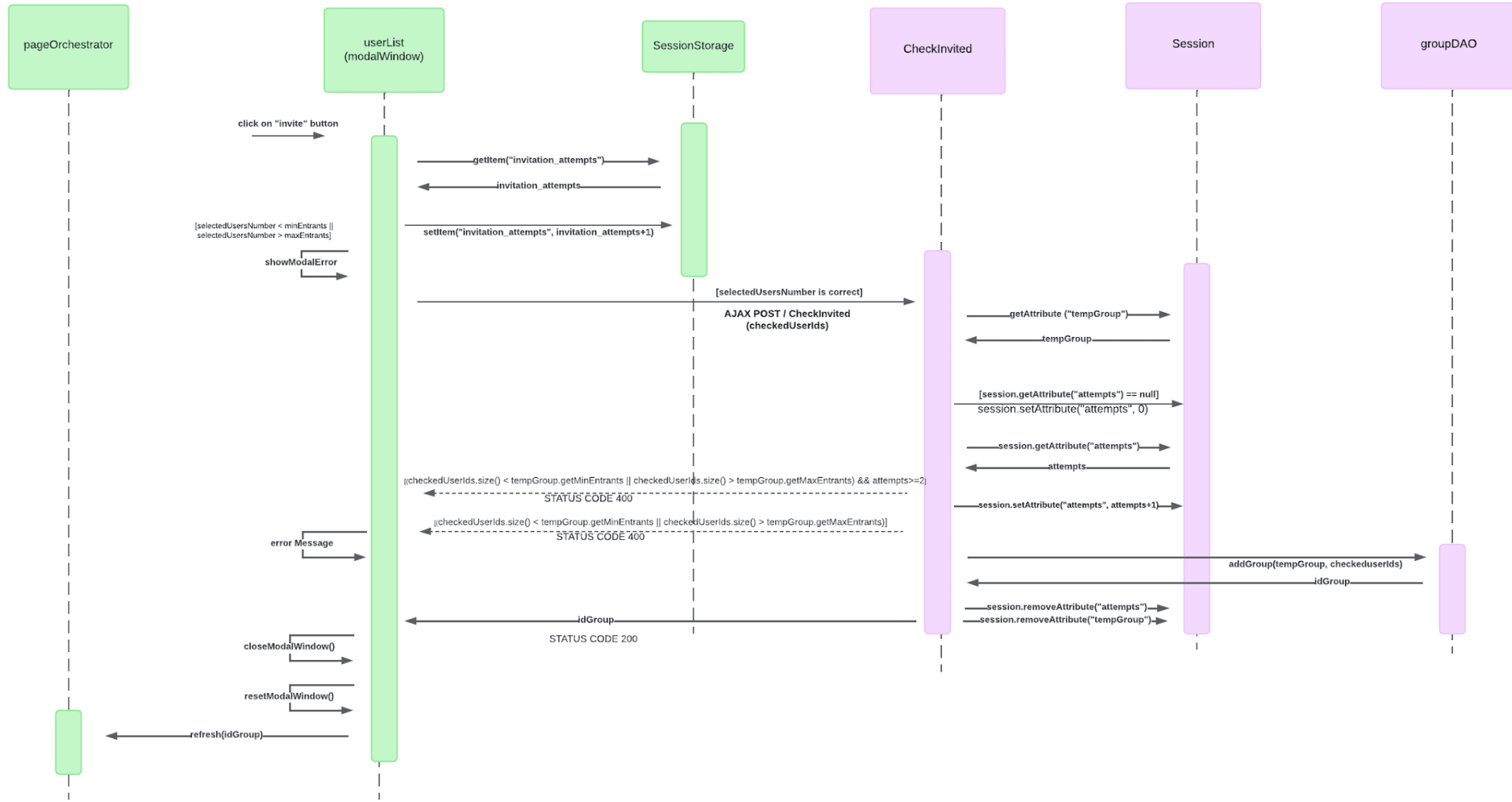


per brevità omettiamo le altre chiamate di reset e show che partono dal pageOrchestrator, che non hanno subito modifiche rispetto ai casi precedenti)

# CREAZIONE DI UN NUOVO GRUPPO (parte 1)



## CREAZIONE DI UN NUOVO GRUPPO (parte 2)



# LOGOUT

