

# DOCUMENTAZIONE PROGETTO

TECNOLOGIE INFORMATICHE PER IL WEB

Valeria Lu

3 – Gruppi di lavoro

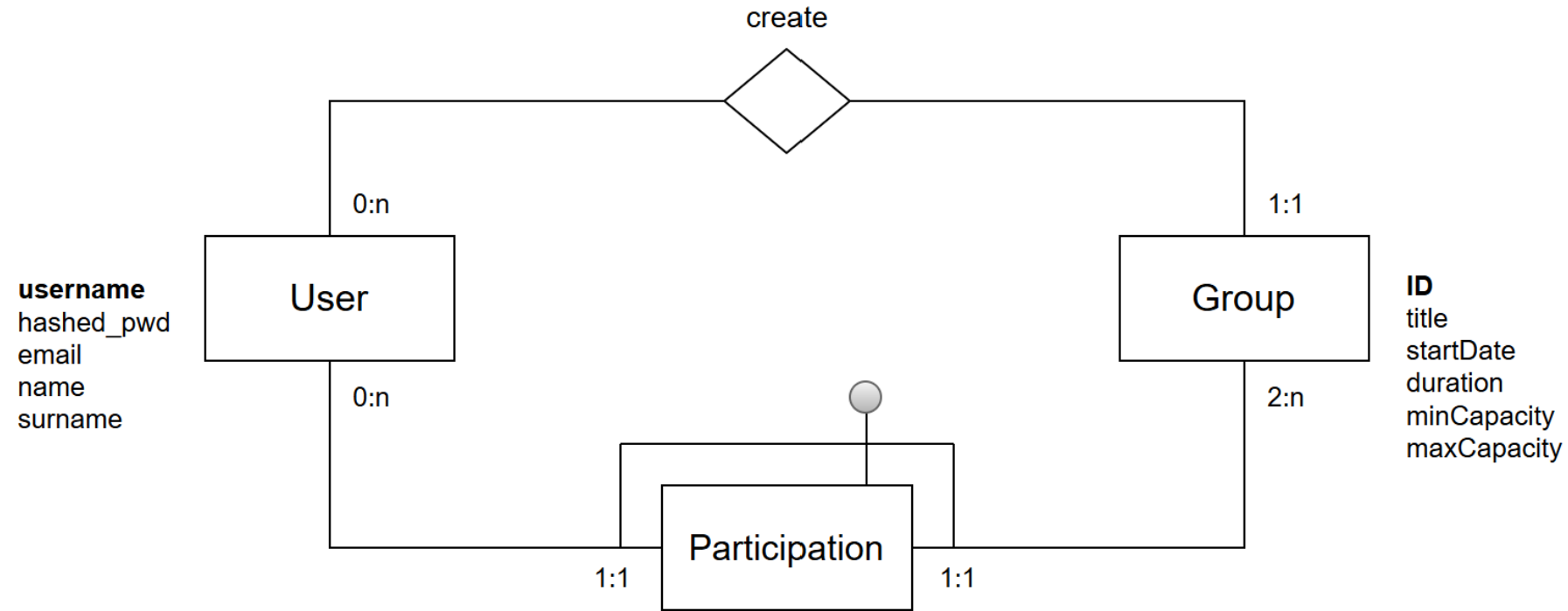
VERSIONE PURE HTML

# Analisi di dati

Un'applicazione web consente la gestione di gruppi di lavoro. 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". 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 delle 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. Quando l'utente inoltra la form di creazione di un nuovo gruppo con il bottone INVIA, appare una pagina ANAGRAFICA con l'elenco degli utenti registrati identificati da nome e cognome e ordinata alfabeticamente per cognome crescente

**Entities**, **attributes**, **relationships**

# Database design



# Local database schema

```
CREATE TABLE `user` (  
  `username` varchar(64) NOT NULL,  
  `hashed_pwd` varchar(64) NOT NULL,  
  `email` varchar(64) NOT NULL,  
  `name` varchar(64) NOT NULL,  
  `surname` varchar(64) NOT NULL,  
  PRIMARY KEY (`username`),  
  UNIQUE KEY  
  `username_UNIQUE` (`username`))
```

```
CREATE TABLE `group` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `title` varchar(128) NOT NULL,  
  `startDate` date NOT NULL,  
  `duration` int NOT NULL,  
  `minCapacity` int NOT NULL,  
  `maxCapacity` int NOT NULL,  
  `creator` varchar(64) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  KEY `creator_idx` (`creator`),  
  CONSTRAINT `creator` FOREIGN KEY  
  (`creator`) REFERENCES `user`  
  (`username`) ON DELETE CASCADE ON  
  UPDATE CASCADE)
```

# Local database schema

```
CREATE TABLE `participation` (  
  `idGroup` int NOT NULL,  
  `participant` varchar(64) NOT NULL,  
  
  PRIMARY KEY (`idGroup`,`participant`),  
  
  KEY `whoParticipates_idx` (`participant`),  
  CONSTRAINT `toWhichGroup` FOREIGN KEY (`idGroup`)  
  REFERENCES `group` (`id`) ON DELETE CASCADE ON  
  UPDATE CASCADE,  
  CONSTRAINT `whoParticipates` FOREIGN KEY  
  (`participant`) REFERENCES `user` (`username`) ON  
  DELETE CASCADE ON UPDATE CASCADE)
```

# Application requirements analysis

Un'applicazione web consente la gestione di gruppi di lavoro. 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". 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 delle 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. Quando l'utente **inoltra la form di creazione** di un nuovo gruppo con il bottone INVIA, appare una **pagina ANAGRAFICA** con **l'elenco degli utenti registrati** identificati da nome e cognome e ordinata alfabeticamente per cognome crescente.

**Pages(views), view components, events, actions**

# Application requirements analysis

L'utente può **scegliere uno o più partecipanti** dall'elenco e **premere il bottone INVITA** per invitarli al gruppo. Se il numero d'invitati è inferiore di X unità rispetto al minimo ammissibile, appare di nuovo la pagina ANAGRAFICA con un messaggio "Troppo pochi utenti selezionati, aggiungerne almeno X". Gli utenti precedentemente selezionati devono rimanere selezionati, ma possono essere sostituiti, in tutto o in parte, da altri utenti. Se il numero d'invitati è superiore di X unità rispetto al massimo ammissibile, appare di nuovo la pagina ANAGRAFICA con un messaggio "Troppi utenti selezionati, eliminarne almeno X". In questo caso, la pagina evidenzia nell'elenco gli utenti scelti in precedenza come preselezionati, in modo che l'utente possa **deselezionarne** alcuni. Se alla pressione del bottone INVITA il numero d'invitati rispetta i vincoli, il gruppo è **memorizzato nella base di dati** e associato agli utenti invitati e l'utente è rimandato alla HOME PAGE.

**Pages(views), view components, events, actions**

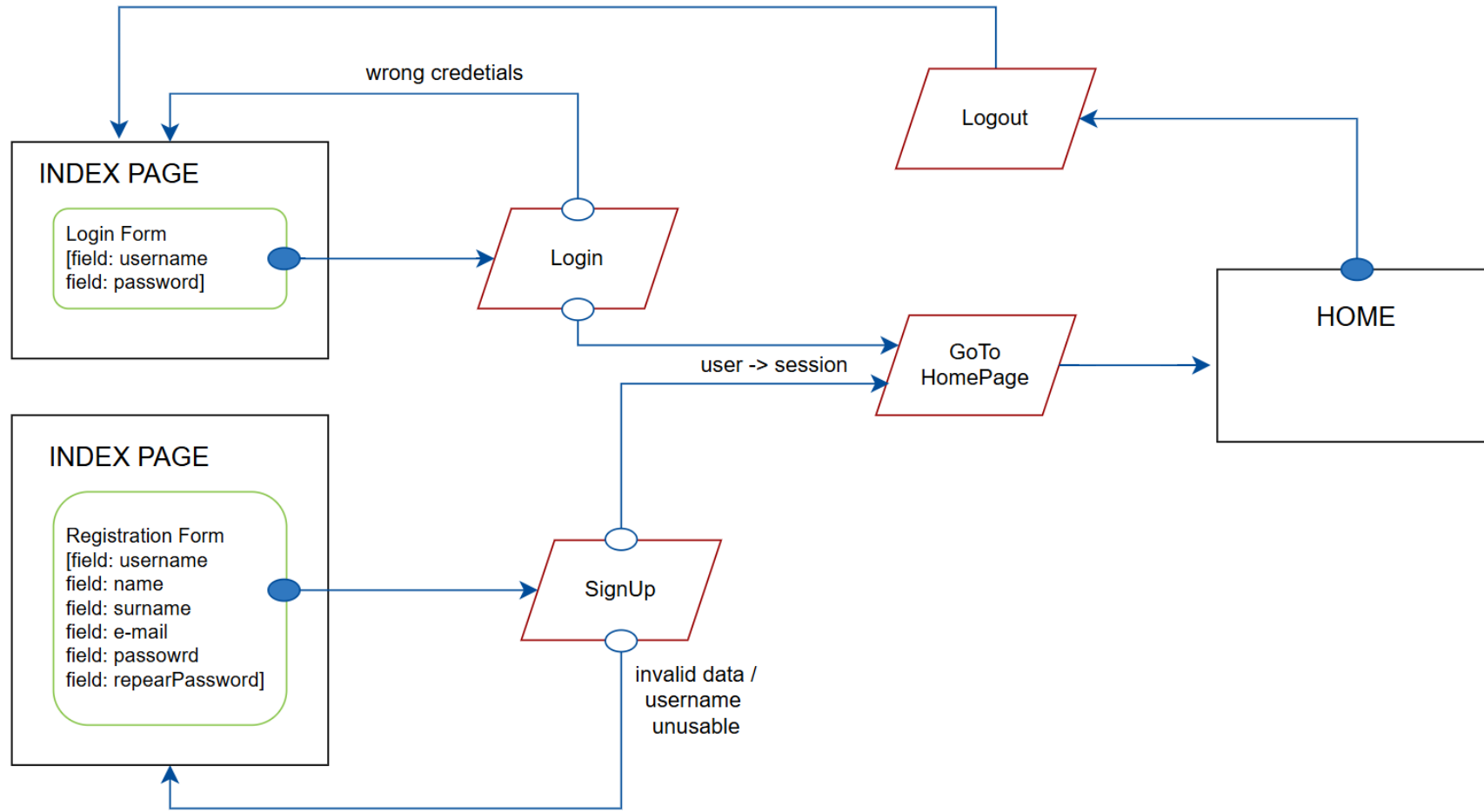


# Application requirements analysis

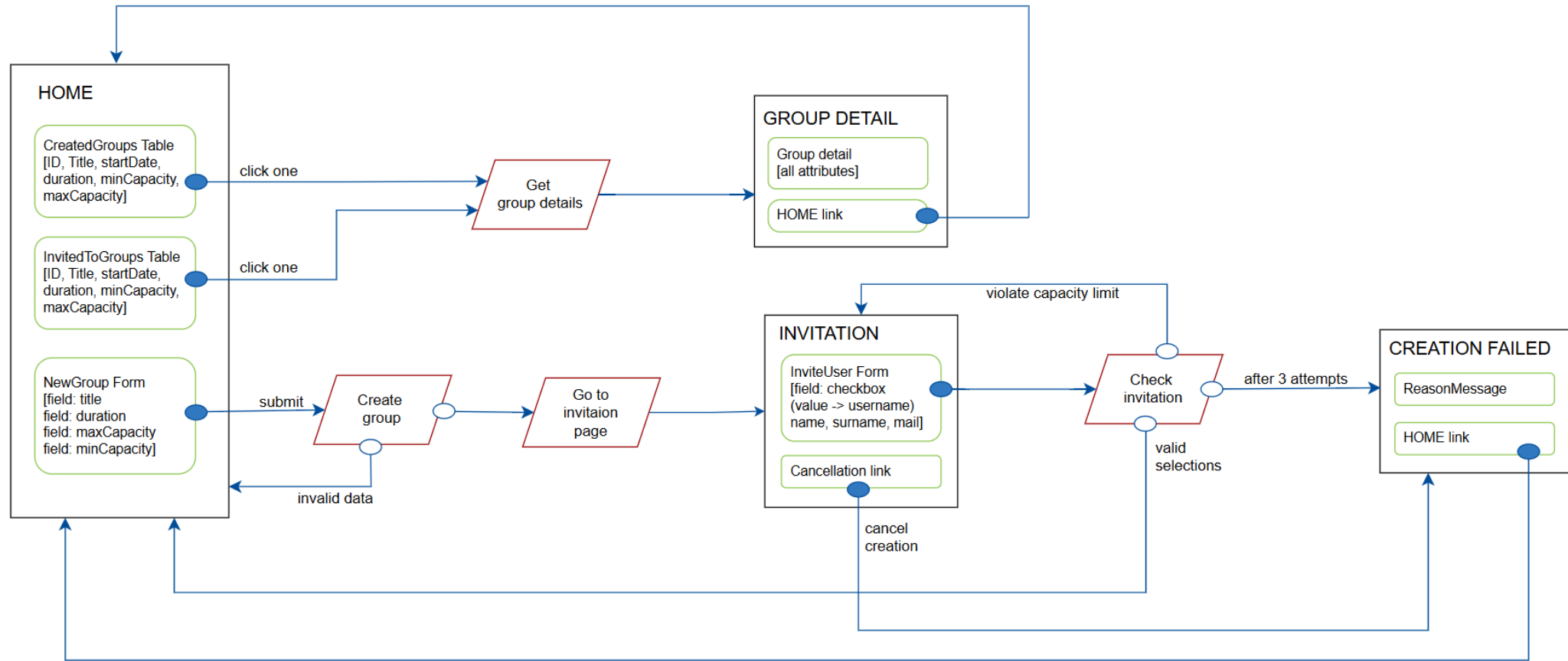
Al **terzo tentativo** scorretto di assegnare un numero errato di invitati a un gruppo appare una **pagina CANCELLAZIONE** con un messaggio “Tre tentativi di definire un gruppo con un numero di partecipanti errato, il gruppo non sarà creato” e un **link** per tornare all’HOME PAGE. In questo caso il gruppo NON è memorizzato nella base di dati. L’applicazione non deve registrare nella base di dati gruppi con numero errato di partecipanti. L’applicazione consente il **logout** dell’utente.

**Pages(views), view components, events, actions**

# Application design



# Application design



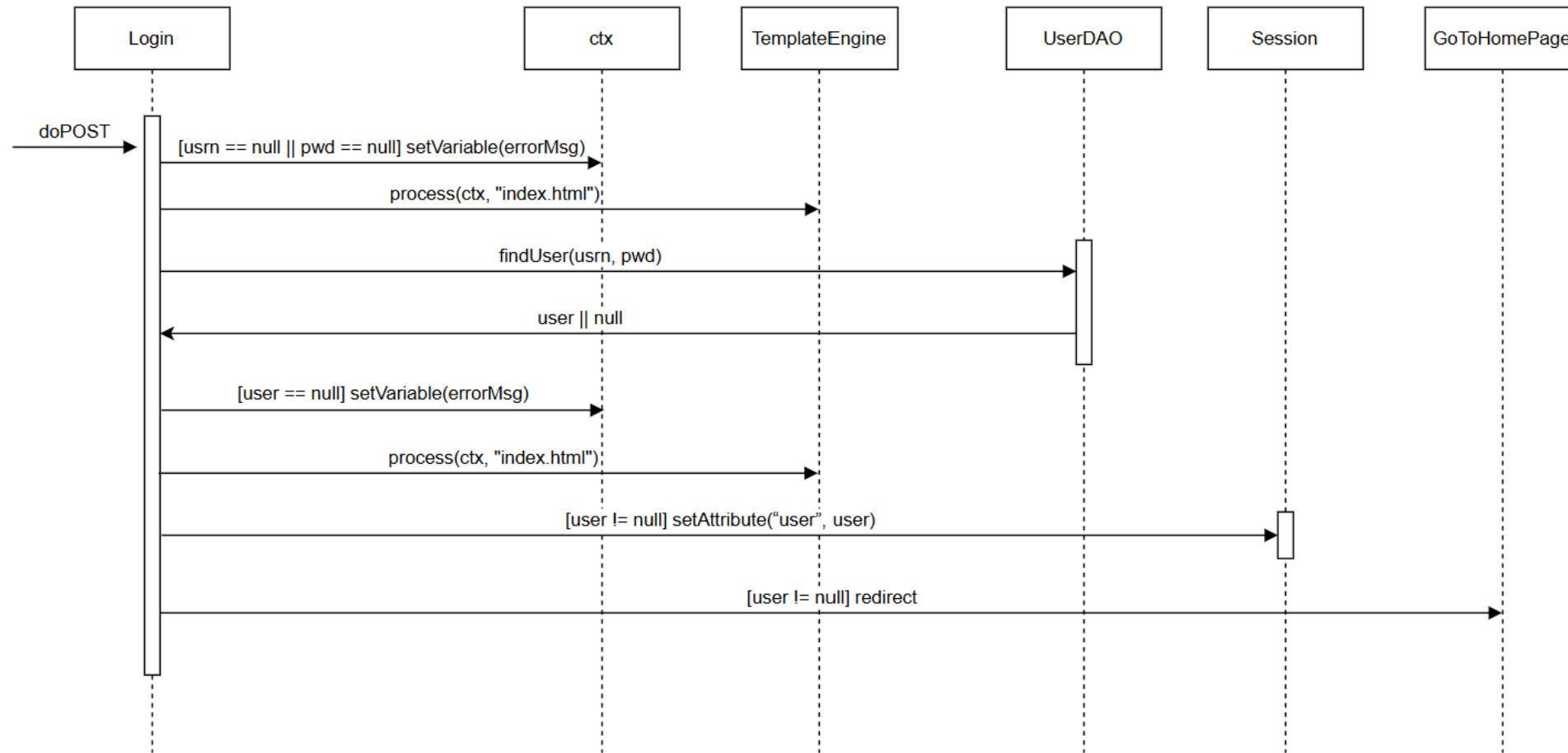
# Componenti

- Model objects (Beans)
  - User
  - Group
  - Participation
- Data Access Objects (Classes)
  - UserDao
    - findUser(usrn, pwd)
    - isUsernameUsable(usrn)
    - registerUser (usern, pwd, email, name, surname)
    - getAllOtherUsers(usrn)
  - GroupDAO
    - createGroup(title, duration, minCapacity, maxCapacity)
    - findGroupsByCreator (creator)
    - findGroupById(idGroup)
  - ParticipationDAO
    - addParticipation(idGroup, usrn)
    - findGroupsByInvitee (usrn)
    - findParticipantsByGroupid(idGroup)
- Controllers (servlets)
  - Login
  - SignUp
  - GoToHomePage
  - GetGroupDetails
  - CreateGroupe
  - GoToInvitationPage
  - CheckInvitations
  - CancelCreation
  - Logout
- Filters
  - AuthenticationCheck
- Utils
  - ConnectionHandler
  - TemplateHandler
  - PathBook

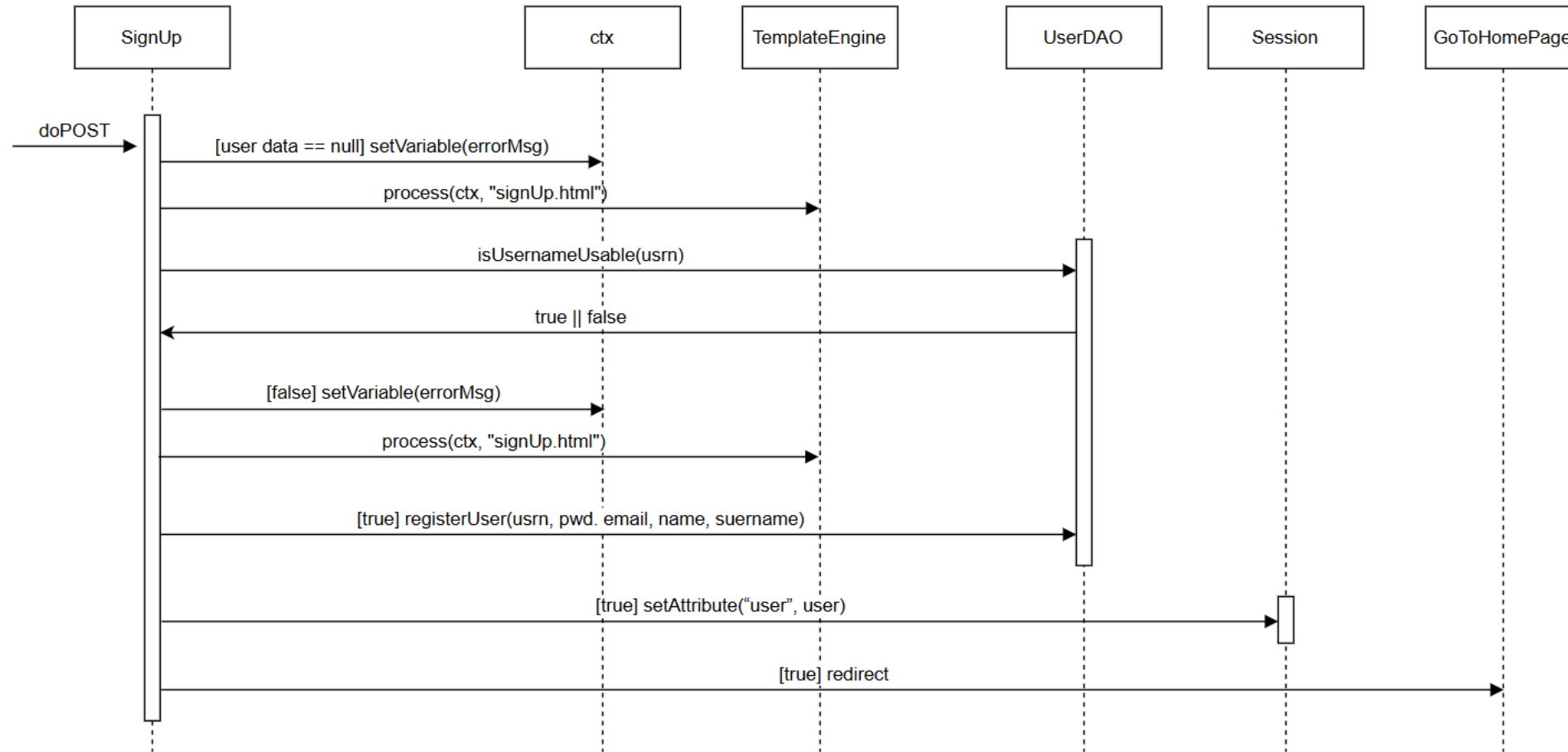
# Componenti

- View (Pages)
  - index.html
  - signUp.html
  - home.html
  - groupDetails.html
  - invitation.html
  - creationFailed.html

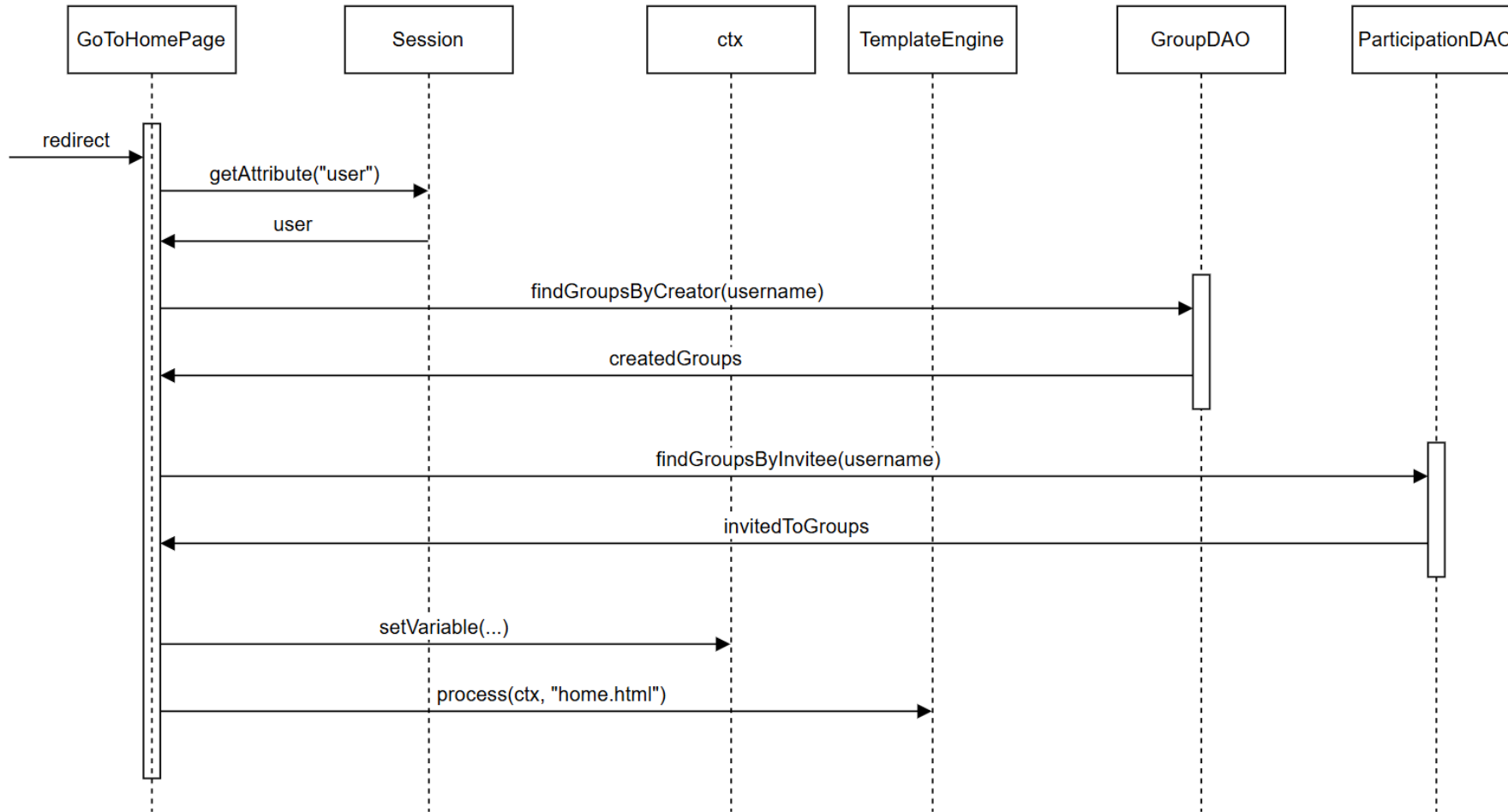
# Event: login



# Event: sign up

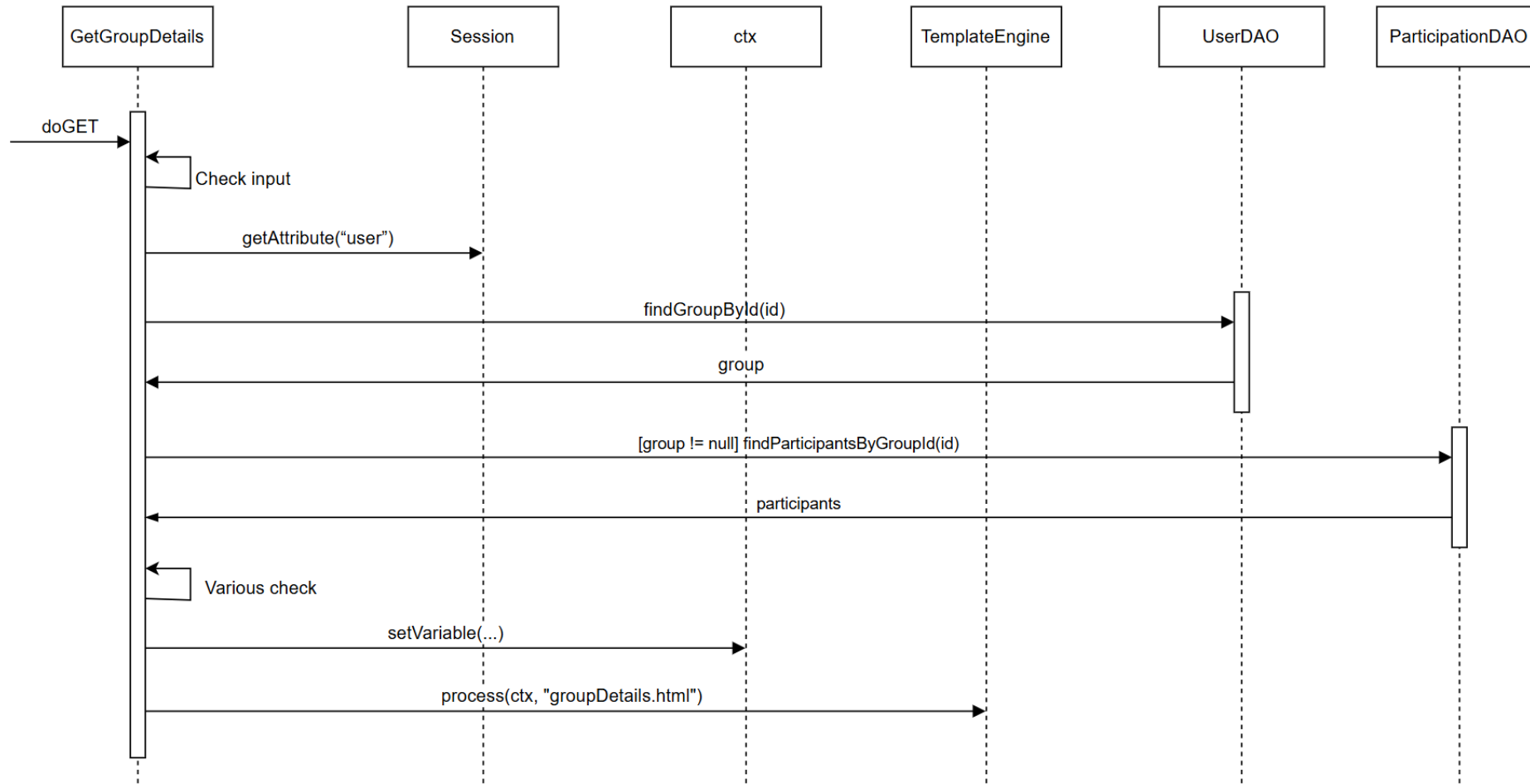


# Event: go to home page



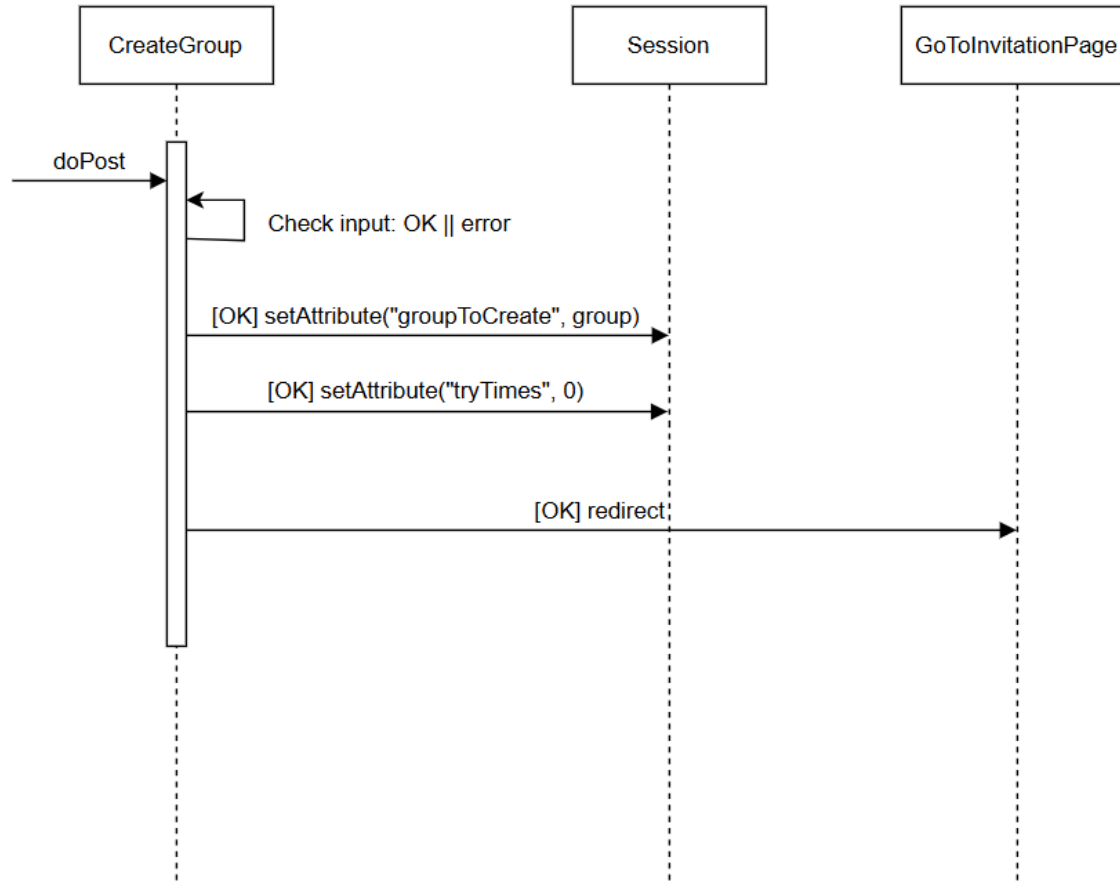


# Event: get group details

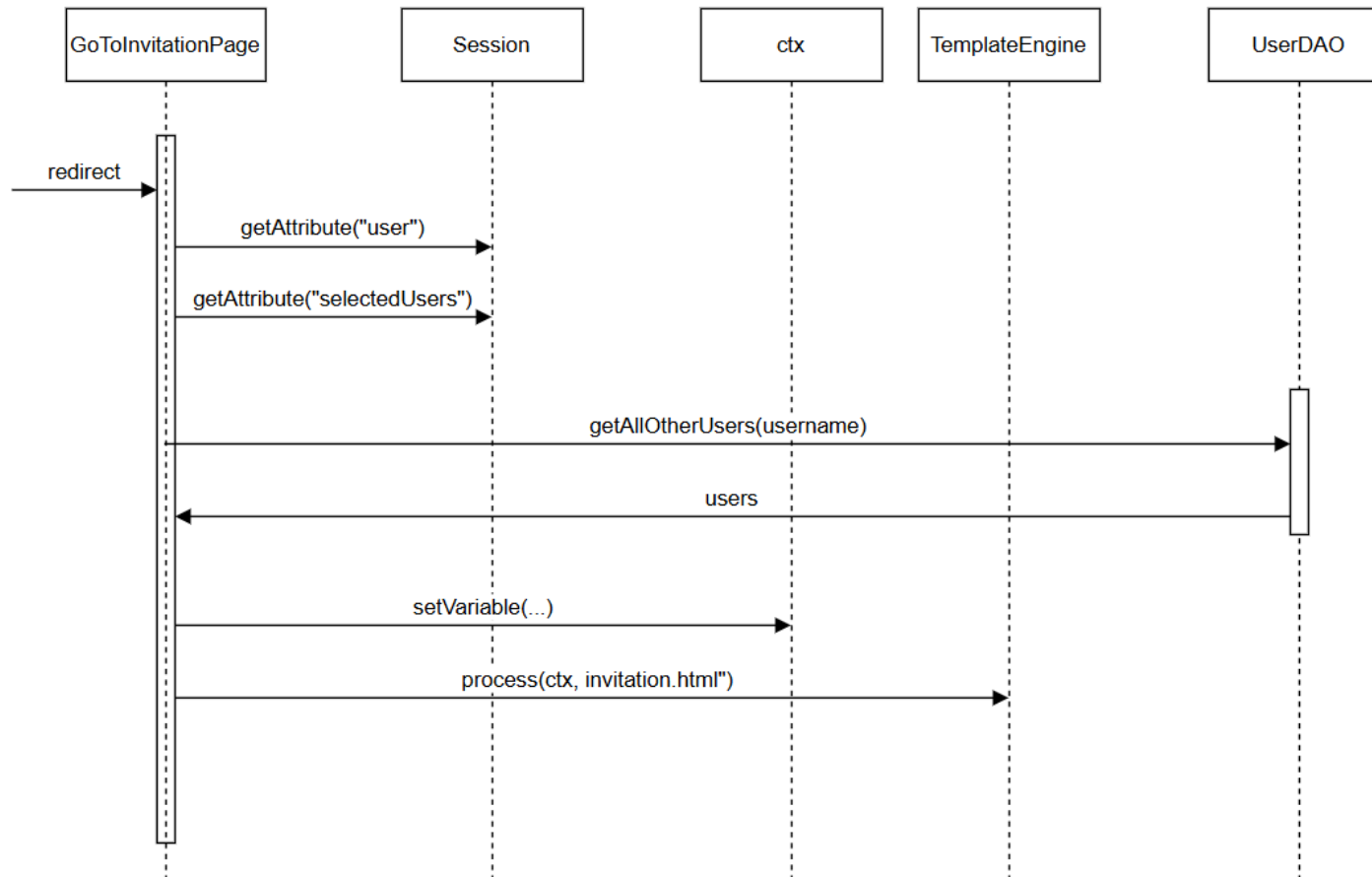


NOTE: in case of null or check ended bad-> sendError

# Event: create group

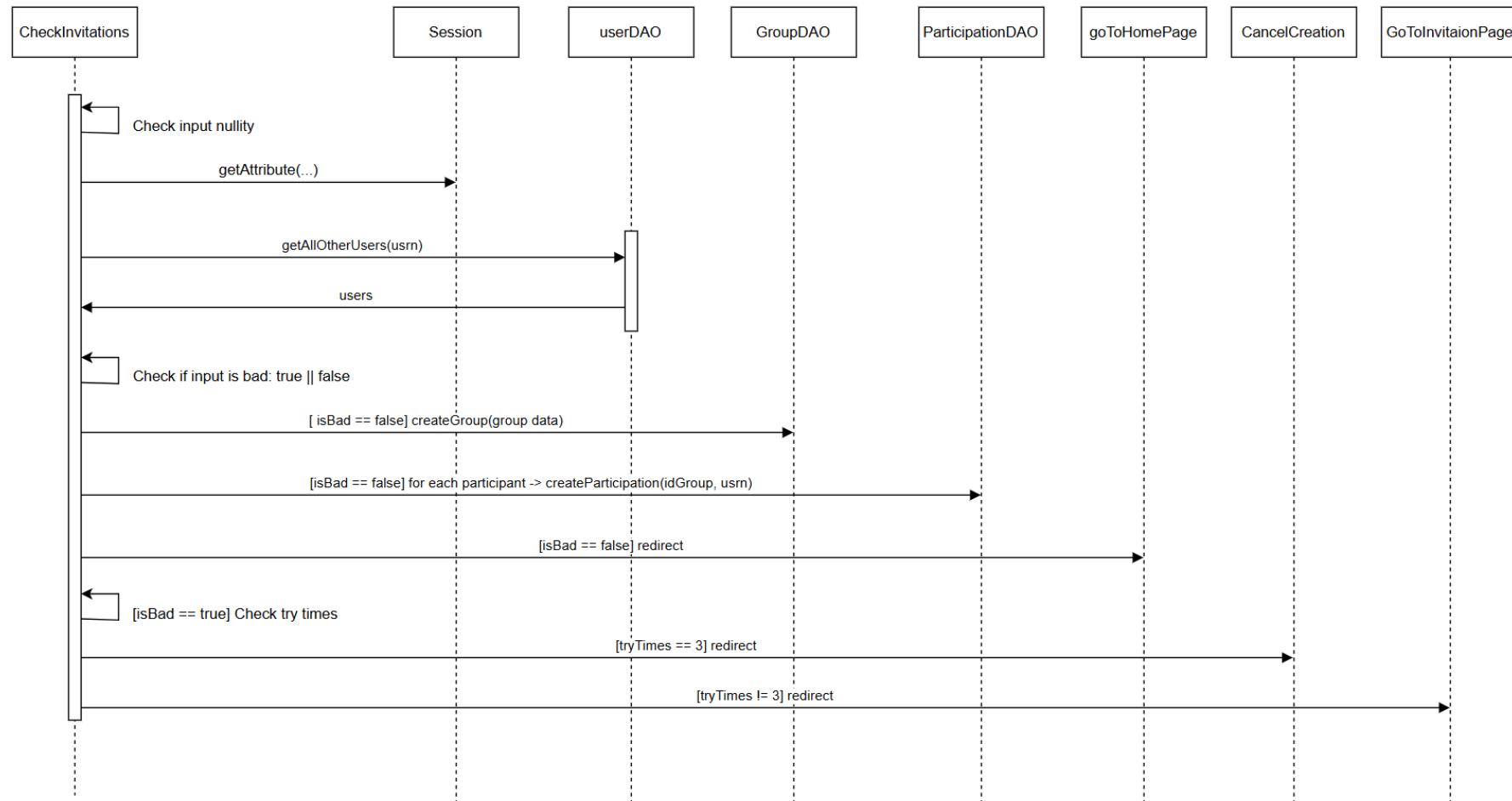


# Event: go to invitation page



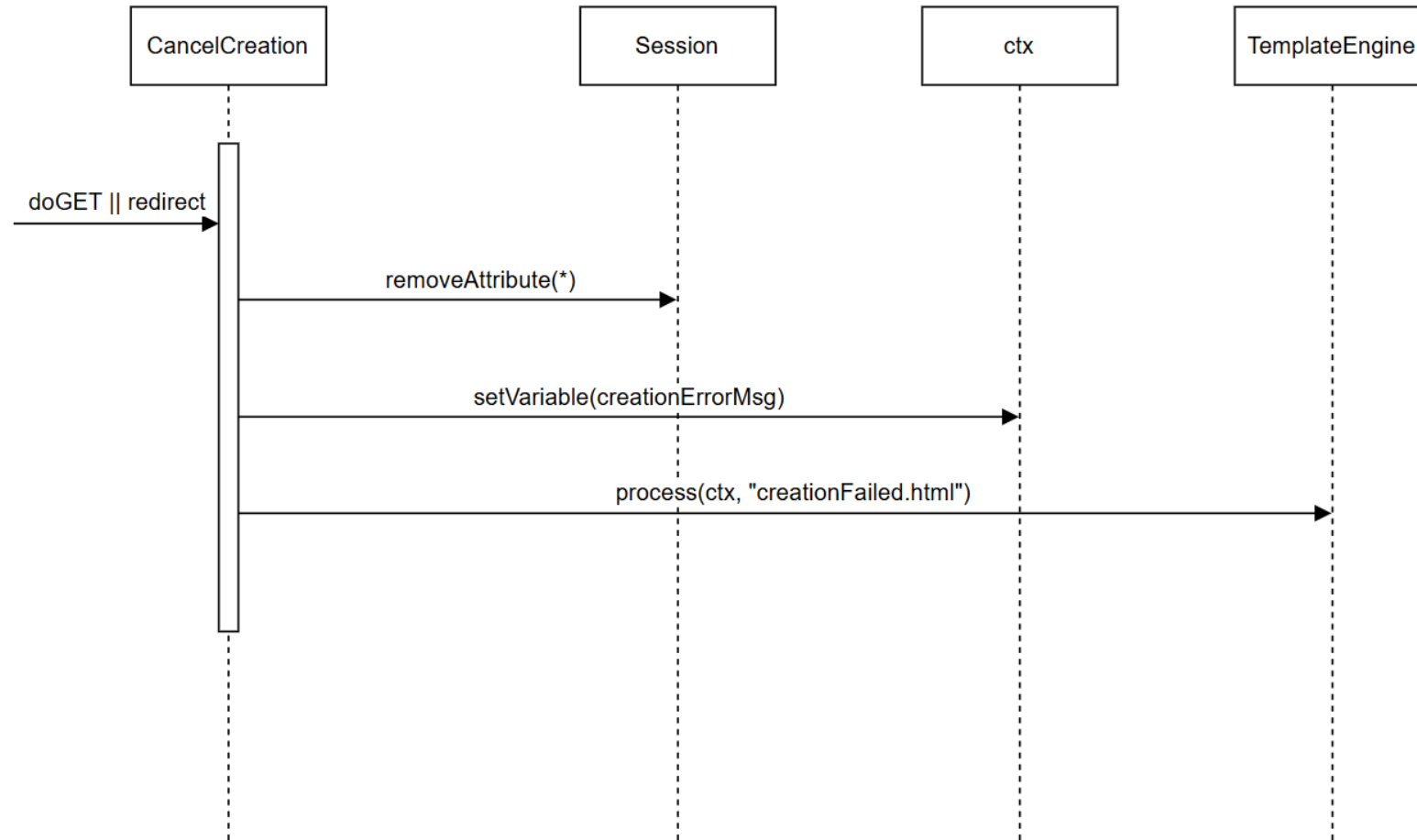
NOTE: if selectedUsers list not exists, it will create new one empty

# Event: check invitations



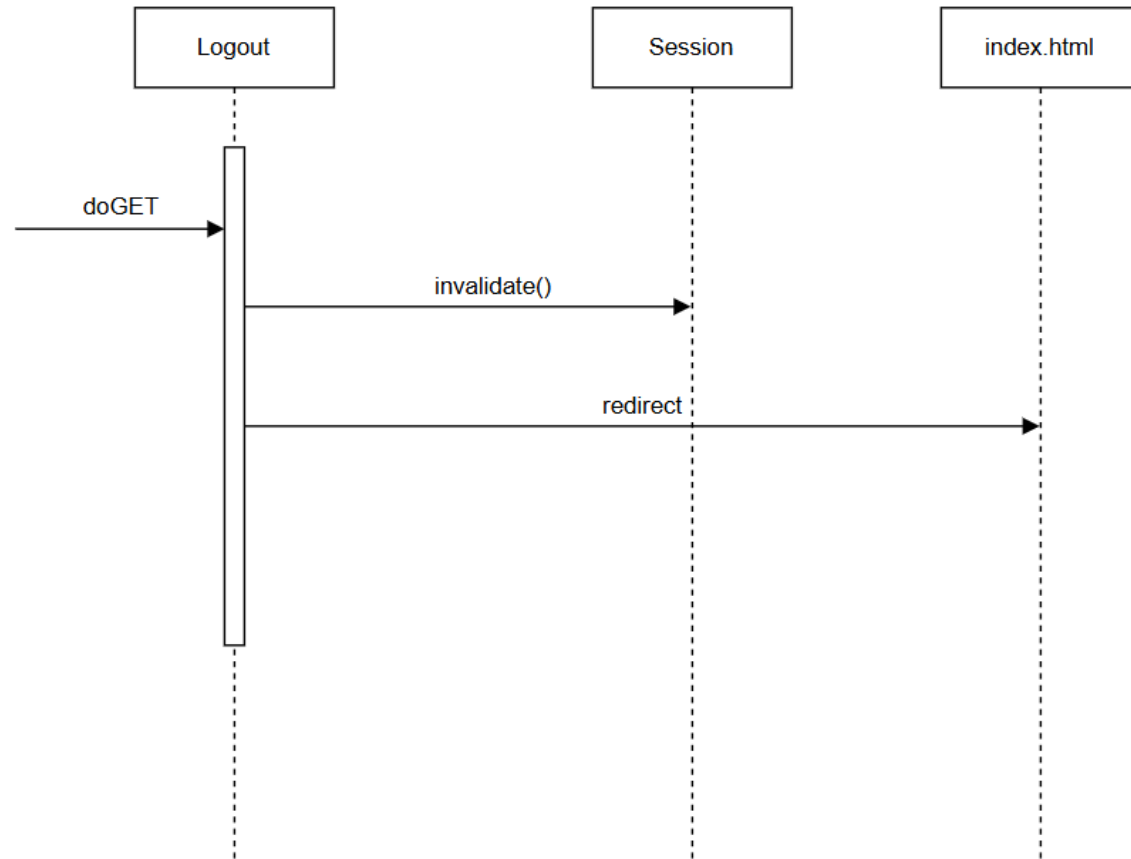
NOTE: input is the selected users

# Event: cancel creation



NOTE: \* -> attributes related to the last group to create

# Event: logout



**VERSIONE RIA**

# Non cambiano rispetto alla versione pure HTML:

- [Analisi di dati](#)
- [Database design](#)
- [Local database schema](#)



# Application requirements analysis

- 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 dell'utente, 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.

**Pages(views), view components, events, actions**

# Application requirements analysis

- Nella pagina **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**.

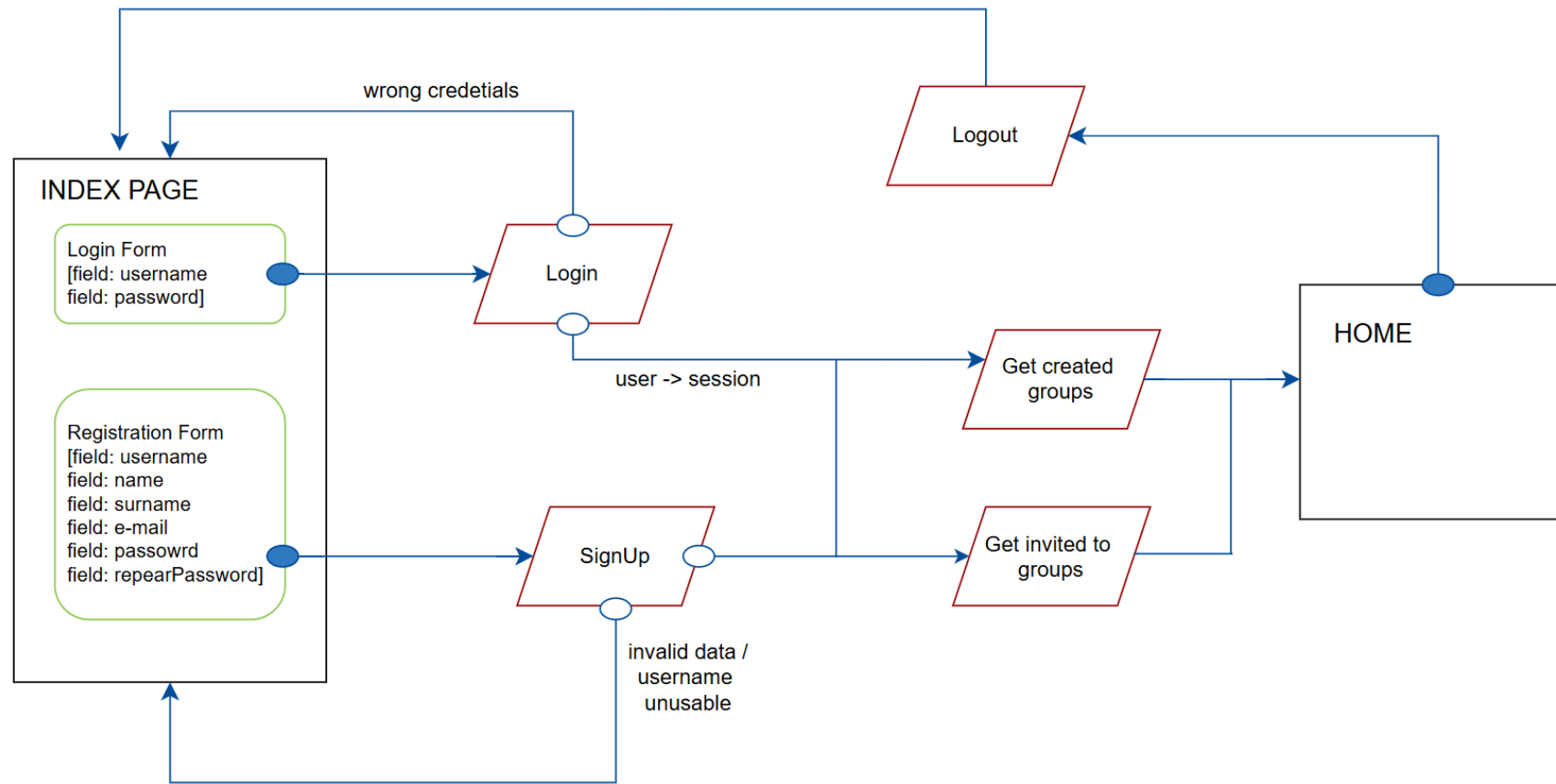
**Pages(views), view components, events, actions**

# Application requirements analysis

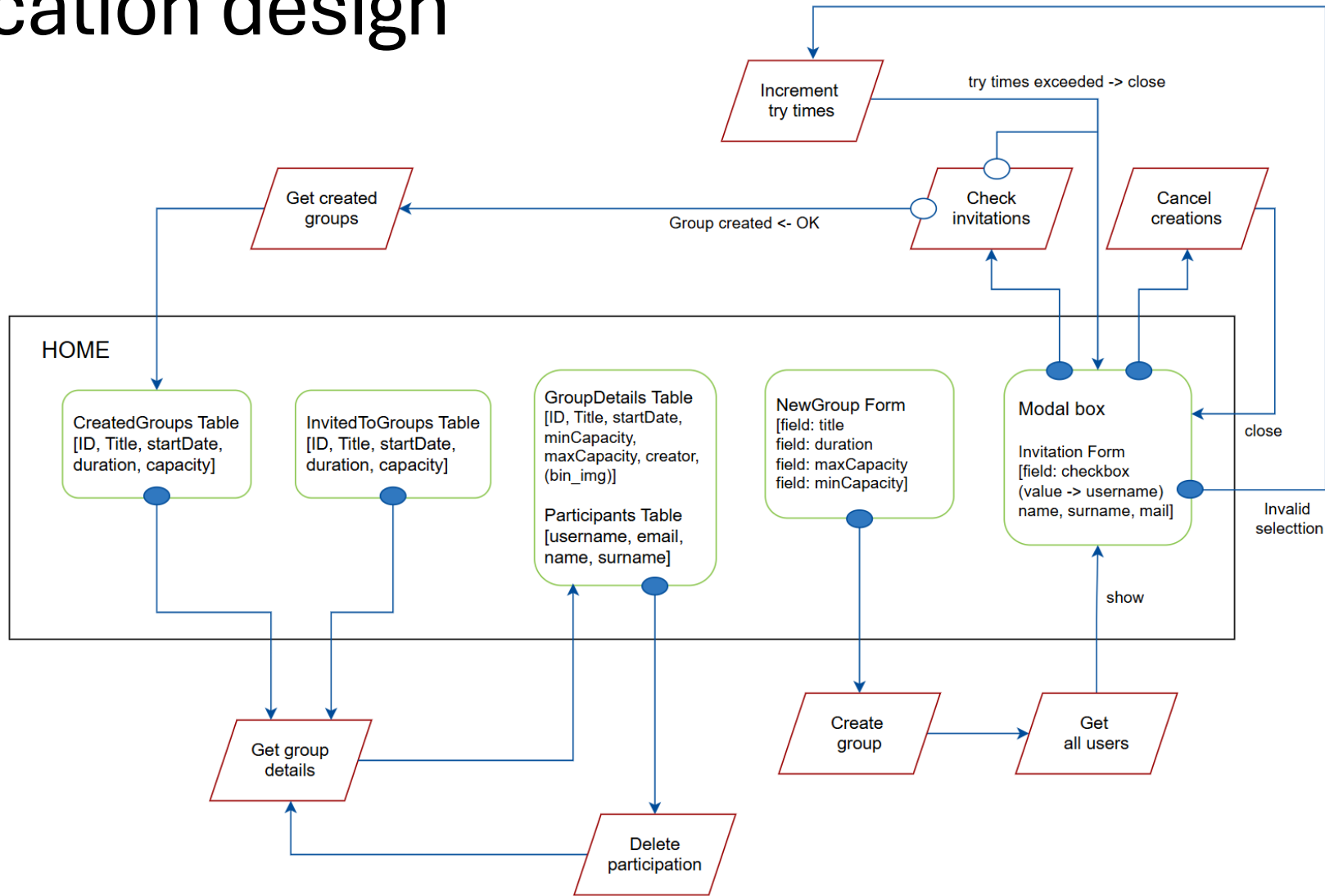
- I **controlli di correttezza** del numero di invitati e del massimo numero di tentativi, con i relativi **messaggi di avvertimento**, devono essere realizzati anche a lato client.
- Lo stato dell'interazione (**numero di tentativi**) deve essere **memorizzato** a lato client.

**Pages(views), view components, events, actions**

# Application design



# Application design



# Componenti

- Model objects (Beans)
  - User
  - Group
  - Participation
- Data Access Objects (Classes)
  - UserDao
    - findUser(usrn, pwd)
    - isUsernameUsable(usrn)
    - registerUser (usern, pwd, email, name, surname)
    - getAllOtherUsers(usrn)
  - GroupDAO
    - createGroup(title, duration, minCapacity, maxCapacity)
    - findGroupsByCreator (creator)
    - findGroupById(idGroup)
  - ParticipationDAO
    - addParticipation(idGroup, usrn)
    - deleteParticipation(idGroup, usrn)
    - findGroupsByInvitee (usrn)
    - findParticipantsByGroupld(idGroup)
- Controllers (servlets)
  - Login
  - SignUp
  - GetCreatedGroups
  - GetInvitedToGroups
  - GetGroupDetails
  - DeleteParticipation
  - CreateGroupe
  - GetAllUsers
  - CheckInvitations
  - IncrementTryTimes
  - CancelCreation
  - Logout
- Filters
  - AuthenticationCheck
- Utils
  - ConnectionHandler

# Componenti

- View (Pages)
  - index.html
    - login form
    - registration form
  - home.html
    - GroupsList → createdGroups, invitedToGroups
    - GroupDetails → groupDetails
    - CreationForm → creation\_form
    - SelectionModal → selection\_modal

# Eventi e azioni - index

Client Side		Server side	
Evento	Azione	Evento	Azione
Click titolo «Login»	Mostra form di login		
Click titolo «Sign up»	Mostra form di registrazione		
Inoltare login form	Controllo dati	POST (user credentials →)	Controllo credenziali
Inoltare sign up form	Controllo dati Controllo sintassi di email, l'uguaglianza tra due campi password	POST (user data →)	Controllo validità Registrazione



# Eventi e azioni – home

Client Side		Server side	
Evento	Azione	Evento	Azione
Load	Inizializza e reset dei vari componenti della pagina	GET (← createdGroups) GET (← invitedToGroups)	Estrazioni gruppi da DB
Click dettaglio simbolo di un gruppo	Mostra dettagli del gruppo con i rispettivi partecipanti, simbolo cestino se l'utente è creatore	GET (group id →) (← participants)	Verificare l'esistenza del gruppo, verificare se l'utente appartiene al gruppo , estrazione partecipanti dal DB
Trascinare un partecipante sul simbolo cestino	Mostra messaggio di conferma + refresh del dettaglio gruppo oppure Mostra messaggio di diniego	GET (group id, username →)	Controlla se la cancellazione viola limiti di capienza Elimina la partecipazione dal DB
Inoltare form di creazione	Controllo dati, apertura finestra modale	POST (group data →)	Controllo dati, salvataggio in sessione gruppo da creare

# Eventi e azioni – home

Client Side		Server side	
Evento	Azione	Evento	Azione
Apertura finestra modale	Caricamento utenti iscritti alla piattaforma	GET (← users)	Estrazione utenti dal DB
Finestra modale → Inoltare form di invitazione	Controllo numero di utenti selezionati e numero di tentativi Incrementa numero di tentativi oppure creazione con successo	GET [richiesta di increm.] or POST(selectedUsers →)	Incremento n. tentativi  Controllo numero e validità partecipanti, controllo tentativi
Finestra modale → Cancella creazione	Chiusura finestra modale	GET [richiesta di cancel.]	Rimozione dei dati relativi al gruppo in corso di creazione salvati in sessione
Creazione con successo	Chiusura finestra modale Refresh dei gruppi creati dall'utente		
Numero tentativi superati	Chiusura finestra modale		

# Sequence diagram: caricamento home

