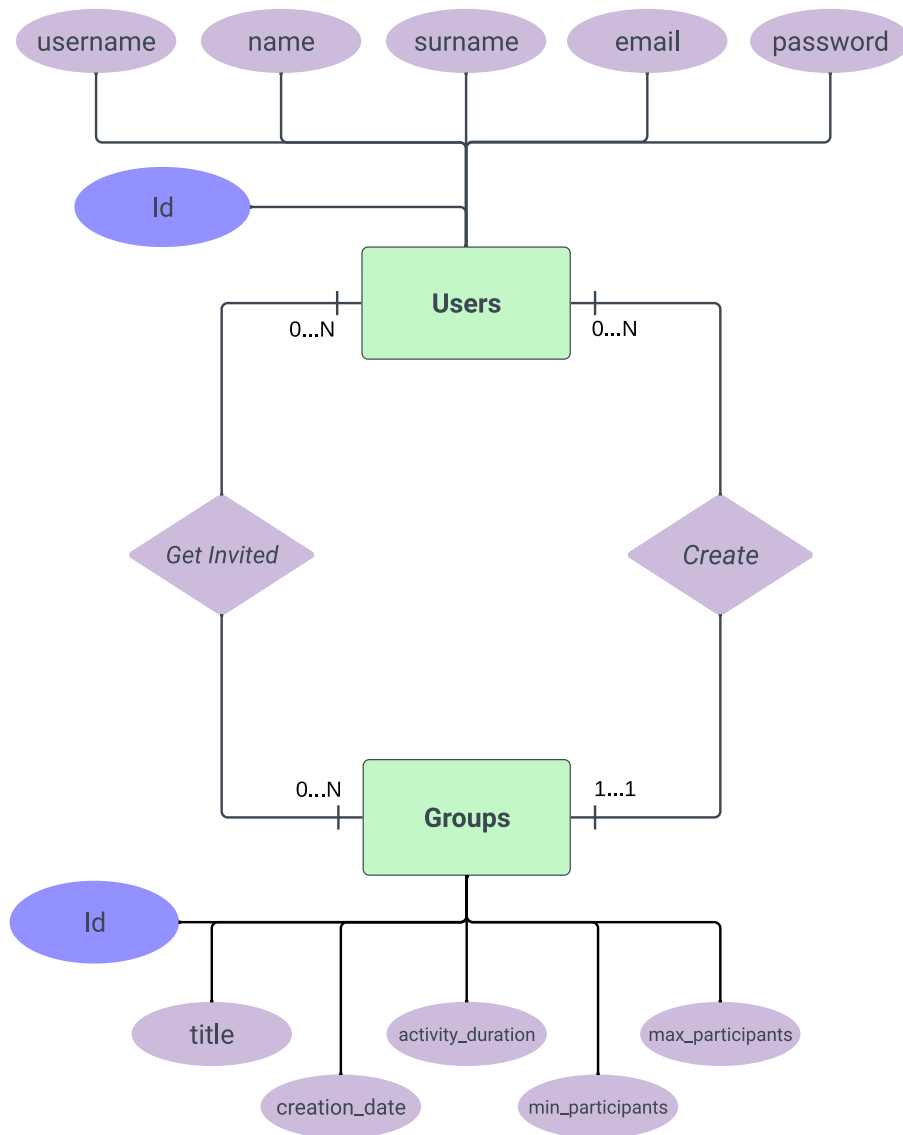


# Project in Web Information Technologies

PURE HTML VERSION  
TRACK 3: WORK GROUPS  
GIULIANO CRESCIMBENI  
MAURO LEPORACE

# Database Design



■ - Entities

■ - Primary Key

■ - Attributes

# Database Schema

```
CREATE TABLE `users` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `username` varchar(45) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `surname` varchar(45) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  `password` varchar(32) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  UNIQUE KEY `username_UNIQUE` (`username`),  
  UNIQUE KEY `email_UNIQUE` (`email`)  
)
```

```
CREATE TABLE `groups` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `title` varchar(45) NOT NULL,  
  `creation_date` timestamp(1) NOT NULL DEFAULT CURRENT_TIMESTAMP(1),  
  `activity_duration` int NOT NULL,  
  `min_participants` int NOT NULL,  
  `max_participants` int NOT NULL,  
  `creator_id` int NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  UNIQUE KEY `title_UNIQUE` (`title`),  
  KEY `group_creator_idx` (`creator_id`),  
  CONSTRAINT `group_creator` FOREIGN KEY (`creator_id`) REFERENCES `users` (`id`)  
)
```

**Get invited -> groups\_invitations**

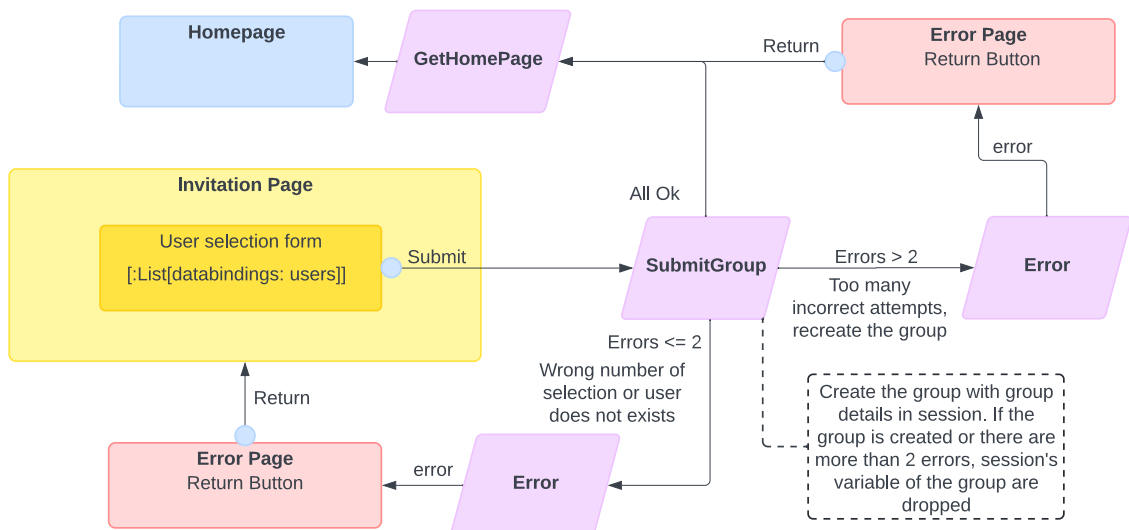
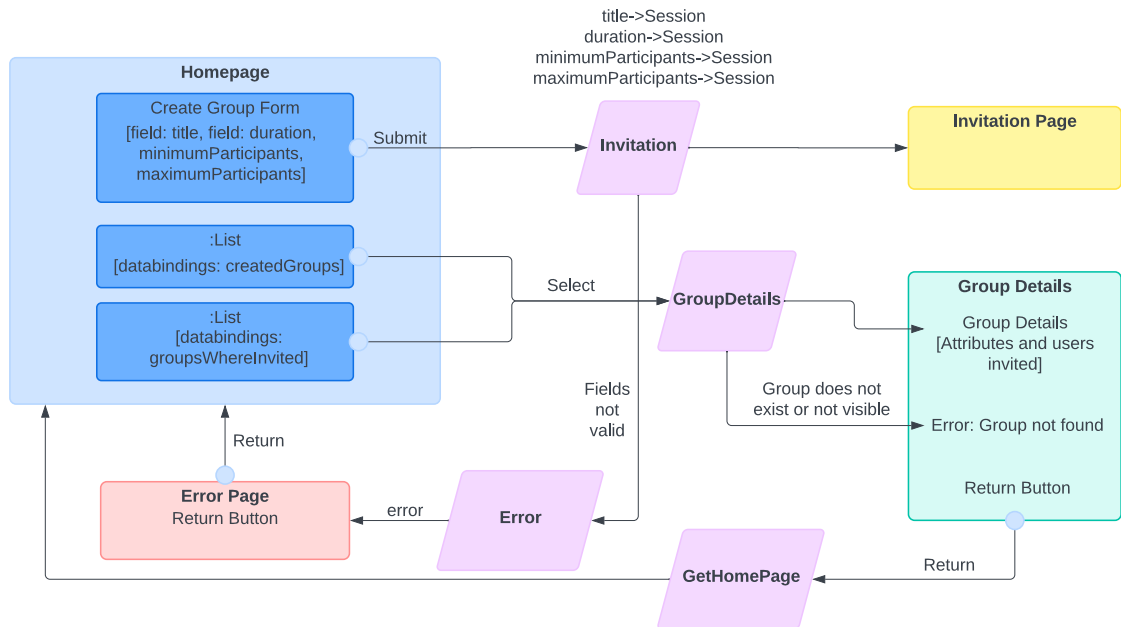
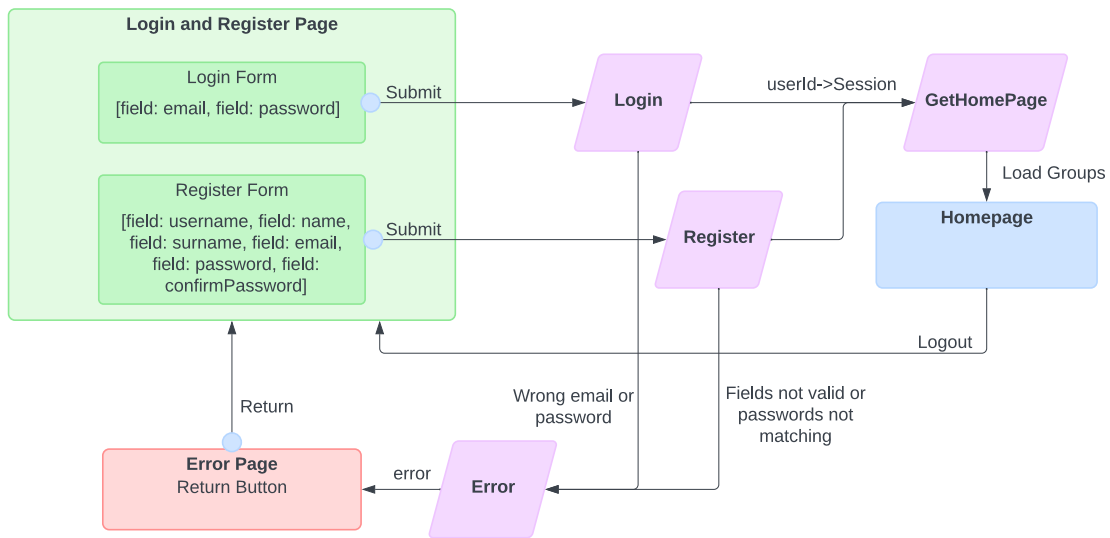
```
CREATE TABLE `groups_invitations` (  
  `id_user` int NOT NULL,  
  `id_group` int NOT NULL,  
  KEY `id_idx` (`id_user`),  
  KEY `id_group_idx` (`id_group`),  
  CONSTRAINT `id_group` FOREIGN KEY (`id_group`) REFERENCES `groups` (`id`) ON DELETE  
  CASCADE,  
  CONSTRAINT `id_user` FOREIGN KEY (`id_user`) REFERENCES `users` (`id`) ON DELETE  
  CASCADE  
)
```

# Specifications

## Implemented Pages:

- [Login e Register](#)
  - [Homepage](#)
  - [Invitation Page](#)
  - [Group Details](#)
  - [Error Page](#)
- [Register](#) Validates form fields and checks for unique usernames.
  - [Login](#) Verifies email and hashed password (SHA-256).
  - The [Homepage](#) presents a form for inserting a new group with field validation controls. The duration in days cannot be less than "1", the minimum number of participants must be at least equal to the number of registered users on the website, and it must be less than or equal to the maximum number of participants.
  - The [Invitation Page](#) checks the number of selected users, which must be between the minimum and maximum number of participants, and also verifies the validity of the users chosen from the form.
  - The [Group Details](#) displays the details of a group. A check is performed to verify that the user has the permissions to view the group details (viewing is only allowed if the user is the creator of the group or is among the invited members).
  - The [Error Page](#) Shows errors for invalid inputs or internal issues.
  - All form fields are mandatory.

# Application Design



# Components

## Model (Beans):

- Group
- User

## Data Access Objects (DAO):

### • UserDAO

- findUserById(id)
- findUserByUsername(username)
- checkCredentials(email, password)
- createUser(username, name, surname, email, password)
- findAllUsers()

### • GroupDAO

- findGroupById(id)
- findGroupByCreatorId(creatorId)
- getGroupsWhereInvited(id)
- getInvitedUsers(id)
- createGroup(title, activityDuration, minParticipants, maxParticipants, creatorId)
- inviteInGroup(userId, groupId)

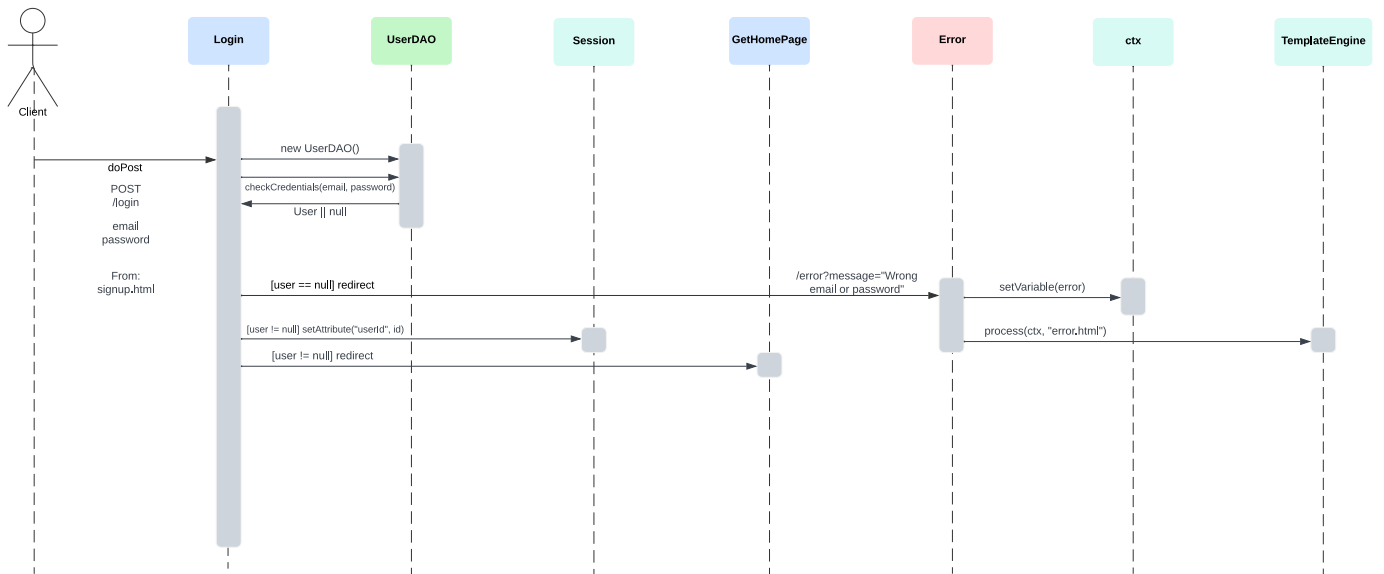
## Controllers:

- Login
- Register
- Logout
- GetHomePage
- CreateGroup
- Invitation
- SubmitGroup
- GroupDetails
- Error
- RedirectionManager (Singleton class for utility)

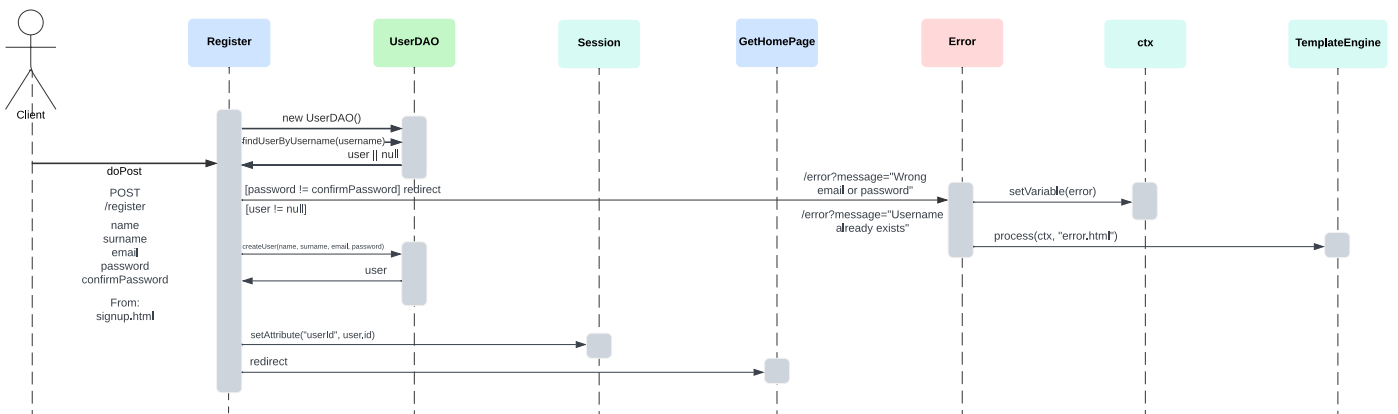
## Views:

- signin
- homepage
- invitation
- groupdetails
- error

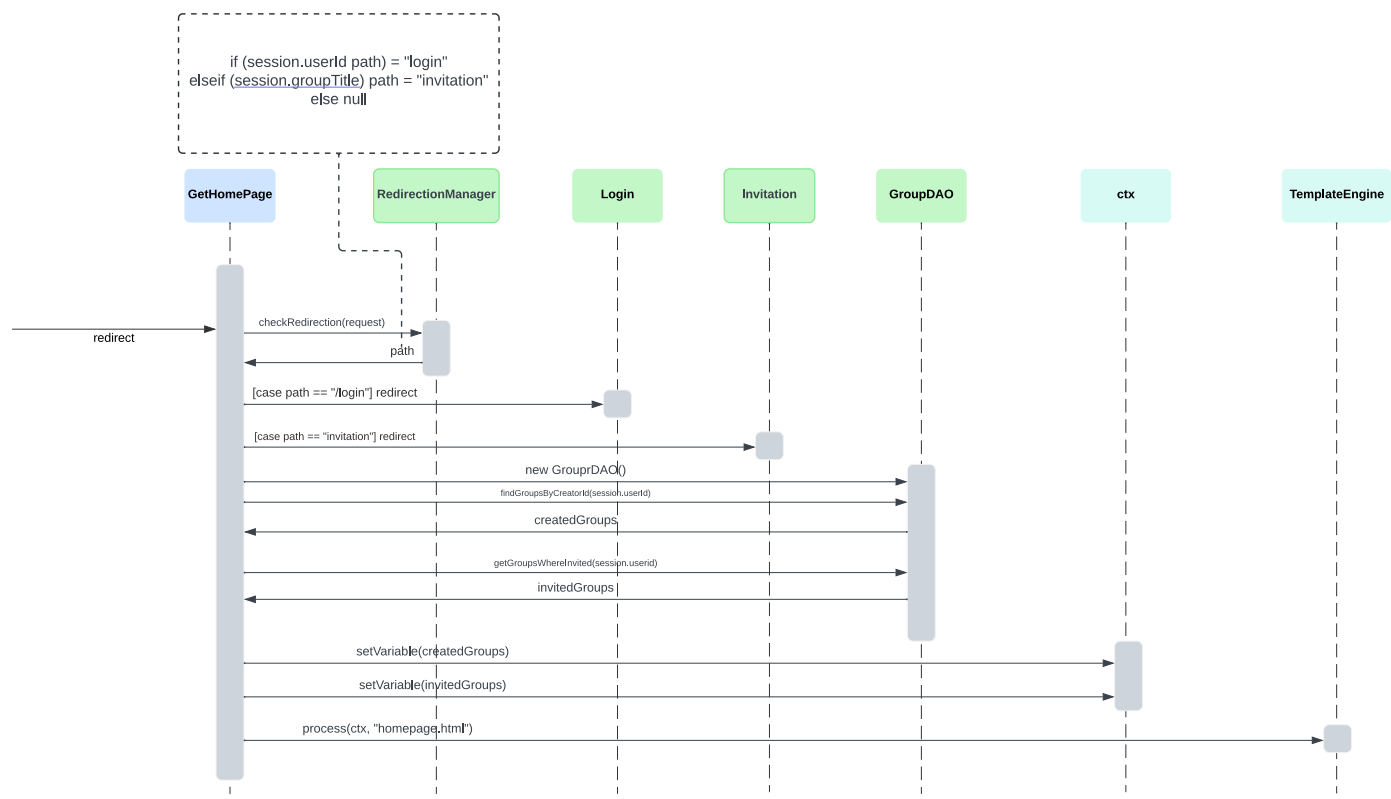
## Event: Login



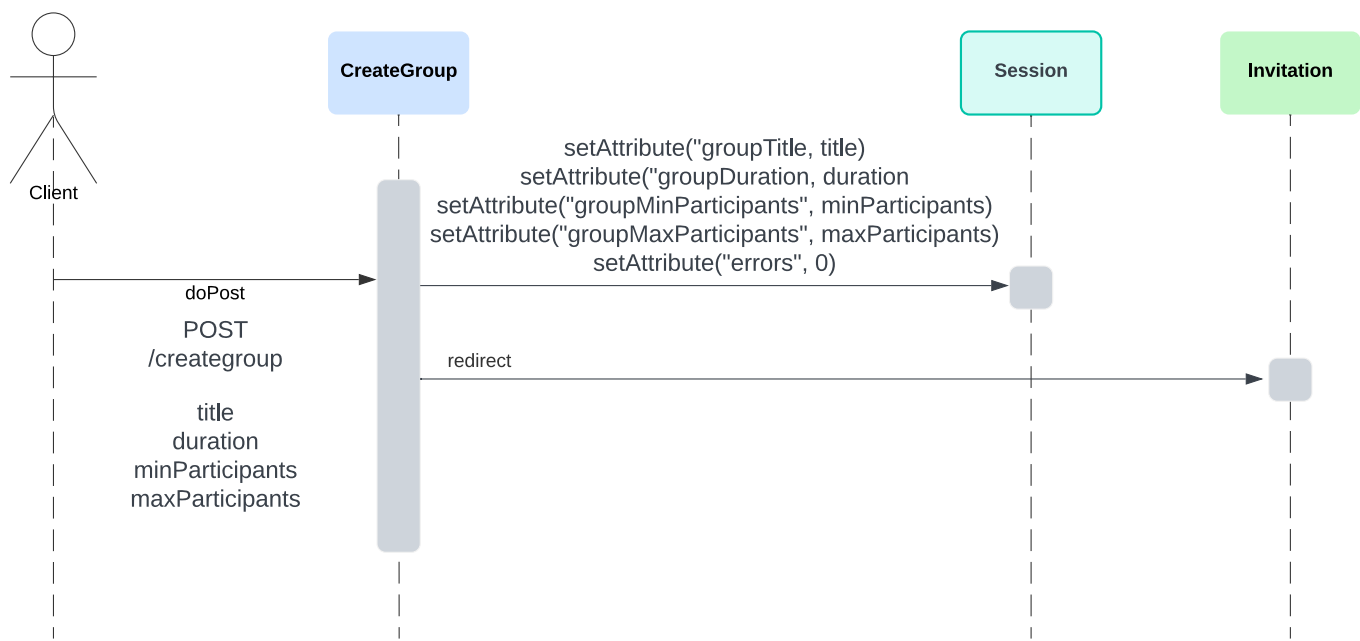
## Event: Register



# Event: Get Homepage

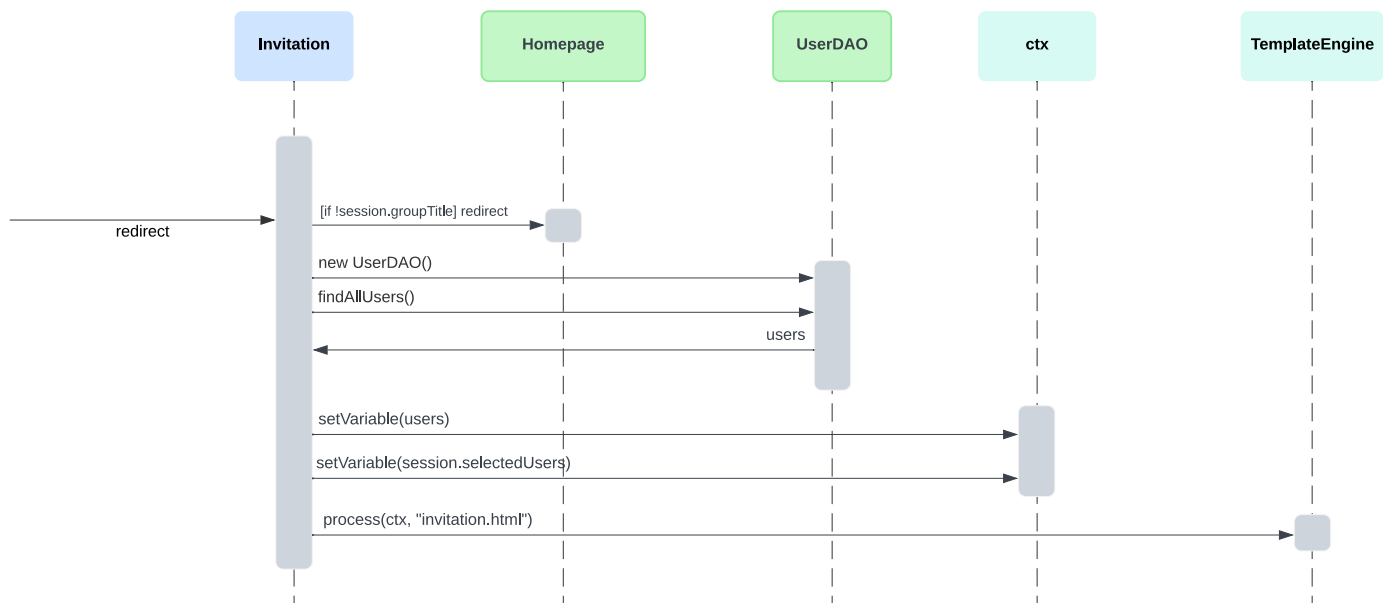


# Event: Create Group

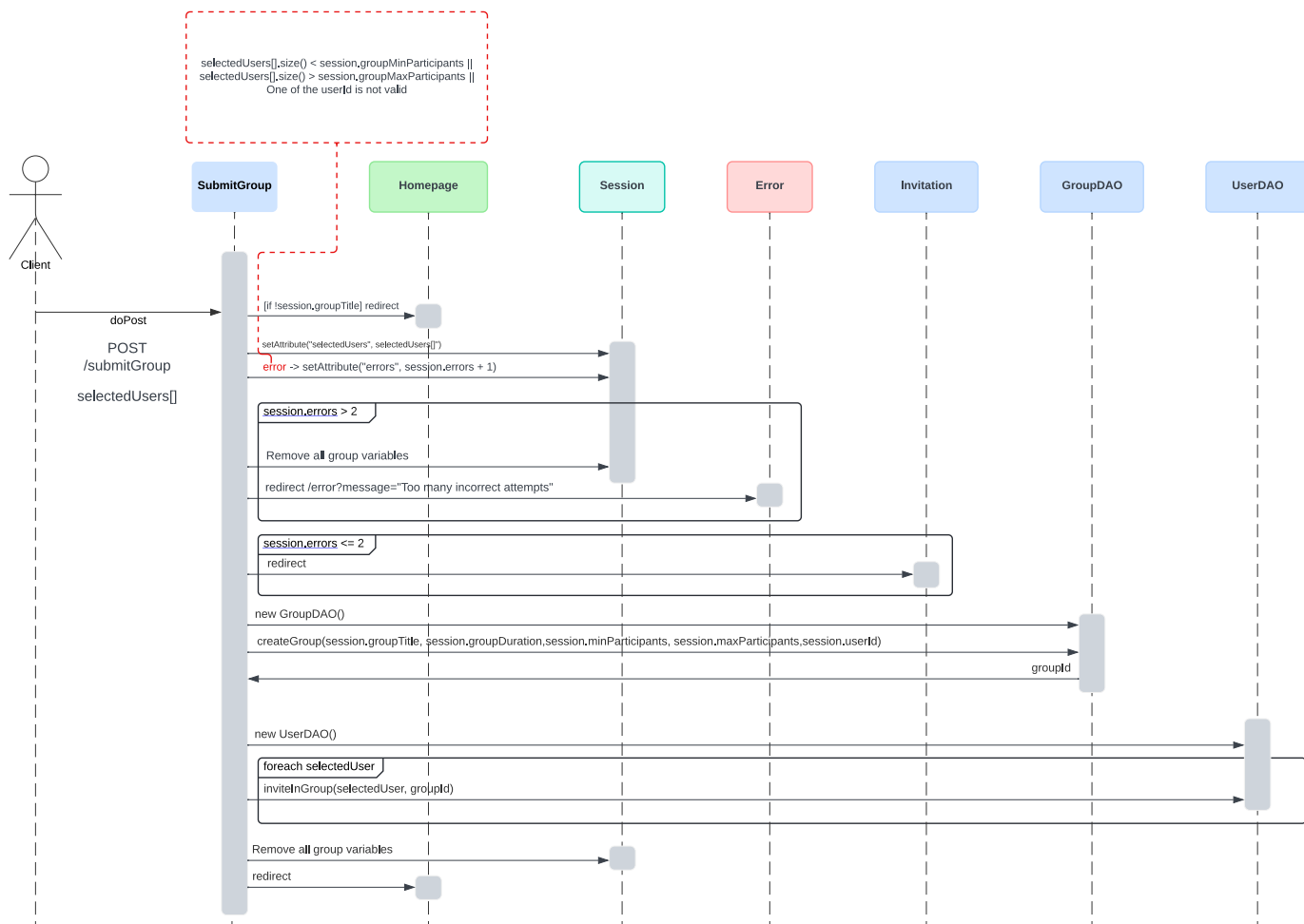




# Event: Invite Users



# Event: Submit Group



# Event: Group Details

