A.Y.
2023
2024

Project in Web Information Technologies

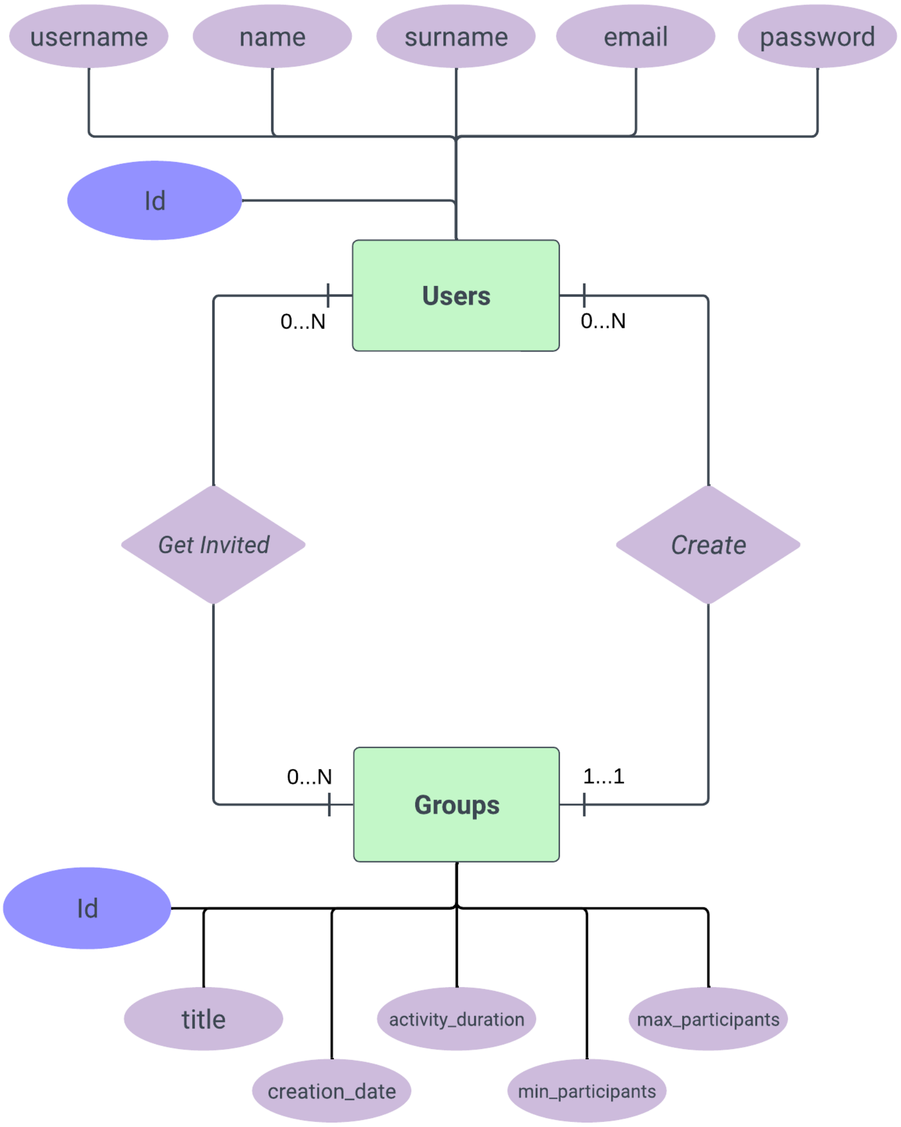
RIA VERSION

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 for Client-Side Scripting

Implemented Pages:

- Login e Register
- Homepage
- Invitation Page
- Group Details
- Error Page

• Register dynamically checks the validity of the username and the match between password and confirm password.

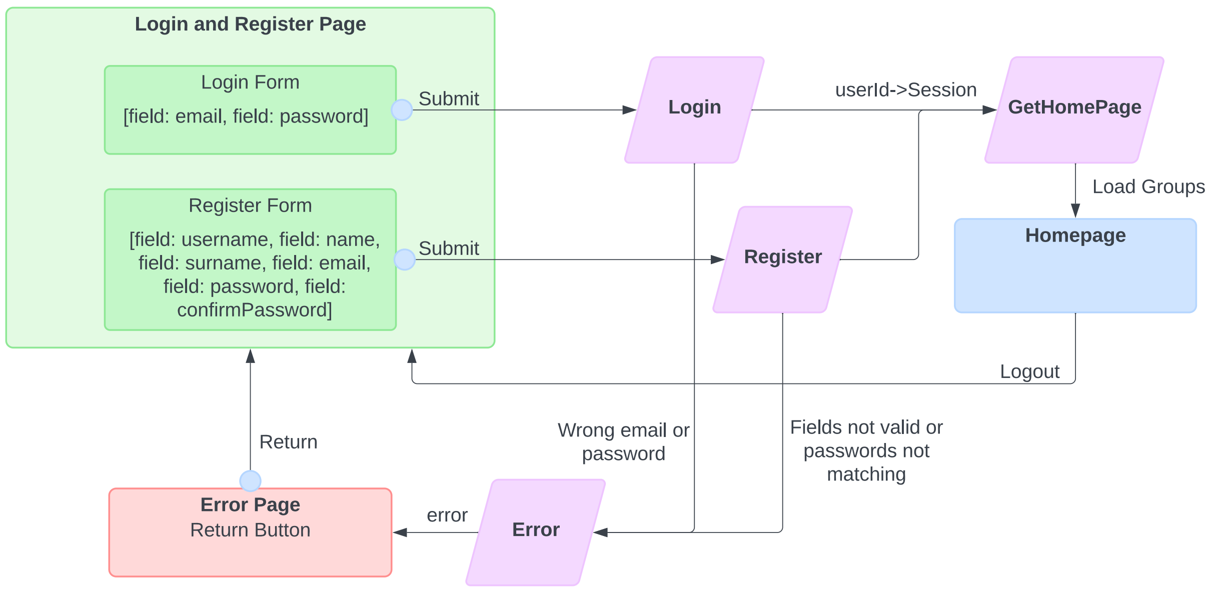
• The entire web application is developed as a single-page application, with all requests handled asynchronously, updating only the affected content.

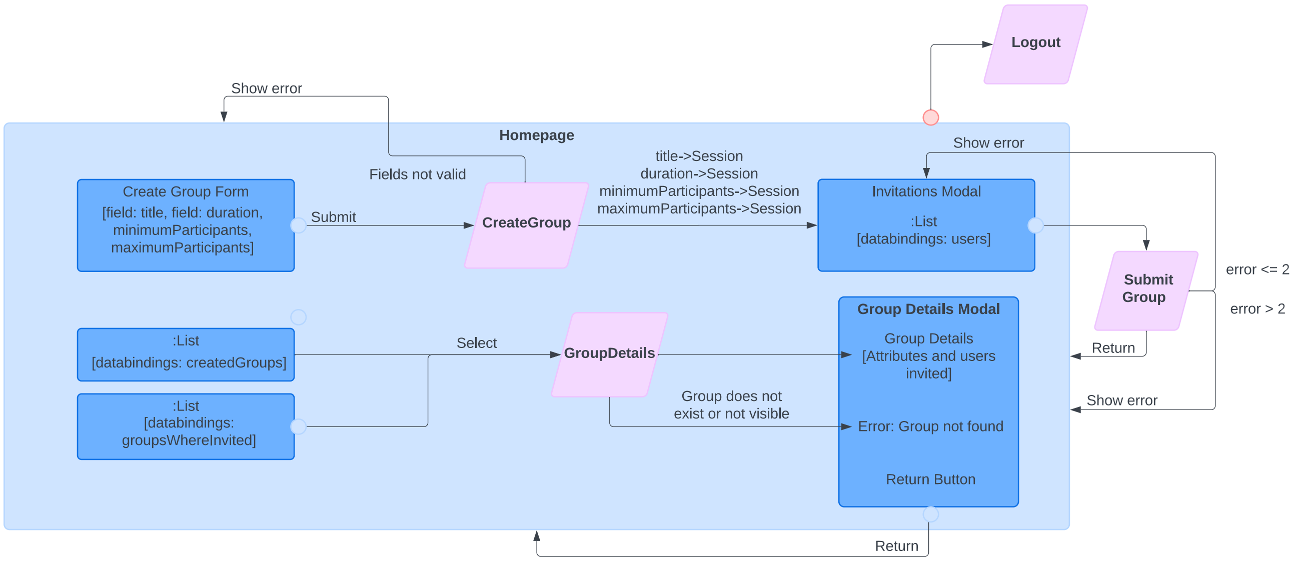
• The Invitation Page is built as a modal page; checks for invited numbers and errors are performed client-side.

• The Group Details Page displays a trash icon if the logged-in user is the group creator. Dragging invited members' names to the trash icon removes them from the group.

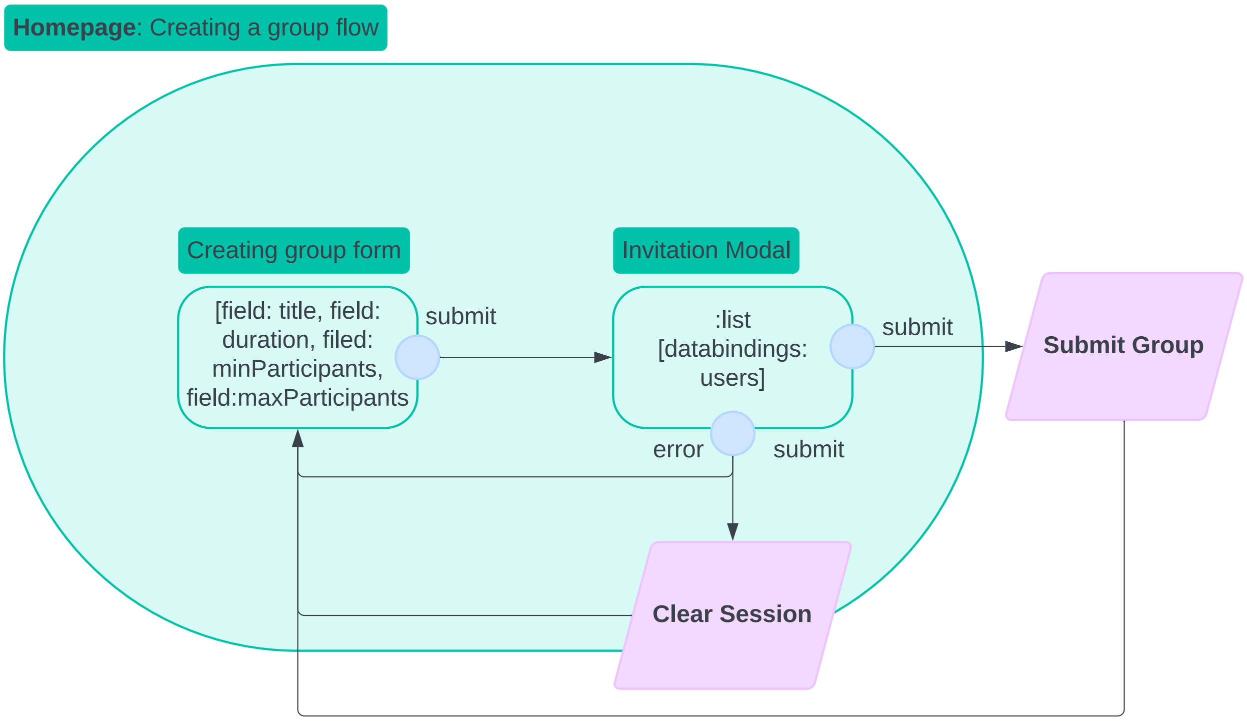
• All specifications from the Pure HTML version apply.

Application Design





Orchestrator for group creation


Homepage: Creating a group flow
Creating group form
[field: title, field: duration, filed: minParticipants, field:maxParticipants
submit
Invitation Modal
:list [databindings: users]
submit
Submit Group
error
submit
Clear Session

Events and actions

Client Side		Server Side	
Event	Action	Event	Action
loginForm→submit	Fields syntax control	POST email password	Fields syntax control and check credentials
Homepage→load	-	GET	Return homepage data
Homepage→createGroupForm→submit	Fields syntax control, open invitation modal	POST title, duration, minParticipants, maxParticipants	Fields syntax control and insert all data into session
Homepage→invitationForm→submit	Check fields validity, load homepage with new data	POST usersSelected[]	Check userId validity
Homepage→groupList→selectGroup	Open group details modal	GET groupId	Check if user can see the group, send back data
Logout	-	GET	Logout the client

Server 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)
    - removeFromGroup(userId, groupId)

Controllers:

- Login
- Register
- Logout
- GetHomePage
- CreateGroup
- SubmitGroup
- GroupDetails
- Error
- CheckUsername
- CleanSession
- GroupListUpdate
- RemoveMember

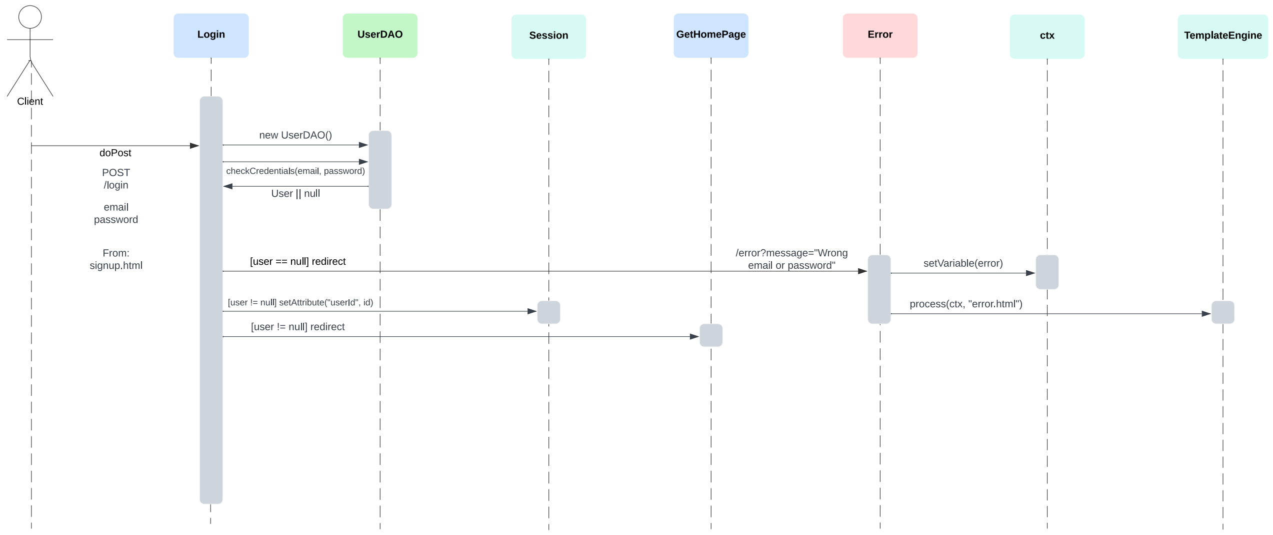
Client Components

View and view components:

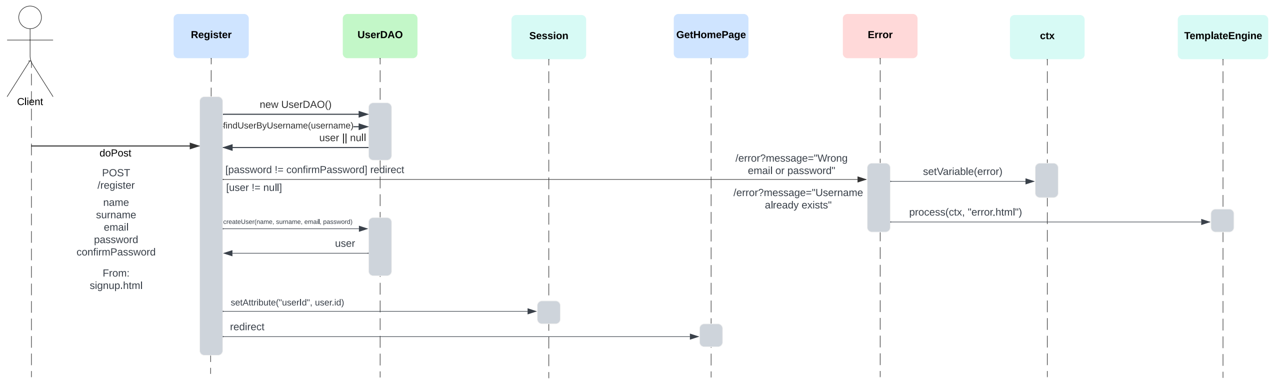
- Signin
  - Login Form
  - Register Form

- Homepage
  - Create group form
  - Invitation form
    - Show()
  - Created groups list
    - attachGroupLinkHandlers()
  - Invited groups list
    - attachGroupLinkHandlers()
  - Group Details
    - updateGroupDetails()
    - initializeDragAndDrop()
  - Notification
    - showNotificationModal()

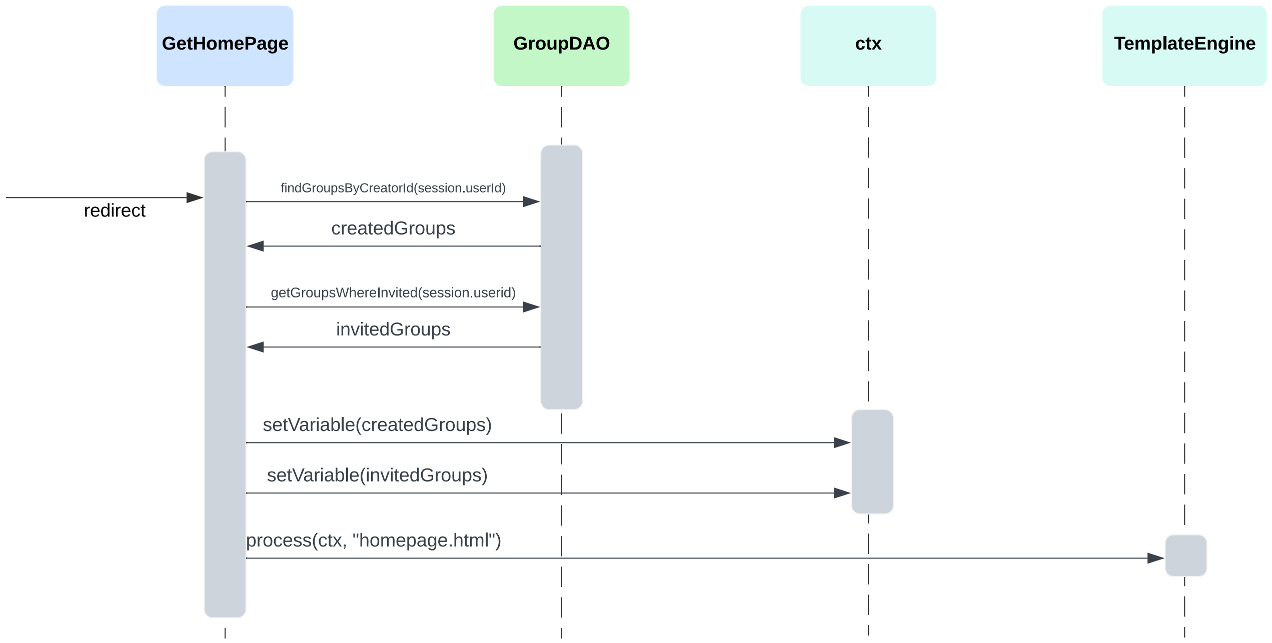
Event: Login



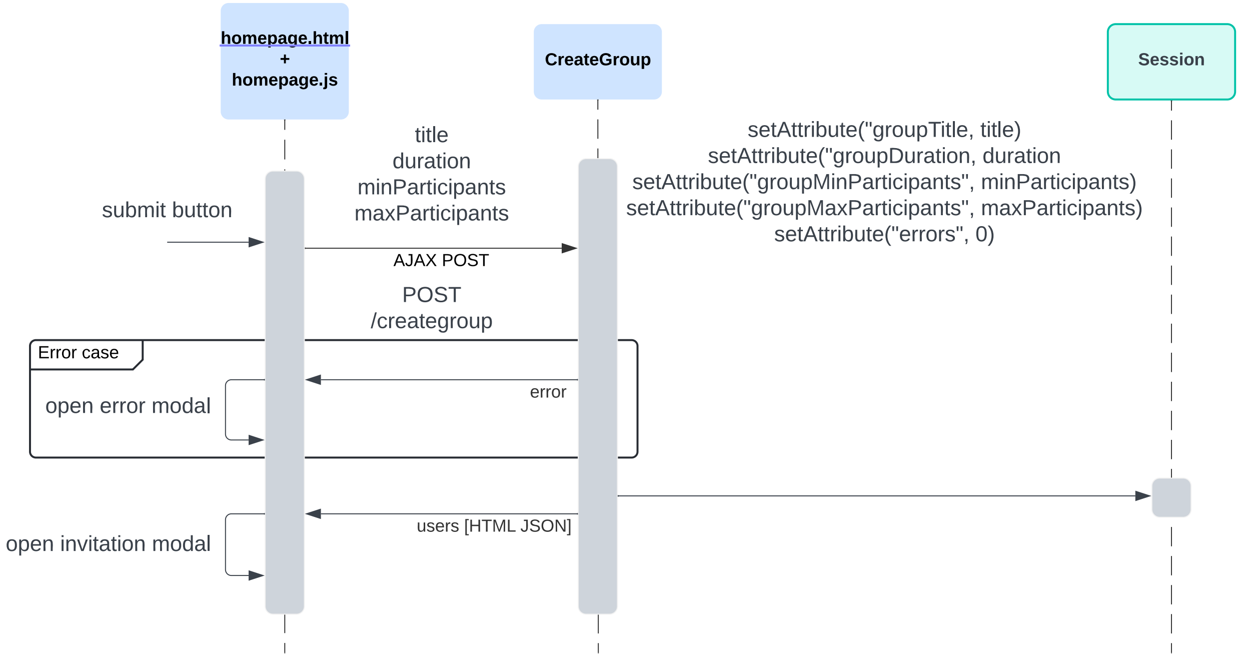
Event: Register



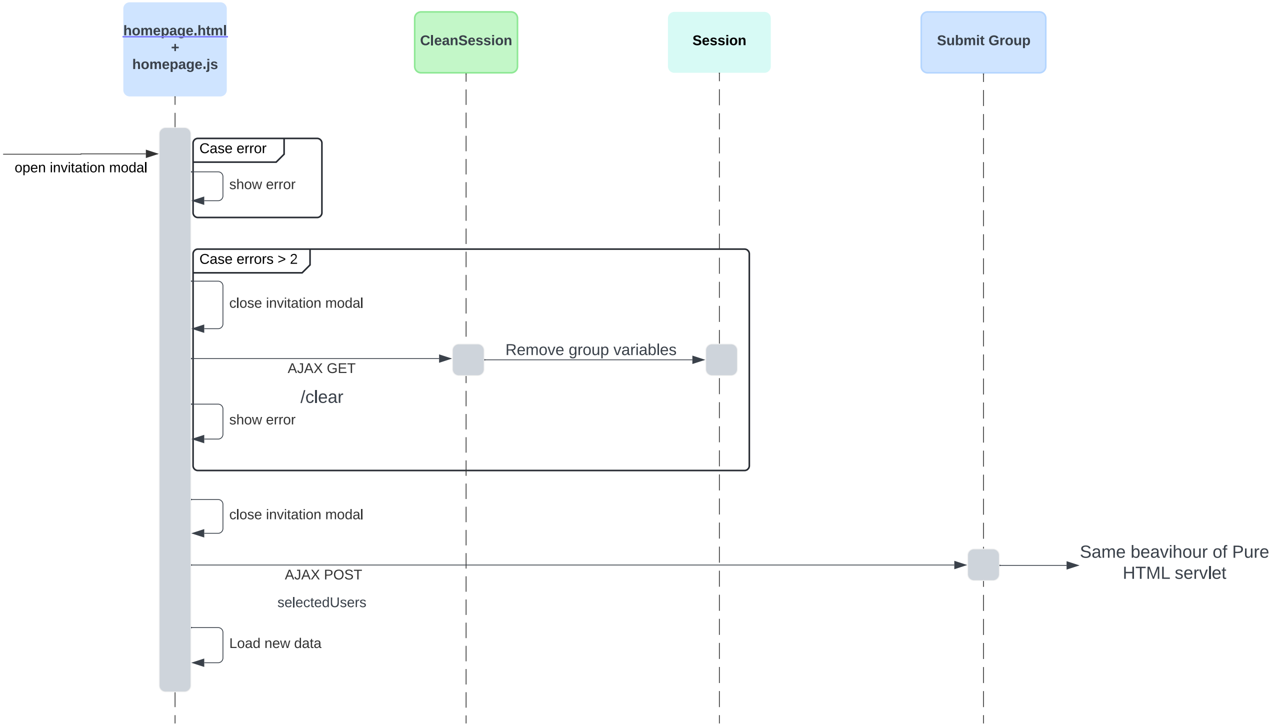
Event: Get Homepage


GetHomePage
GroupDAO
ctx
TemplateEngine
redirect
findGroupsByCreatorId(session.userId)
createdGroups
getGroupsWhereInvited(session.userid)
invitedGroups
setVariable(createdGroups)
setVariable(invitedGroups)
process(ctx, "homepage.html")

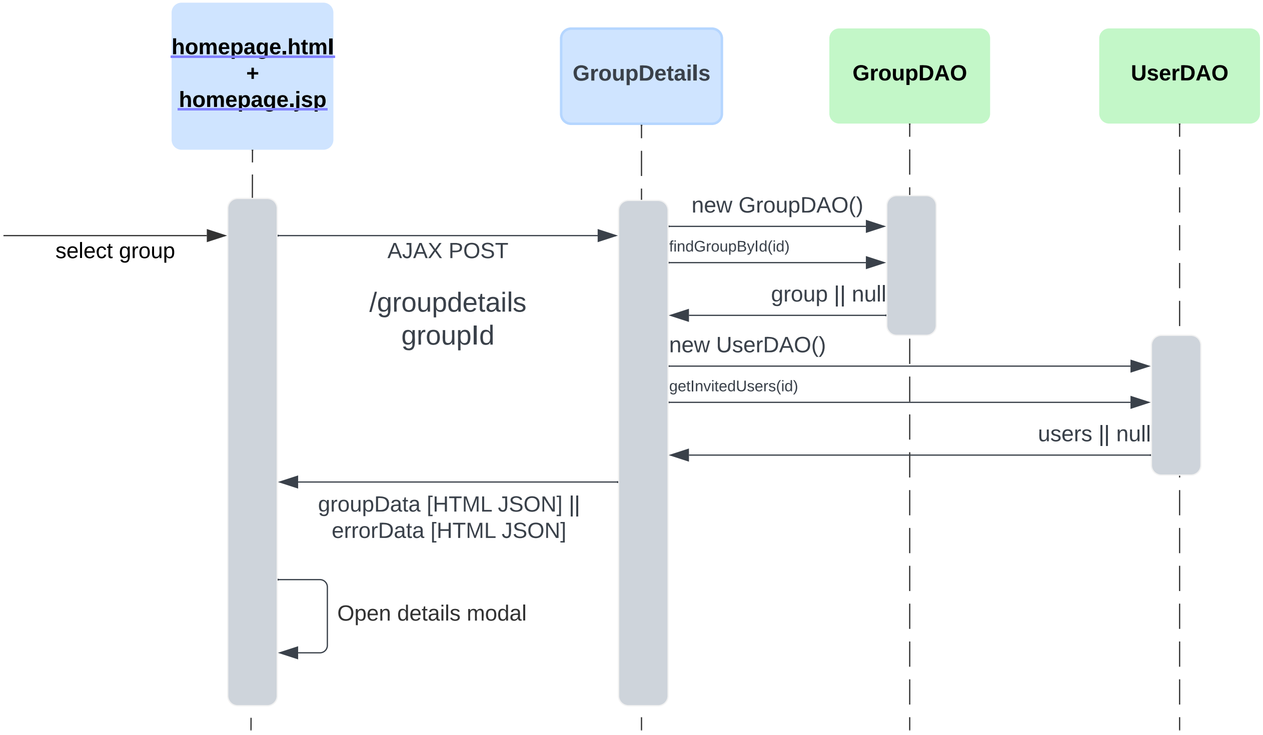
Event: Create Group


homepage.html
+
homepage.js
CreateGroup
Session
submit button
title
duration
minParticipants
maxParticipants
AJAX POST
POST
/creategroup
setAttribute("groupTitle, title)
setAttribute("groupDuration, duration
setAttribute("groupMinParticipants", minParticipants)
setAttribute("groupMaxParticipants", maxParticipants)
setAttribute("errors", 0)
Error case
open error modal
error
users [HTML JSON]
open invitation modal

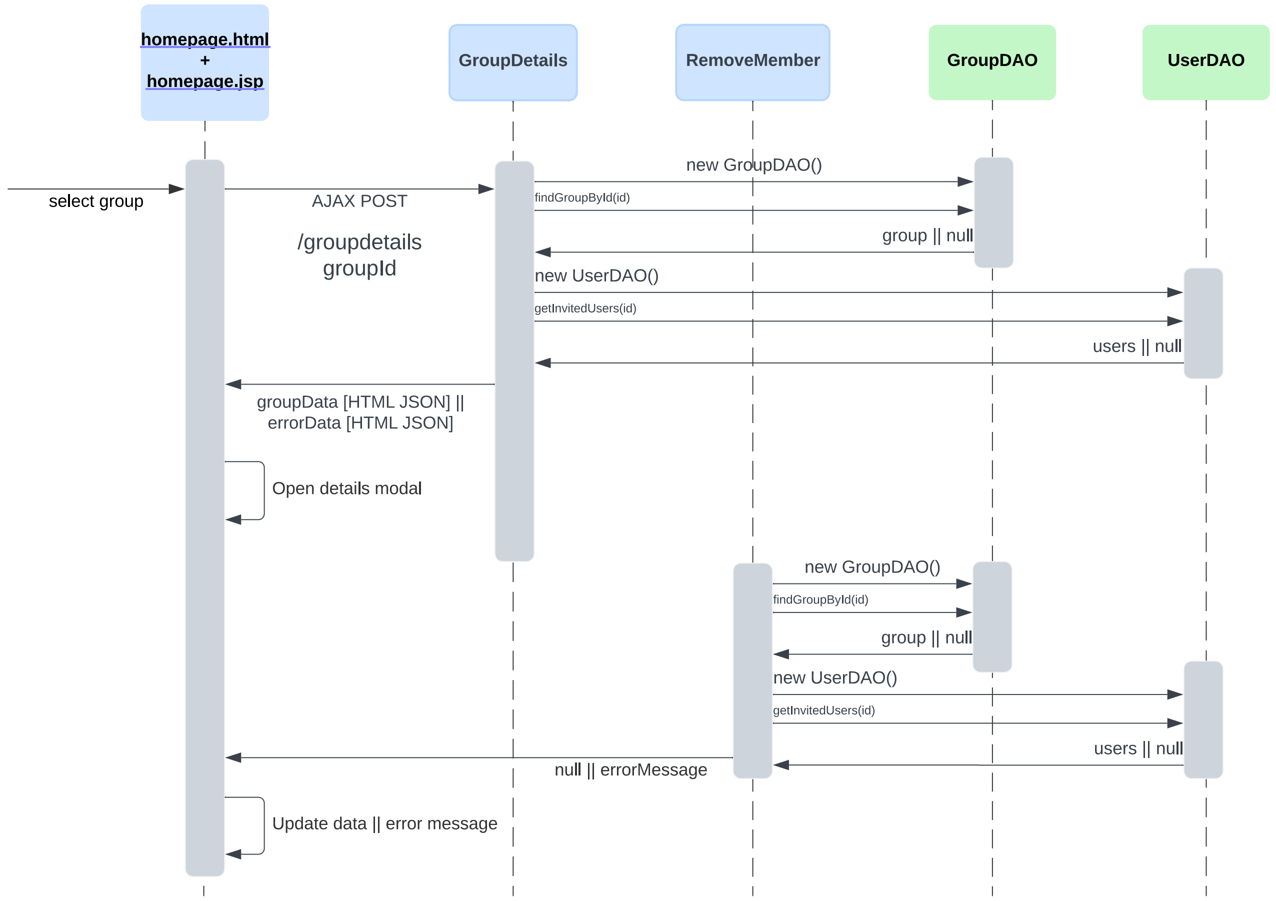
Event: Invite Users and Group Submit



Event: Group Details



Event: Remove User


homepage.html
+
homepage.jsp
GroupDetails
RemoveMember
GroupDAO
UserDAO
select group
AJAX POST
/groupdetails
groupId
new GroupDAO()
findGroupById(id)
group || null
new UserDAO()
getInvitedUsers(id)
users || null
groupData [HTML JSON] ||
errorData [HTML JSON]
Open details modal
new GroupDAO()
findGroupById(id)
group || null
new UserDAO()
getInvitedUsers(id)
users || null
null || errorMessage
Update data || error message