

Progetto TIW

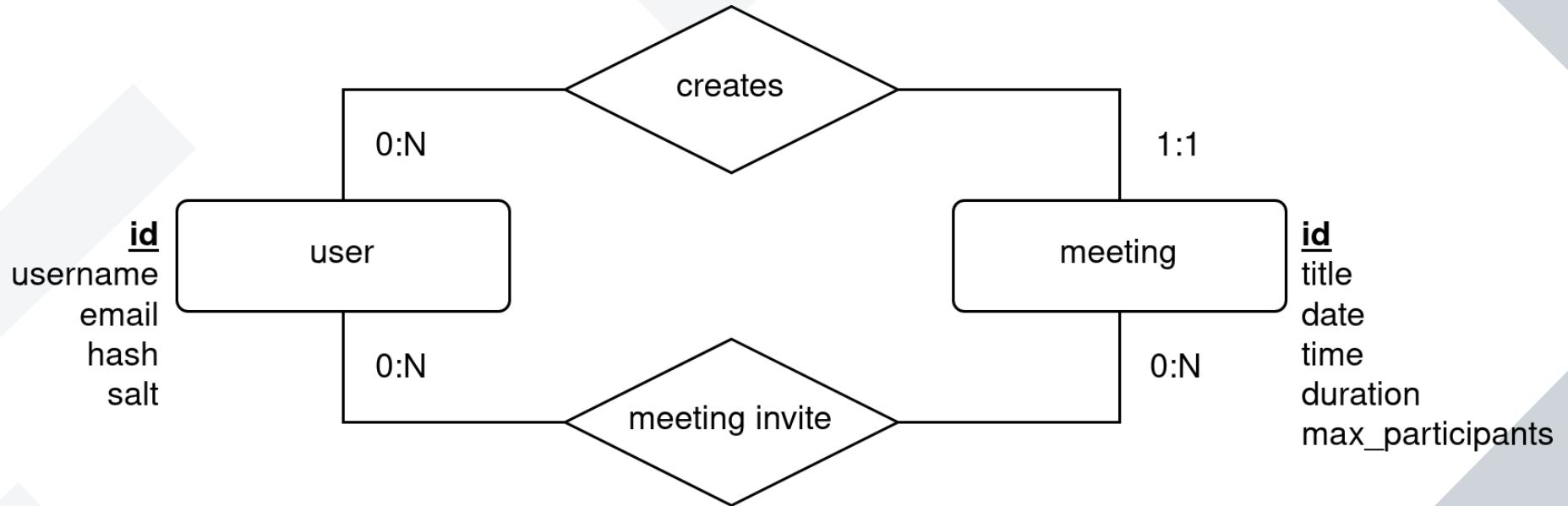
A.A. 2021/2022

Matteo Rampone
5 Settembre 2022

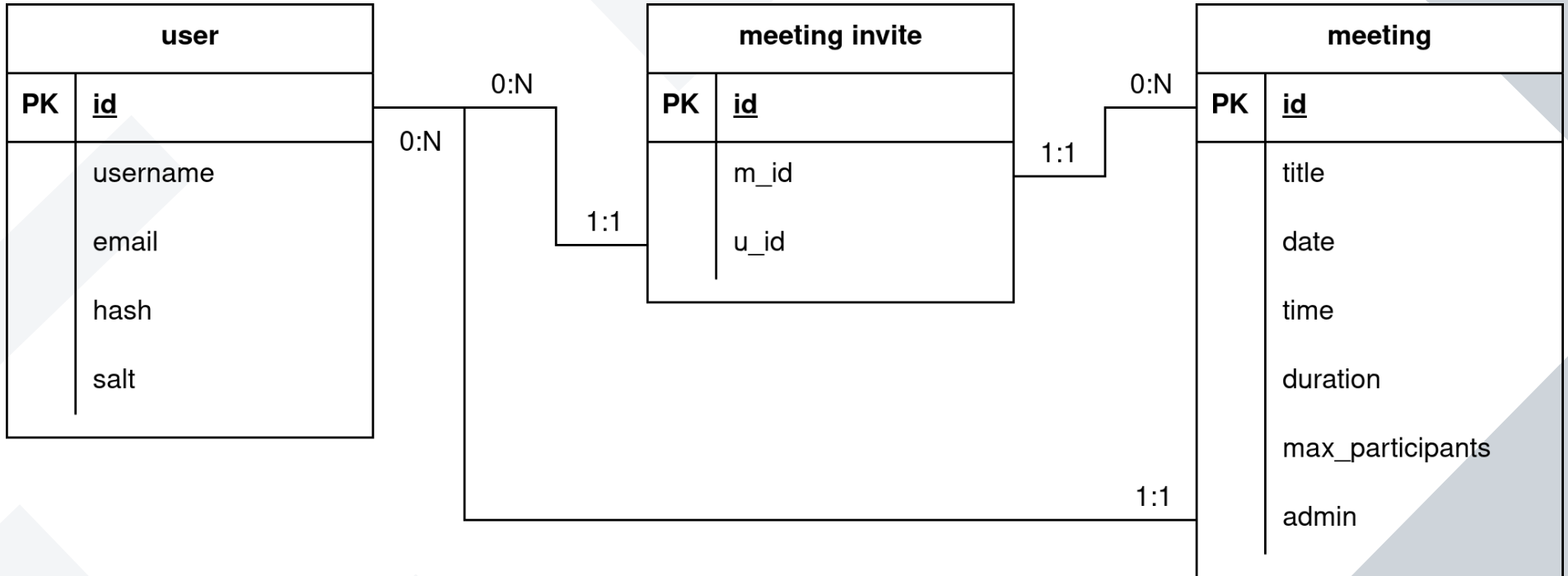
Data Specification analysis

Un'applicazione web consente la gestione di riunioni online. 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**. Una **riunione** ha un **titolo**, una **data**, un'**ora**, una **durata** e un **numero massimo di partecipanti**. L'**utente** fa il login e, se autenticato, accede all'HOME page che mostra l'elenco delle riunioni indette da lui e non ancora scadute, l'elenco delle riunioni cui è stato invitato e non ancora scadute, e una form per creare una nuova riunione. Quando l'utente inoltra la form con il bottone INVIA, appare una pagina ANAGRAFICA con l'elenco degli utenti registrati. L'utente può scegliere uno o più partecipanti dall'elenco e premere il bottone INVITA per **invitarli alla riunione**. 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". 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 è inferiore al massimo ammissibile, la riunione è memorizzata nella base di dati e associata agli utenti invitati e l'utente è rimandato alla HOME PAGE. Al terzo tentativo scorretto di assegnare troppi invitati a una riunione appare una pagina CANCELLAZIONE con un messaggio "Tre tentativi di definire una riunione con troppi partecipanti, la riunione non sarà creata" e un link per tornare all'HOME PAGE. In questo caso la riunione NON è memorizzata nella base di dati. L'applicazione non deve registrare nella base di dati riunioni con numero eccessivo di partecipanti. L'applicazione consente il logout dell'utente.

Database Design



Database Design



Local database schema

```
create table if not exists meeting_invite (  
    id      serial primary key,  
    m_id bigint unsigned not null references meeting (id),  
    u_id bigint unsigned not null references user (id),  
    constraint no_duplicate_invite unique(m_id, u_id)  
);
```

```
create table if not exists user (  
    id      serial primary key,  
    username varchar(20) not null unique,  
    email varchar(255) not null unique,  
    hash  binary(64)  not null,  
    salt  binary(16)  not null  
);
```

```
create table if not exists meeting (  
    id      serial primary key,  
    title   varchar(50)    not null,  
    date    date           not null,  
    time    time           not null,  
    duration int unsigned  not null,  
    max_participants int unsigned  not null,  
    admin   bigint unsigned not null references user (id)  
);
```

Note: procedures and triggers have been created to further secure the database.

Application requirements analysis

Un'applicazione web consente la gestione di riunioni online. 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. Una riunione ha un titolo, una data, un'ora, una durata e un numero massimo di partecipanti. L'utente fa il login e, se autenticato, accede all'**HOME page** che mostra l'elenco delle riunioni indette da lui e non ancora scadute, l'elenco delle riunioni cui è stato invitato e non ancora scadute, e una form per creare una nuova riunione. Quando l'utente inoltra la form con il bottone INVIA, appare una **pagina ANAGRAFICA** con l'elenco degli utenti registrati. L'utente può scegliere uno o più partecipanti dall'elenco e premere il bottone INVITA per invitarli alla riunione. 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". 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 è inferiore al massimo ammissibile, la riunione è memorizzata nella base di dati e associata agli utenti invitati e l'utente è rimandato alla **HOME PAGE**. Al terzo tentativo scorretto di assegnare troppi invitati a una riunione appare una **pagina CANCELLAZIONE** con un messaggio "Tre tentativi di definire una riunione con troppi partecipanti, la riunione non sarà creata" e un link per tornare all'**HOME PAGE**. In questo caso la riunione NON è memorizzata nella base di dati. L'applicazione non deve registrare nella base di dati riunioni con numero eccessivo di partecipanti. L'applicazione consente il logout dell'utente.

Si realizzi un'applicazione client server web che modifica le specifiche precedenti 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.
- La scelta dall'anagrafica deve essere realizzata con una pagina modale con i bottoni invia e cancella. NB: è una finestra che si apre per dare una qualche scelta o un qualche messaggio all'utente: 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.

Application requirements analysis

Un'applicazione web consente la gestione di riunioni online. 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. Una riunione ha un titolo, una data, un'ora, una durata e un numero massimo di partecipanti. L'utente fa il login e, se autenticato, accede all'HOME page che mostra l'**elenco delle riunioni indette da lui** e non ancora scadute, l'**elenco delle riunioni cui è stato invitato** e non ancora scadute, e una form per creare una nuova riunione. Quando l'utente inoltra la form con il **bottone INVIA**, appare una pagina ANAGRAFICA con l'**elenco degli utenti registrati**. L'utente può scegliere uno o più partecipanti dall'elenco e premere il **bottone INVITA** per invitarli alla riunione. 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"**. 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 è inferiore al massimo ammissibile, la riunione è memorizzata nella base di dati e associata agli utenti invitati e l'utente è rimandato alla HOME PAGE. Al terzo tentativo scorretto di assegnare troppi invitati a una riunione appare una pagina CANCELLAZIONE con un **messaggio "Tre tentativi di definire una riunione con troppi partecipanti, la riunione non sarà creata"** e un **link per tornare all'HOME PAGE**. In questo caso la riunione NON è memorizzata nella base di dati. L'applicazione non deve registrare nella base di dati riunioni con numero eccessivo di partecipanti. L'applicazione consente il logout dell'utente.

Si realizzi un'applicazione client server web che modifica le specifiche precedenti 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.
- La scelta dall'anagrafica deve essere realizzata con una **pagina modale** con i **bottoni invia** e **cancella**. NB: è una finestrella che si apre per dare una qualche scelta o un qualche messaggio all'utente: 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.

Application requirements analysis

Un'applicazione web consente la gestione di riunioni online. 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. Una riunione ha un titolo, una data, un'ora, una durata e un numero massimo di partecipanti. L'utente fa il login e, se autenticato, accede all'HOME page che mostra l'elenco delle riunioni indette da lui e non ancora scadute, l'elenco delle riunioni cui è stato invitato e non ancora scadute, e una form per creare una nuova riunione. Quando l'utente **inoltra la form** con il bottone INVIA, appare una pagina ANAGRAFICA con l'elenco degli utenti registrati. L'utente può scegliere uno o più partecipanti dall'elenco e **premere il bottone INVITA** per invitarli alla riunione. 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". 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 è inferiore al massimo ammissibile, la riunione è memorizzata nella base di dati e associata agli utenti invitati e l'utente è rimandato alla HOME PAGE. Al terzo tentativo scorretto di assegnare troppi invitati a una riunione appare una pagina CANCELLAZIONE con un messaggio "Tre tentativi di definire una riunione con troppi partecipanti, la riunione non sarà creata" e un link per tornare all'HOME PAGE. In questo caso la riunione NON è memorizzata nella base di dati. L'applicazione non deve registrare nella base di dati riunioni con numero eccessivo di partecipanti. L'applicazione consente il logout dell'utente.

Si realizzi un'applicazione client server web che modifica le specifiche precedenti 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.
- La **scelta dall'anagrafica** deve essere realizzata con una pagina modale con i bottoni invia e cancella. NB: è una finestra che si apre per dare una qualche scelta o un qualche messaggio all'utente: 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.

Application requirements analysis

Un'applicazione web consente la gestione di riunioni online. 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. Una riunione ha un titolo, una data, un'ora, una durata e un numero massimo di partecipanti. L'utente fa il **login** e, se autenticato, accede all'HOME page che mostra l'elenco delle riunioni indette da lui e non ancora scadute, l'elenco delle riunioni cui è stato invitato e non ancora scadute, e una form per **creare una nuova riunione**. Quando l'utente inoltra la form con il bottone INVIA, appare una pagina ANAGRAFICA con l'elenco degli utenti registrati. L'utente può scegliere uno o più partecipanti dall'elenco e premere il bottone INVITA per **invitarli alla riunione**. 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". 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 è inferiore al massimo ammissibile, **la riunione è memorizzata nella base di dati e associata agli utenti invitati** e l'utente è rimandato alla HOME PAGE. Al terzo tentativo scorretto di assegnare troppi invitati a una riunione appare una pagina CANCELLAZIONE con un messaggio "Tre tentativi di definire una riunione con troppi partecipanti, la riunione non sarà creata" e un link per tornare all'HOME PAGE. In questo caso **la riunione NON è memorizzata nella base di dati**. L'applicazione non deve registrare nella base di dati riunioni con numero eccessivo di partecipanti. L'applicazione consente il **logout** dell'utente.

Si realizzi un'applicazione client server web che modifica le specifiche precedenti 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.
- La scelta dall'anagrafica deve essere realizzata con una pagina modale con i bottoni invia e cancella. NB: è una finestra che si apre per dare una qualche scelta o un qualche messaggio all'utente: 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.

Hypotheses, Changes and Additions to the original Specifications

Common between models

- Usernames are required to log-in and register, emails are not.
However: emails are checked for uniqueness during registration too. This is because emails are used to identify meeting participants.
- For the purposes of integrity to the database, all constraints to the internal representation of the data are handled both on a DAO's level and on the DataBase logic.
- A username is a string of alphanumeric characters optionally interleaved with underscores, with maximum length 20.
- An email is a string matching the RFC 5322 standard.
- A password must:
 - Be at least 8 characters long
 - Contain at least 1 special character
 - Contain at least 1 upper case english character
 - Contain at least 1 lower case english character
 - Contain at least 1 numeric character
- A meeting cannot be created in the past. So long as the meeting start time and date have already passed during the creation of the element, the server will deny the request.
- A meeting is considered expired once the sum of start time and duration is in the past compared to the current time on the server.
- A meeting title is at maximum 50 characters long.
- A meeting has an admin user, the admin cannot be invited to the meeting as it is already linked to it.
- The database stores information about dates and times using the UTC format.

Hypotheses, Changes and Additions to the original Specifications

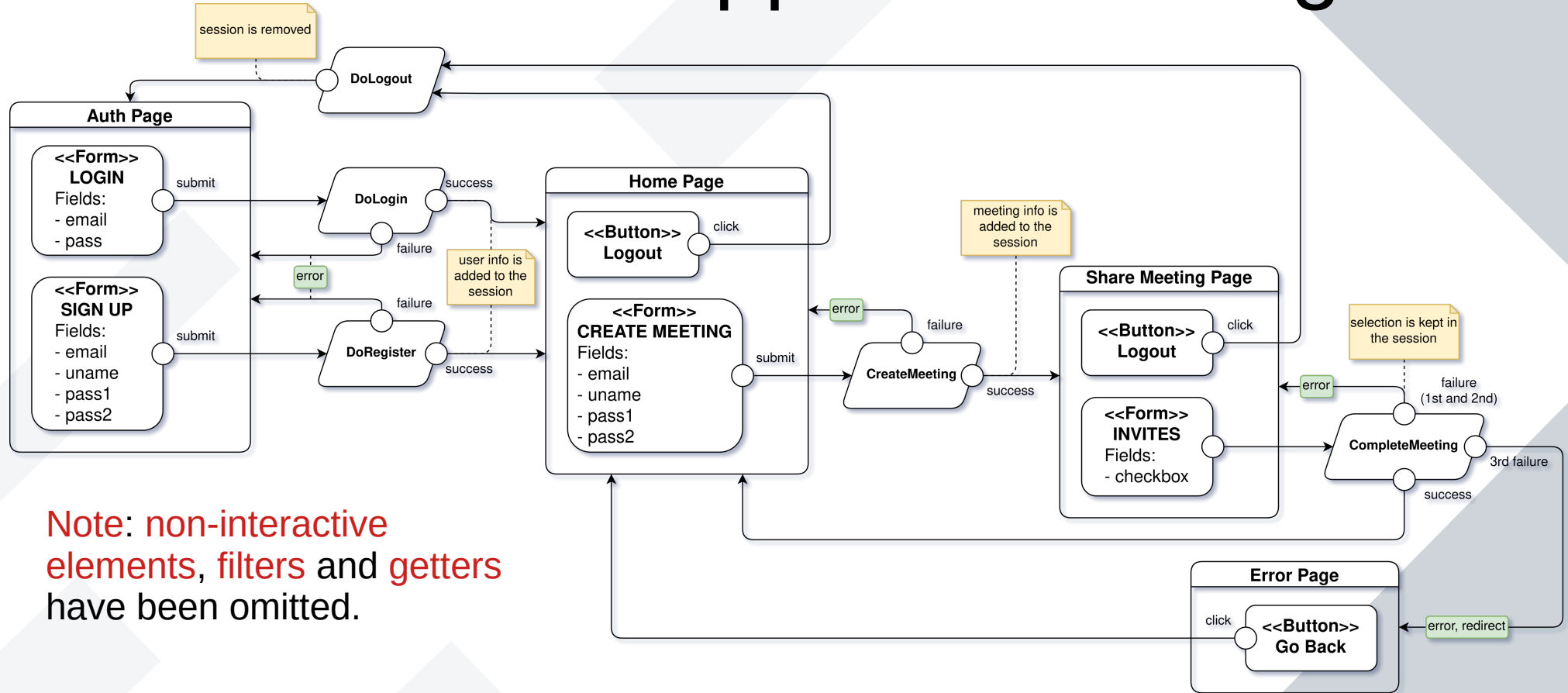
Pure HTML

- There is no timezone selection tool during the creation of the form, so users should be mindful of submitting only UTC aligned dates (following the GMT timezone).

HTML + JS model

- There is no server-side “failed meeting submission” counter since there is no need to store the meeting creation data in a session object.
- If the client’s form submission controls let a submission go through to the server, and that submission is subsequently denied, the client is to interpret any form-related failure as a third failed attempt on submission.
- Wherever possible, the server will streamline responses to the minimum. Most errors do not need a dedicated error message to be shipped along side the response, so most of the error reporting to a client’s JS modules is done through HTTP status codes.

Pure HTML Application design



Note: non-interactive elements, filters and getters have been omitted.

PureHTML Components

Model objects:

- MeetingBean
- MeetingCreationBean
- UserBean

Data Access Objects:

- MeetingDAO
 - getCreatedMeetings(admin)
 - getInvitedMeetings(user)
 - createMeeting(meetingInfo, invites)
- UserDao
 - getUserCount()
 - getOtherUsers(user)
 - checkUniqueEmail(email)
 - checkUniqueUname(uname)
 - authenticate(email, pass)
 - addUser(email, pass, uname)

Controllers:

- CompleteMeeting
- CreateMeeting
- DoLogin
- DoLogout
- DoRegister
- GetDefaultPage

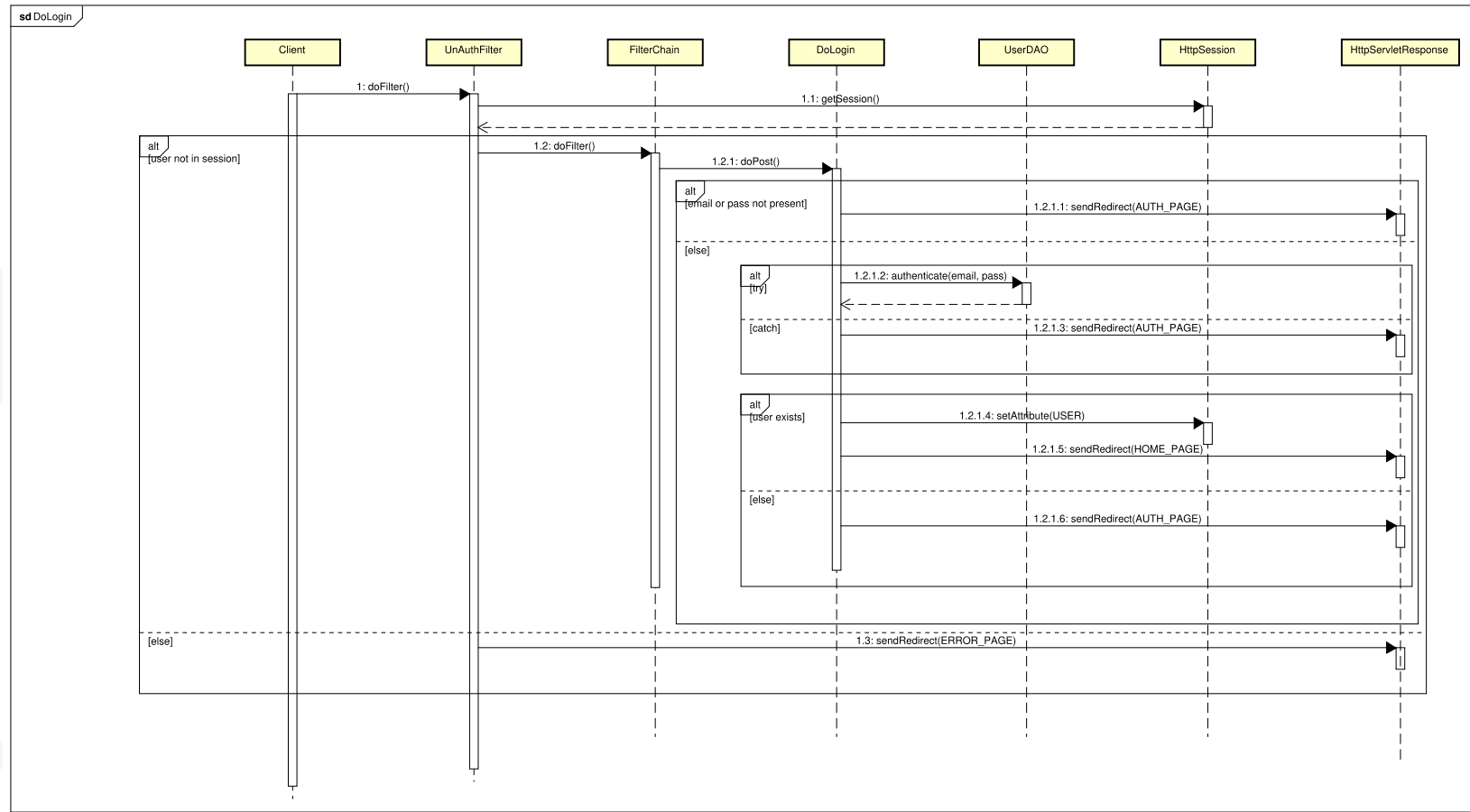
Filters:

- AuthFilter
- UnAuthFilter

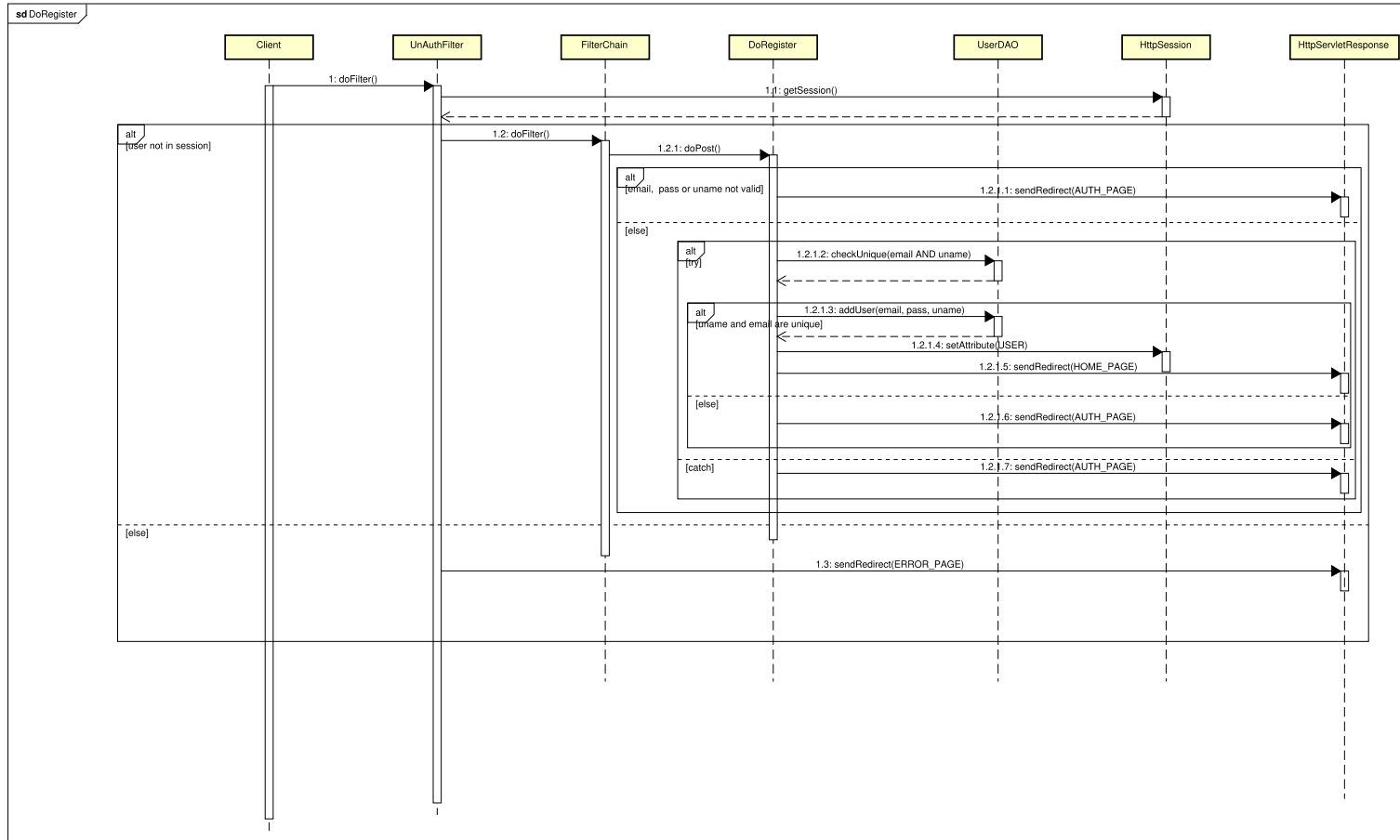
Views:

- GetAuthentication
- GetDefaultPage
- GetErrors
- GetHome
- GetInvites

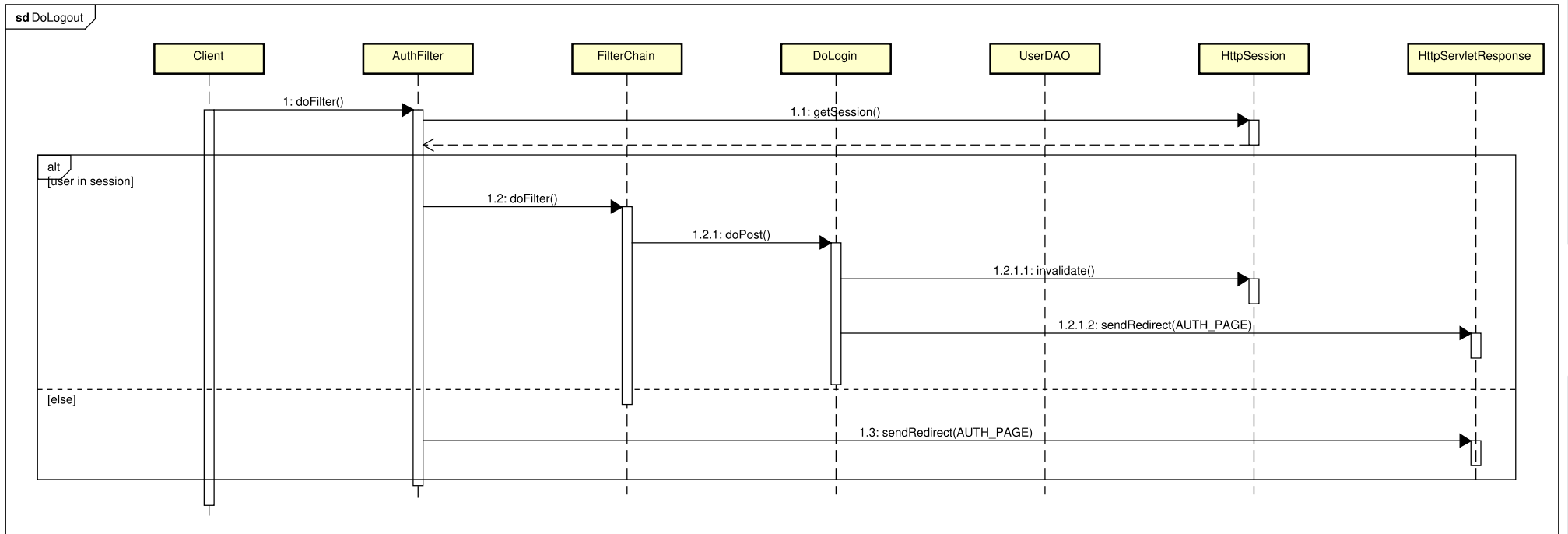
PureHTML Events



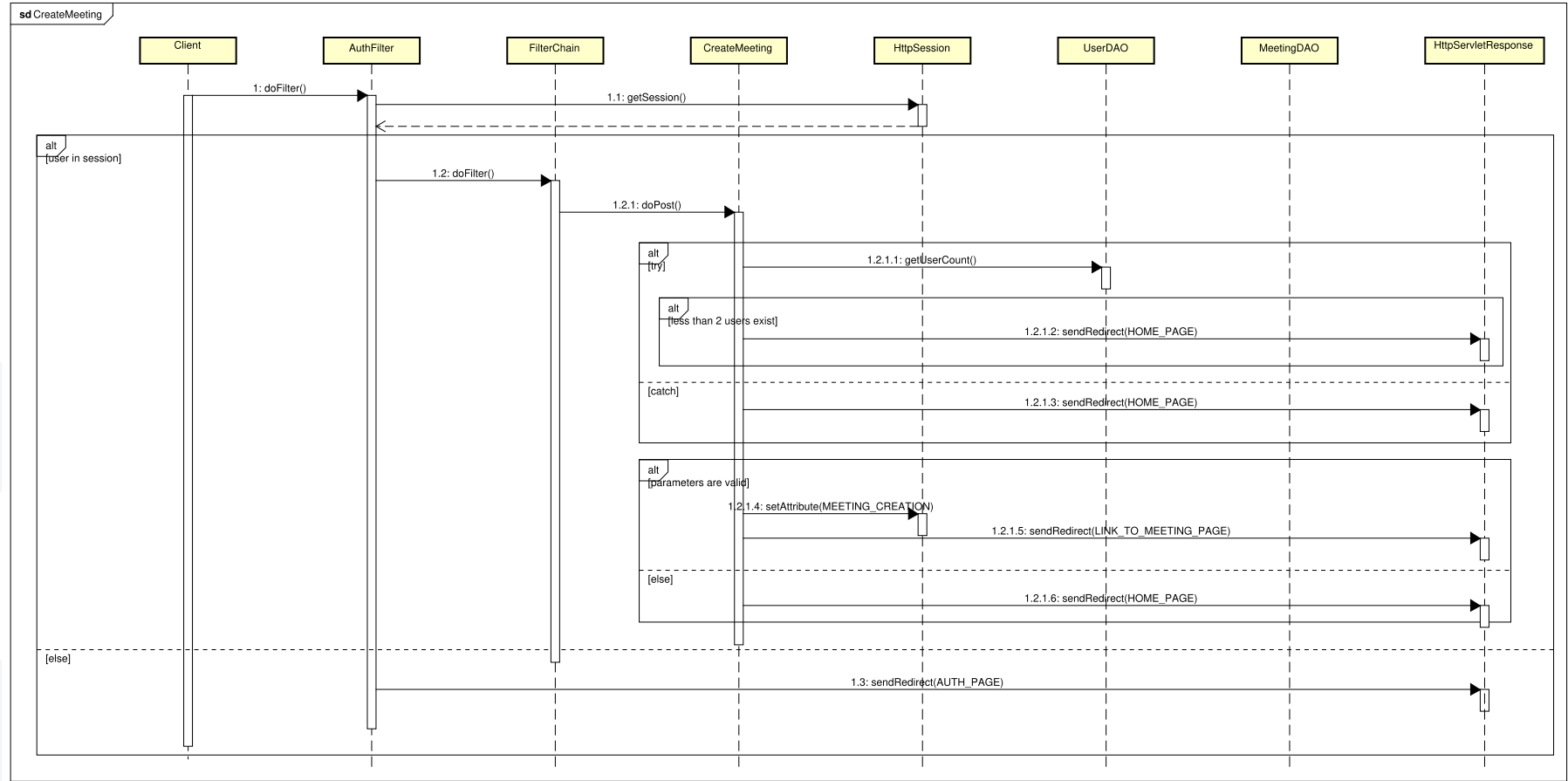
PureHTML Events



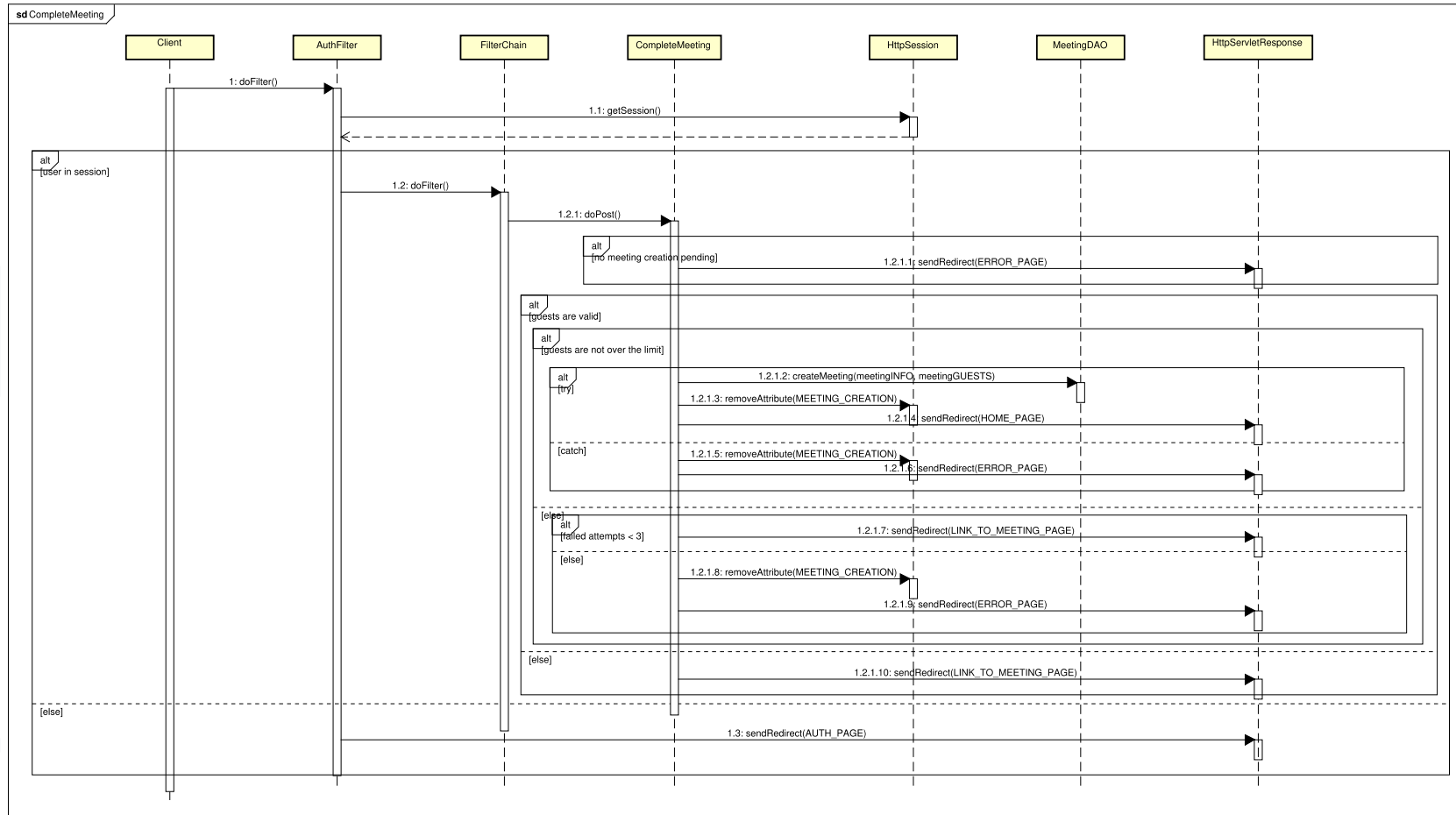
PureHTML Events



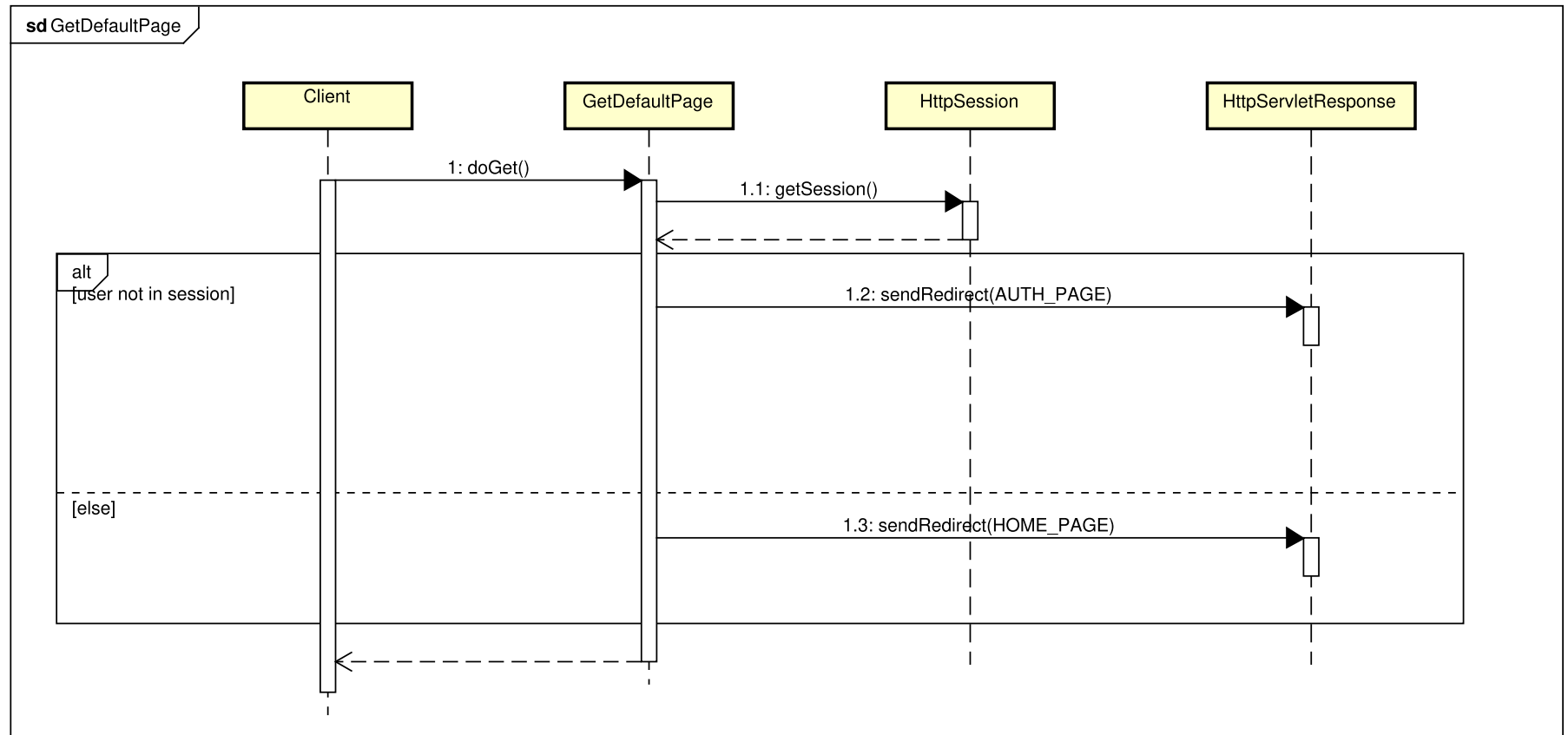
PureHTML Events



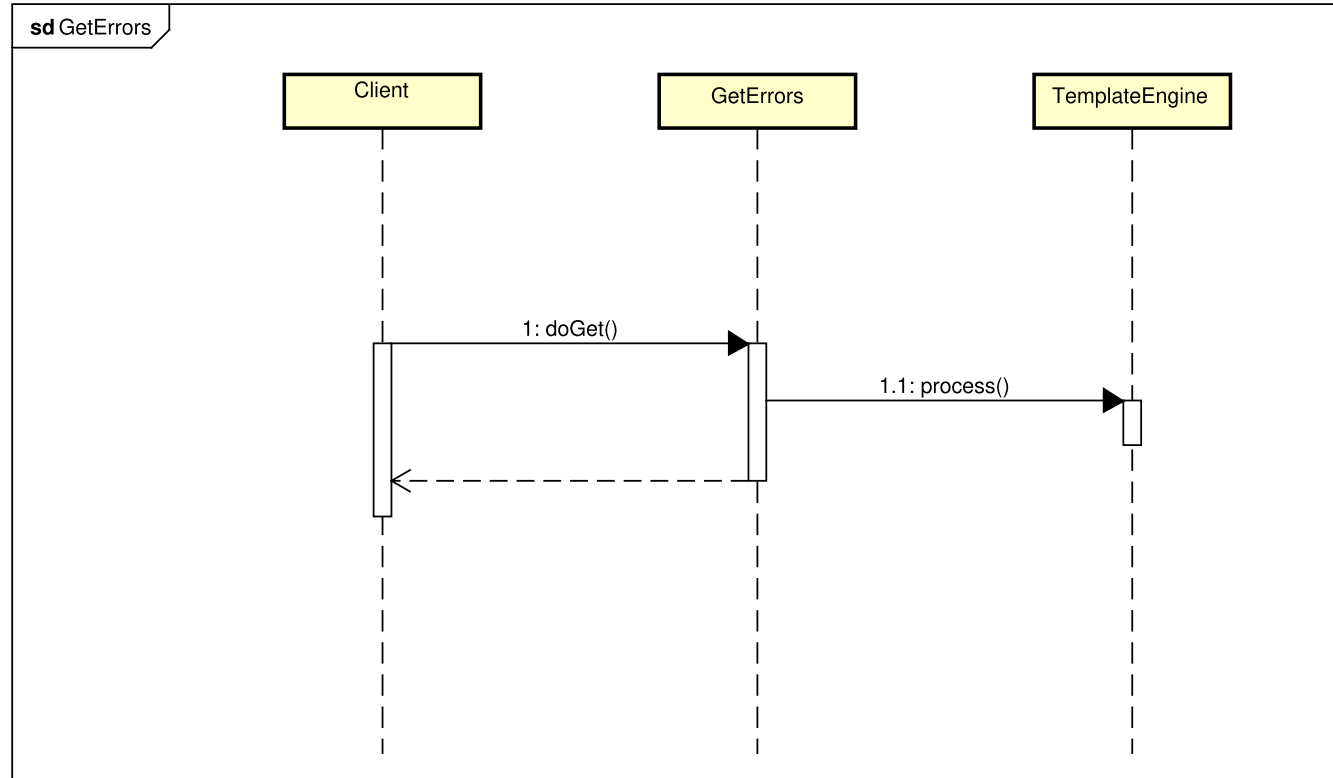
PureHTML Events



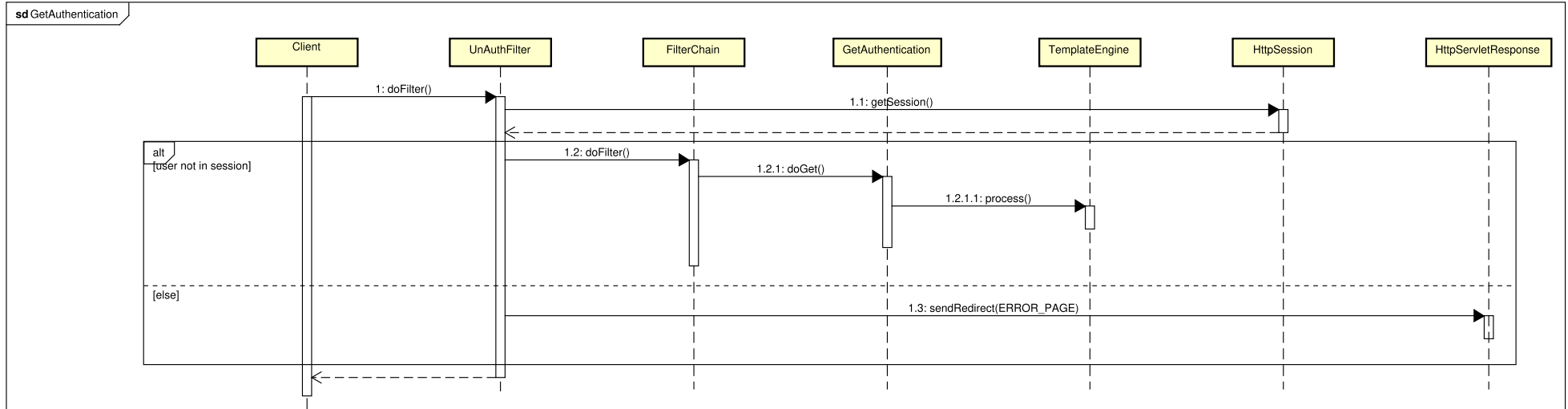
PureHTML Events



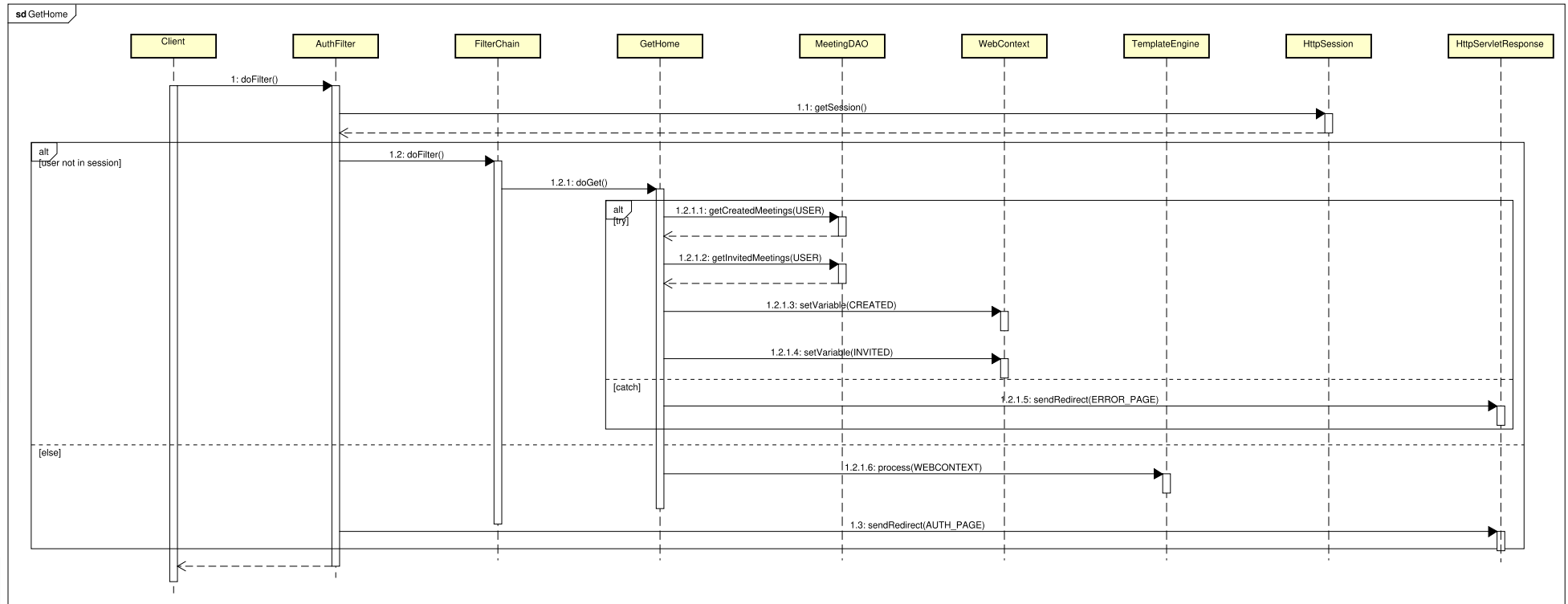
PureHTML Events



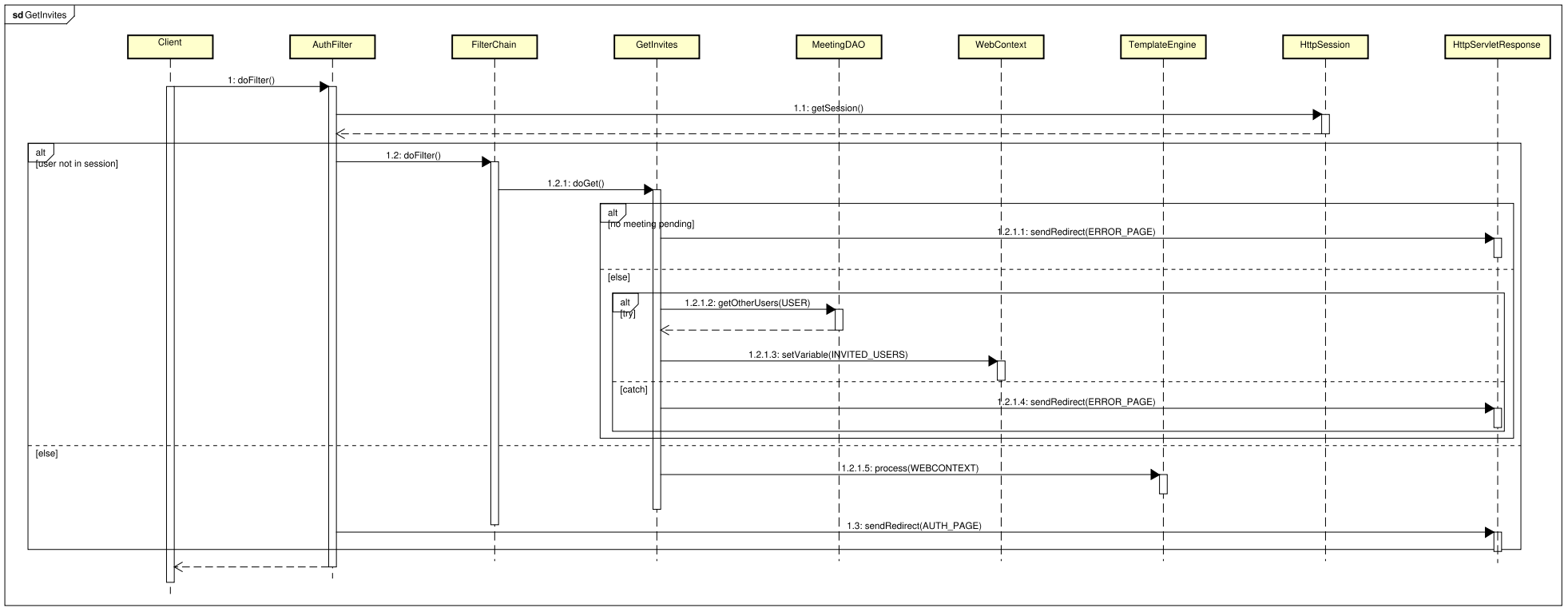
PureHTML Events



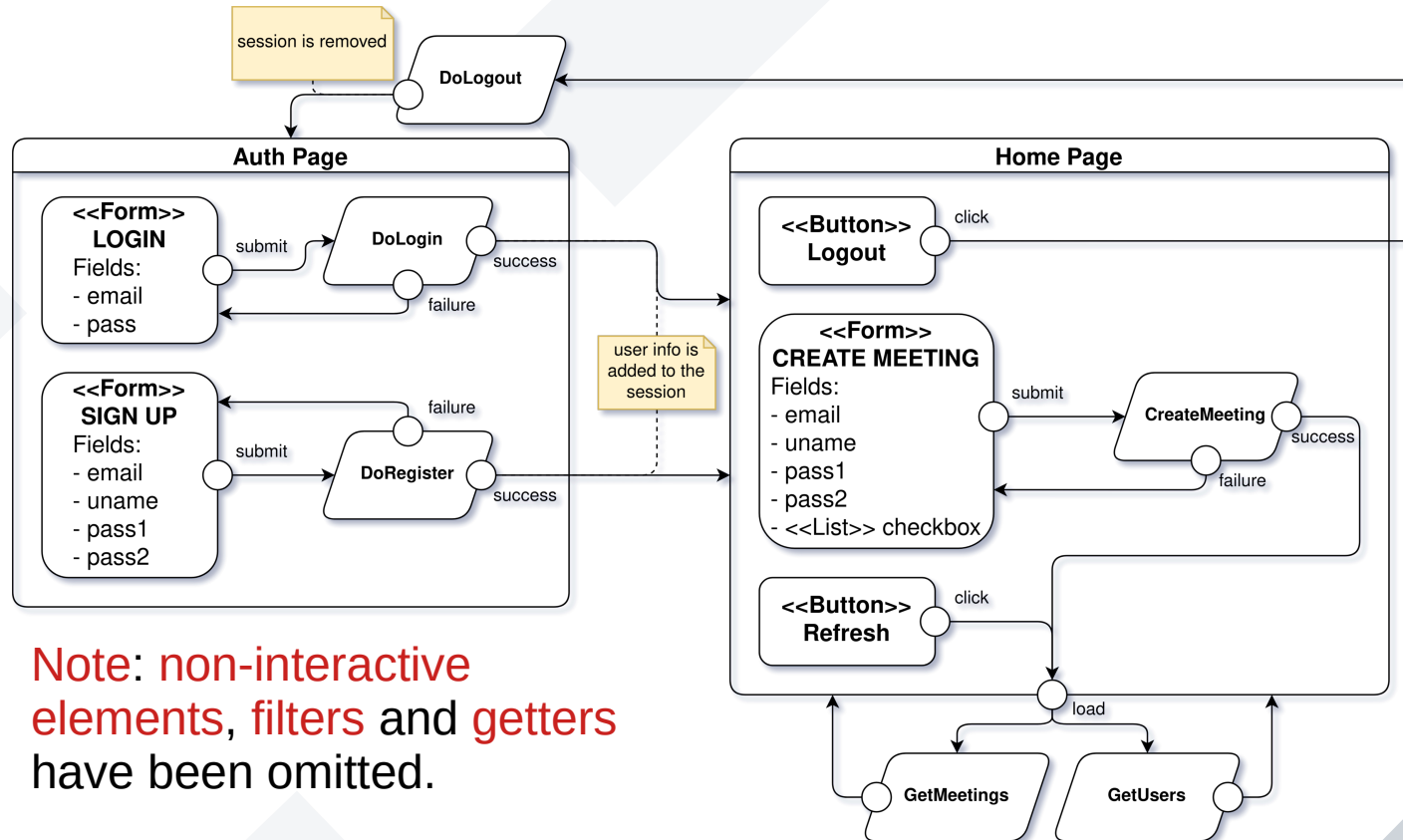
PureHTML Events



PureHTML Events



HTML + JS Application design



Note: non-interactive elements, filters and getters have been omitted.

HTML + JS Components

Model objects:

- MeetingBean
- UserBean

Data Access Objects:

- MeetingDAO
 - getCreatedMeetings(admin)
 - getInvitedMeetings(user)
 - createMeeting(meetingInfo, invites)
- UserDAO
 - getUserCount()
 - getOtherUsers(user)
 - checkUnique(email, uname)
 - authenticate(email, pass)
 - addCredentials(email, pass, uname)

Controllers:

- CreateMeeting
- DoLogin
- DoLogout
- DoRegister
- GetMeetings
- GetUsers
- LoadDefaultPage

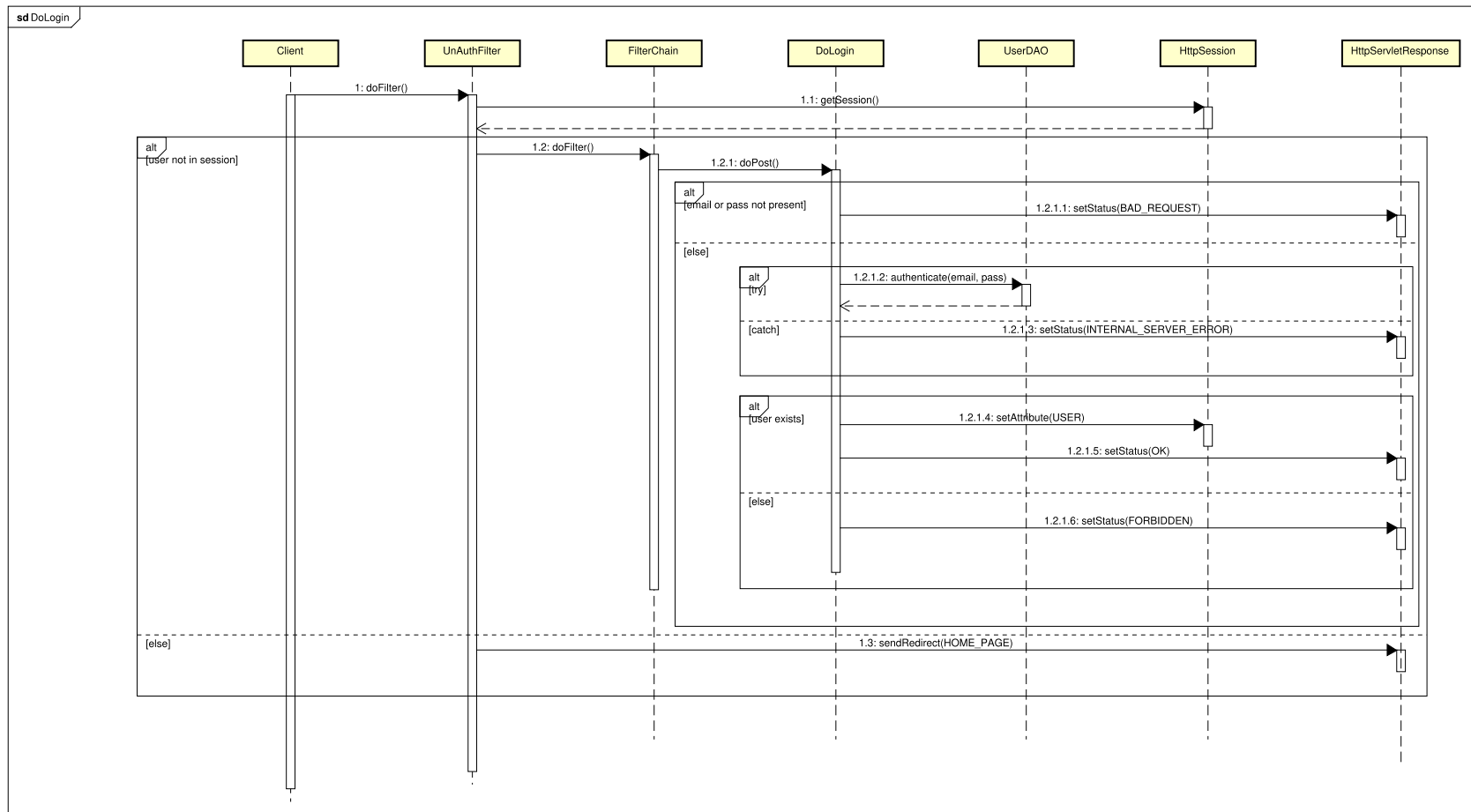
Filters:

- AuthFilter
- UnAuthFilter

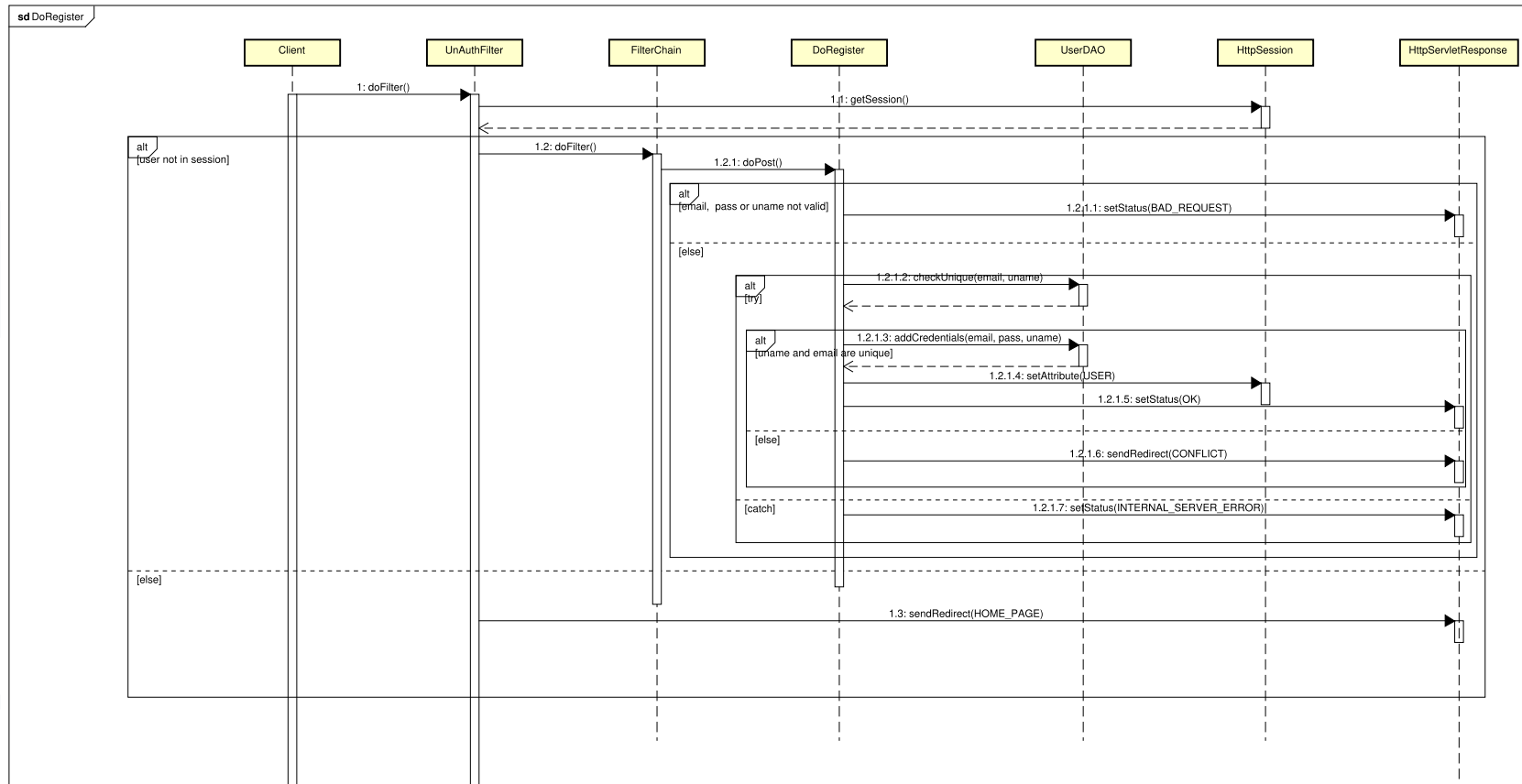
Views:

- auth.html
- home.html

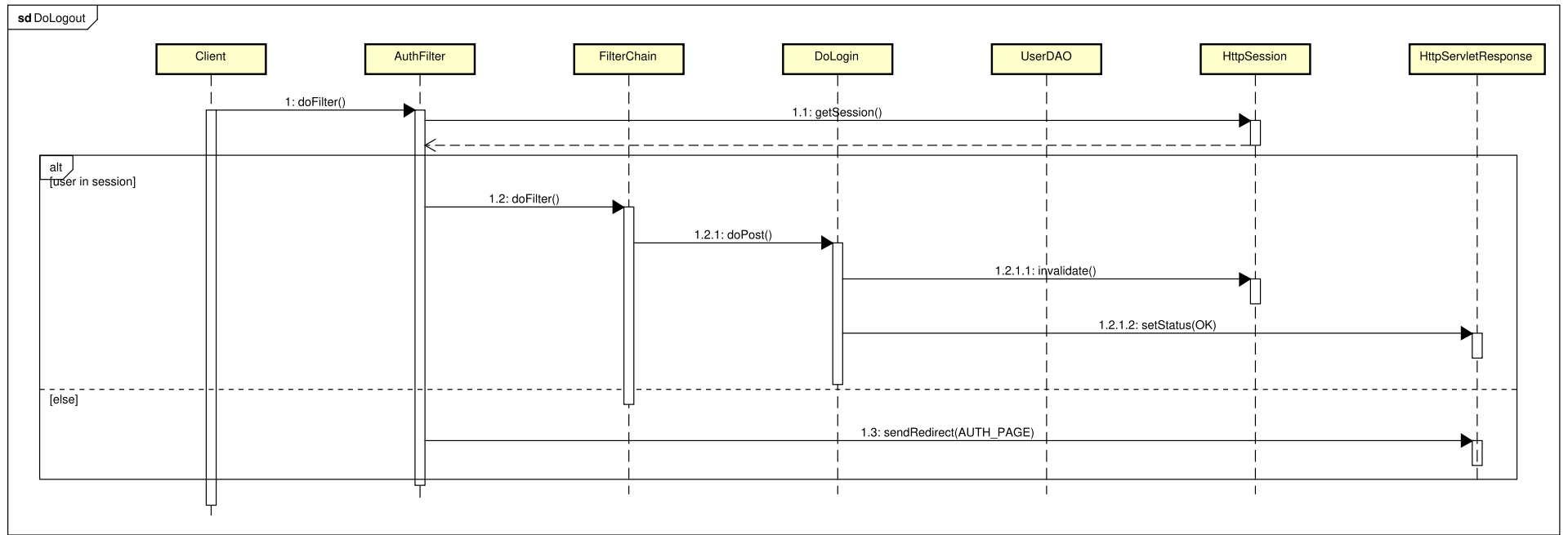
HTML + JS Events



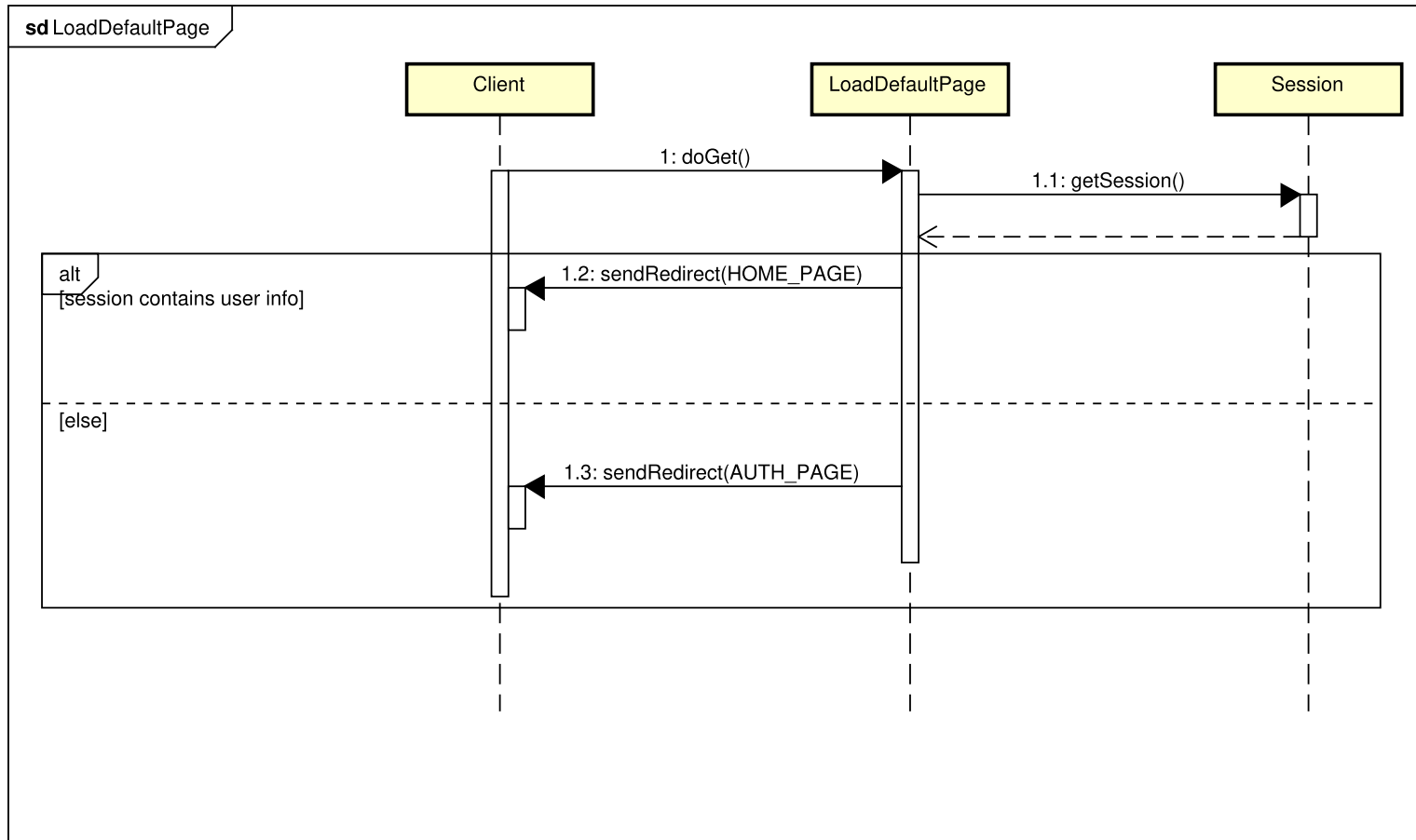
HTML + JS Events



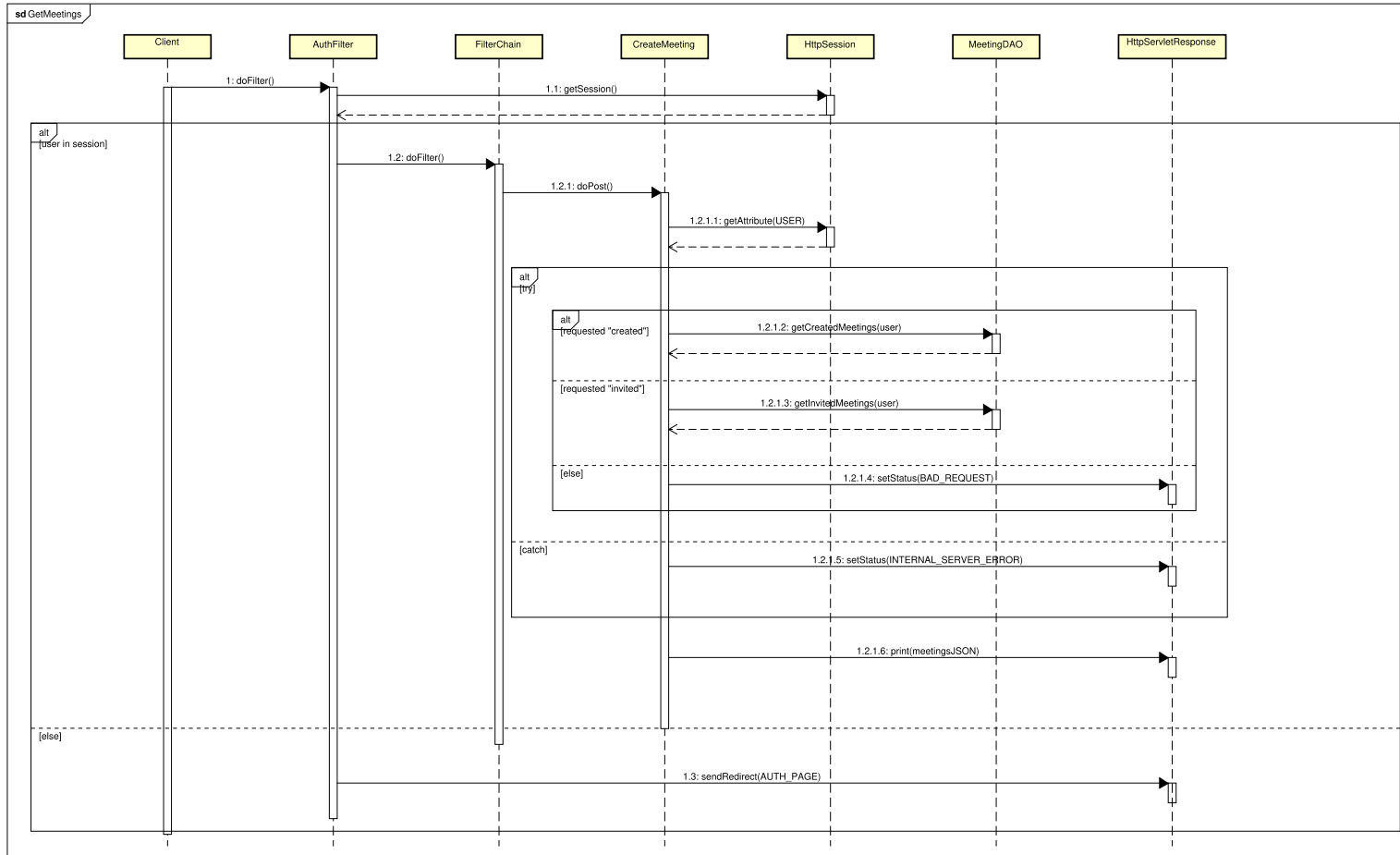
HTML + JS Events



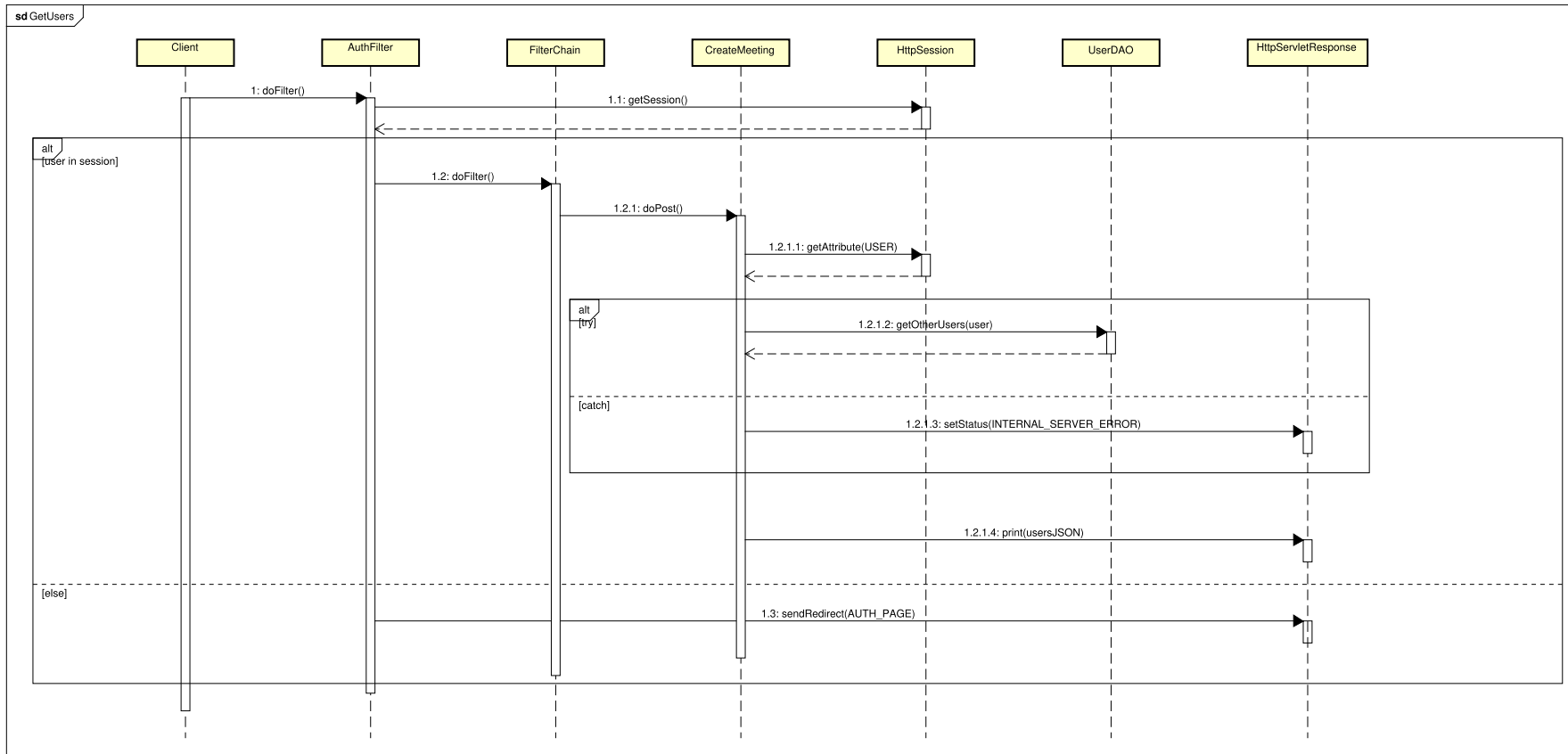
HTML + JS Events



HTML + JS Events



HTML + JS Events



HTML + JS Events

