PROGETTO NUMERO 5: riunioni online

Mosconi Alessandro

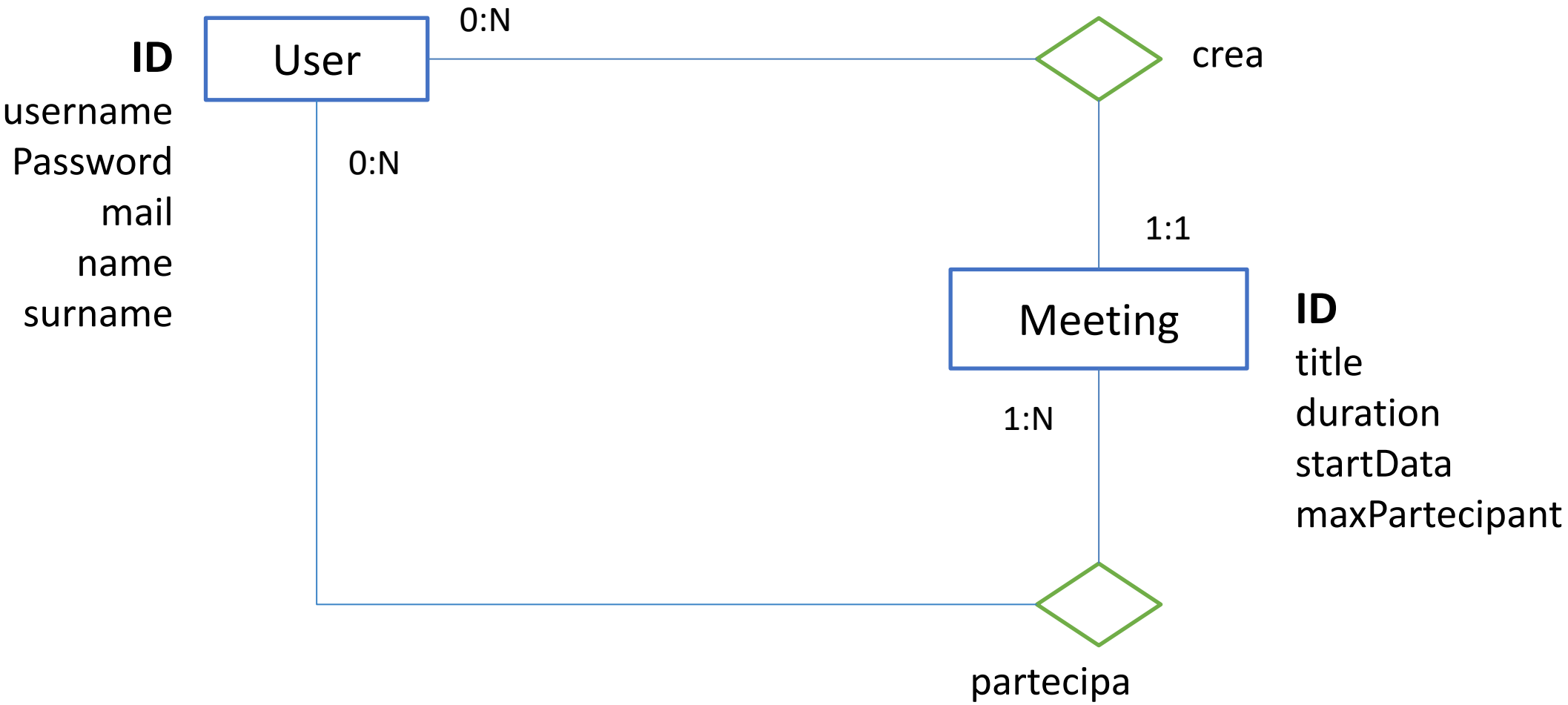
Codice persona: 10681624

Numero matricola: 932922

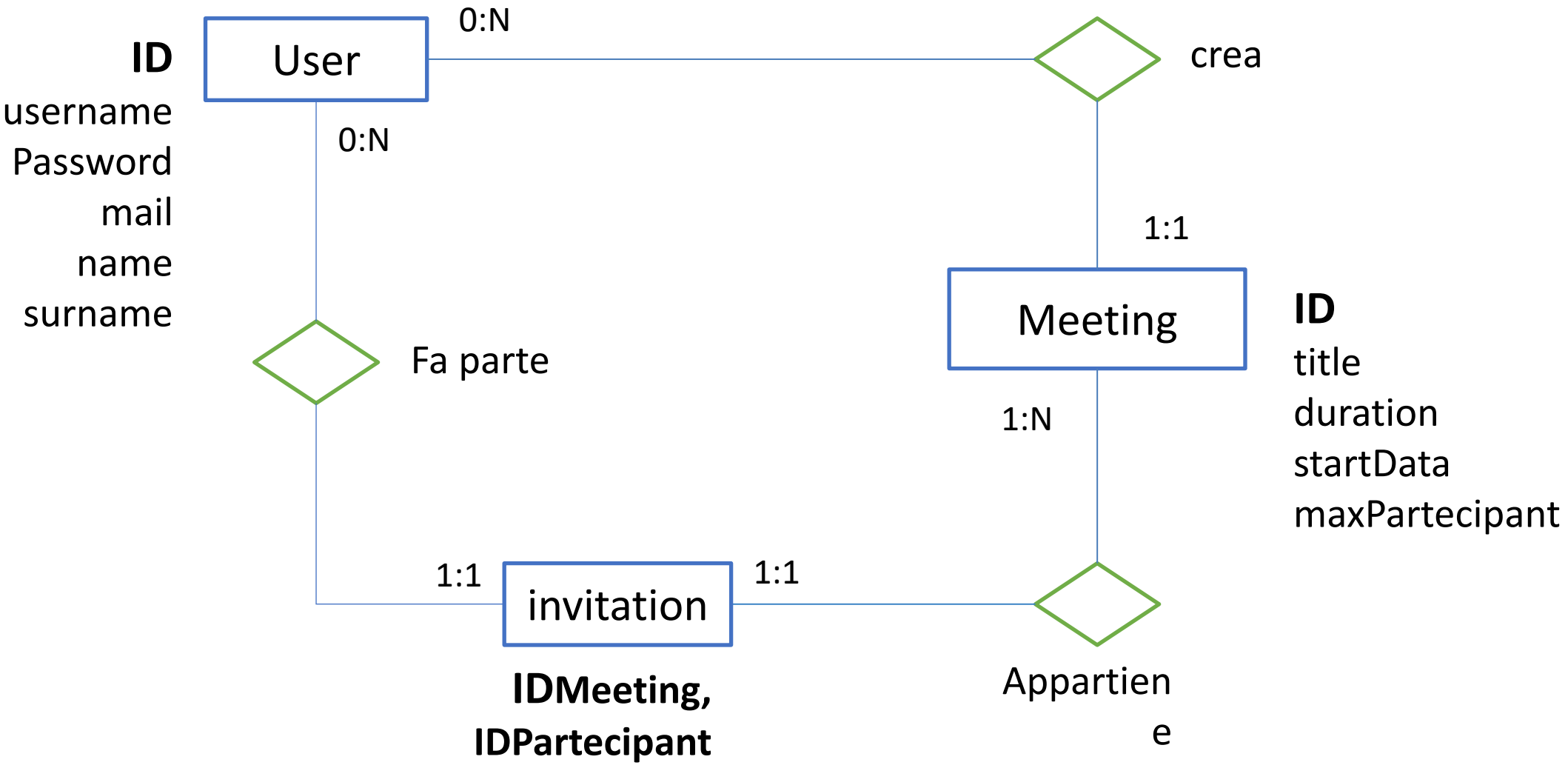
Un’applicazione web consente la gestione di riunioni online. L’applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell’indirizzo di email e l’uguaglianza tra i campi “password” e “ripeti password”. La registrazione 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.

- Entities, attributes, relationships

Database design



Database design



Local database schema

```
CREATE TABLE `user` (
  `iduser` int NOT NULL AUTO_INCREMENT,
  `mail` varchar(45) NOT NULL,
  `password` varchar(20) NOT NULL,
  `name` varchar(30) NOT NULL,
  `surname` varchar(30) NOT NULL,
  PRIMARY KEY (`iduser`)
)
CREATE TABLE `meeting` (
  `idmeeting` int NOT NULL AUTO_INCREMENT,
  `idowner` int NOT NULL,
  `title` varchar(30) NOT NULL,
  `max_partecipant` int NOT NULL,
  `duration` int NOT NULL,
  `date` datetime NOT NULL,
  PRIMARY KEY (`idmeeting`),
  KEY `fk_owner_idx` (`idowner`),
  CONSTRAINT `fk_owner` FOREIGN KEY (`idowner`)
REFERENCES `user` (`iduser`) ON DELETE CASCADE ON
UPDATE CASCADE
)
```

Logical database schema

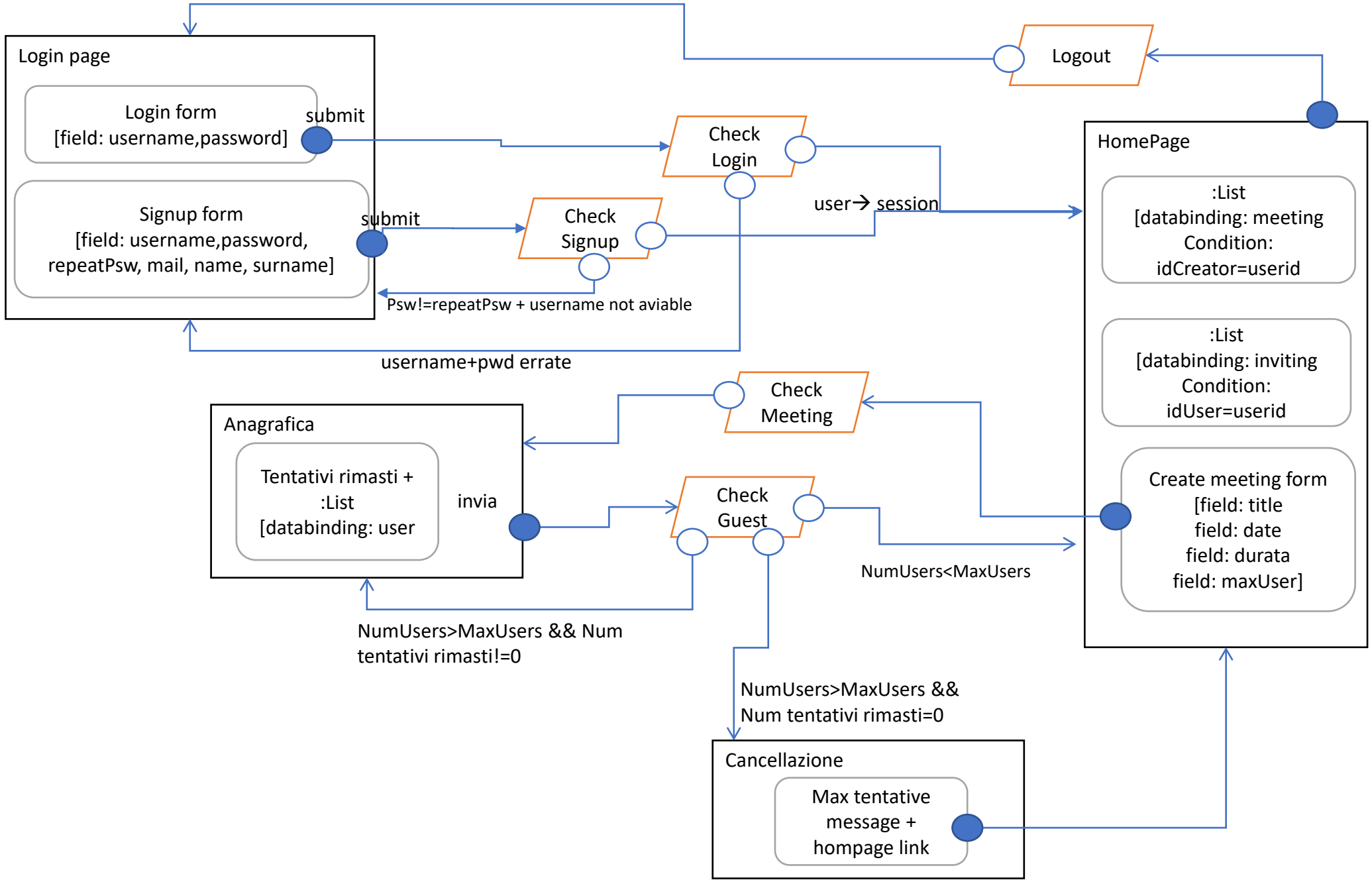
```
CREATE TABLE `invitation` (
  `idmeeting` int NOT NULL,
  `idpartecipant` int NOT NULL,
  PRIMARY KEY
(`idmeeting`,`idpartecipant`),
  KEY `fk_partecipant_idx`
(`idpartecipant`),
  CONSTRAINT `fk_meeting` FOREIGN KEY
(`idmeeting`) REFERENCES `meeting`
(`idmeeting`) ON DELETE CASCADE ON
UPDATE CASCADE,
  CONSTRAINT `fk_partecipant` FOREIGN
KEY (`idpartecipant`) REFERENCES `user`
(`iduser`) ON DELETE CASCADE ON UPDATE
CASCADE
)
```

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.

Pages (views), view components, events, actions

- La pagina di default contiene la form di login e di signup
- Tutti i dati dei form sono obbligatori
- Non possono esistere attributi negativi(duratà, massimo numero di partecipanti…)
- I meeting non possono essere creati con date precedenti a quella odierna

Application design



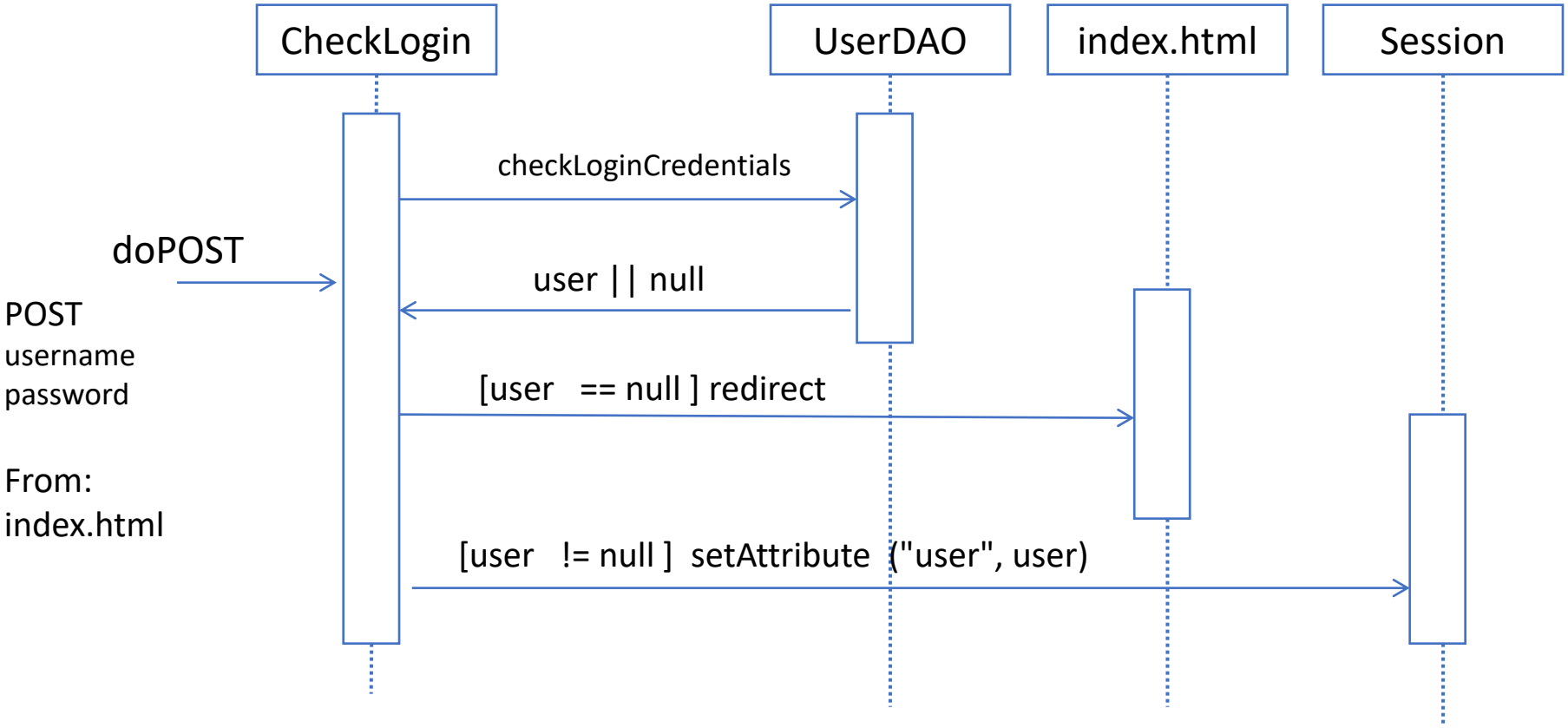
Components

- Model objects (Beans)
  - User
  - Meeting
  - Invitation

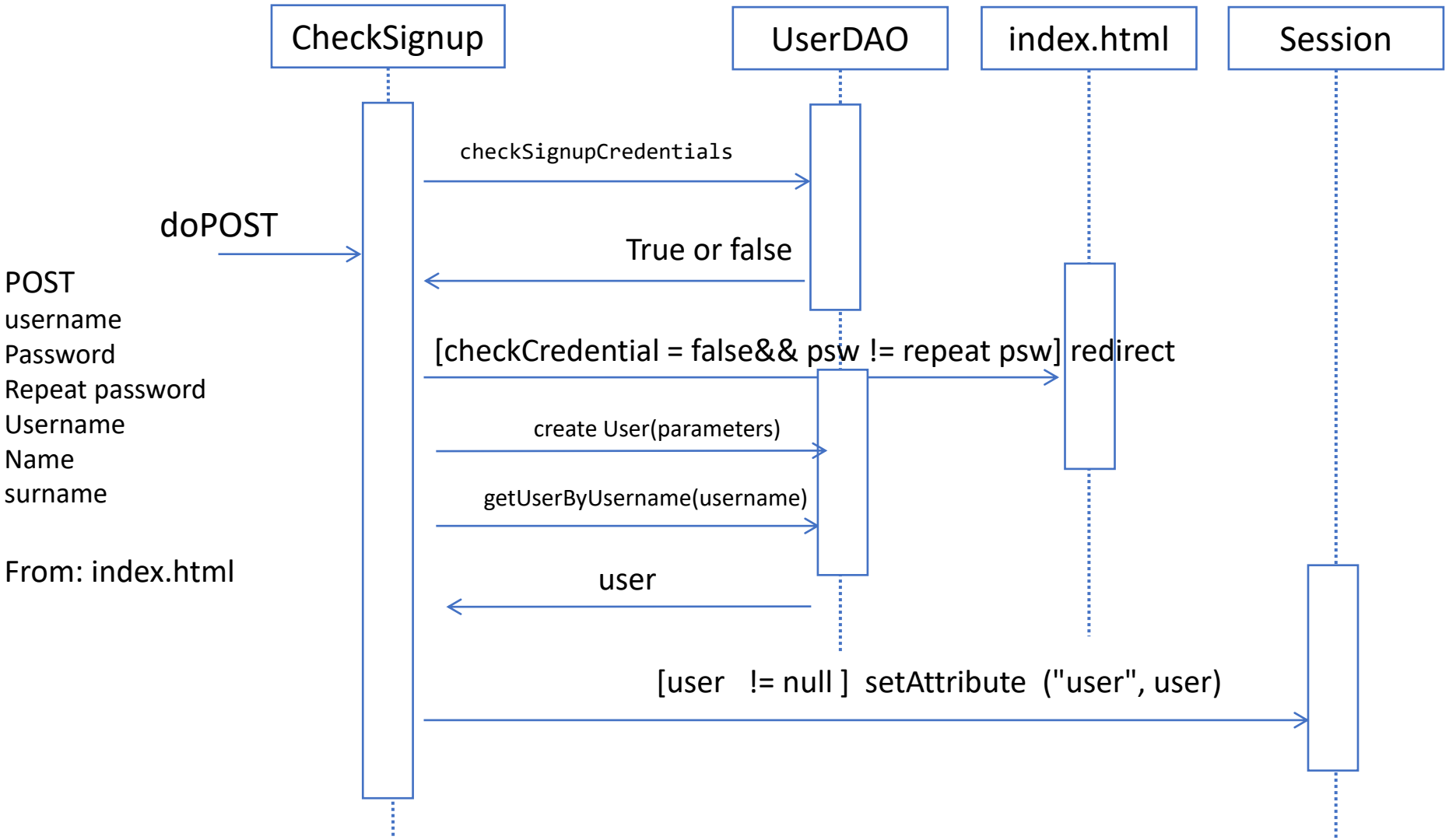
- Data Access Objects (Classes)
  - UserDAO
    - checkLoginCredentials(user, spw)
    - checkSignupCredentials(user)
    - findAllUsers()
    - getUserByUsername(user)
    - getUser(id)
  - MeetingDAO
    - findMeetingByCreator(id)
    - findLastMeetingByCreator(id)
    - findMeetingByGuest(id)
    - createMeeting(title, startDate, duration, maxGuests, creator)
  - InvitationDAO
    - createInvitation(idMeeting, idGuest);

- Controllers (servlets)
  - CheckLogin
  - CheckSignup
  - Logout
  - GoToHome
  - GoToAnagrafe
  - CreateMeeting
  - CheckGuest
- Views (Templates) & components
  - Login
    - Login form
  - Home
    - Created meeting list
    - Invited meeting list
    - Creation form
  - Anagrafe
    - User list form
    - Max guest message
    - Number of attempts message
  - Cancellazione
    - Message
    - Link to home

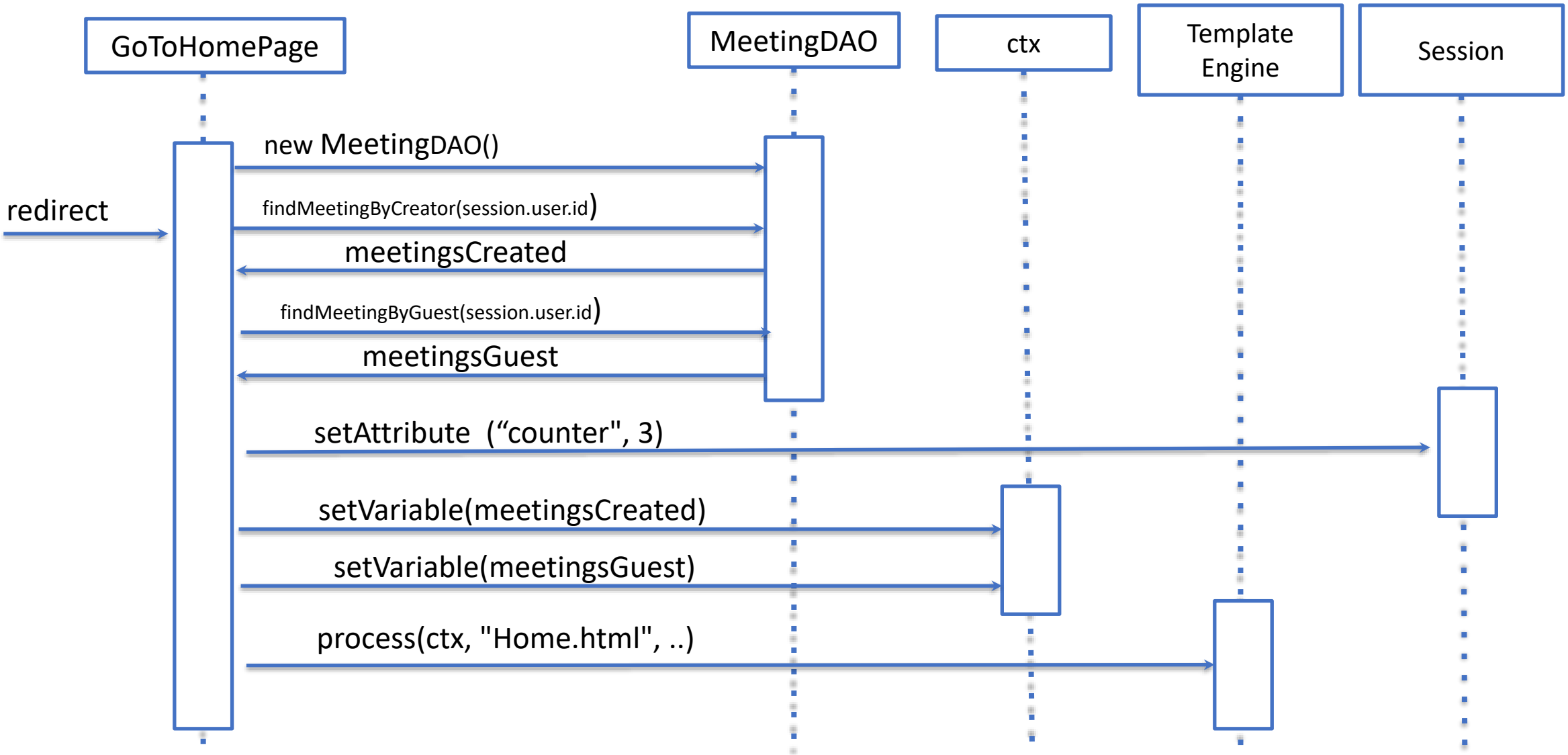
Event: Login


CheckLogin
UserDAO
index.html
Session
checkLoginCredentials
doPOST
user || null
POST
username
password
[user == null] redirect
From:
index.html
[user != null] setAttribute ("user", user)

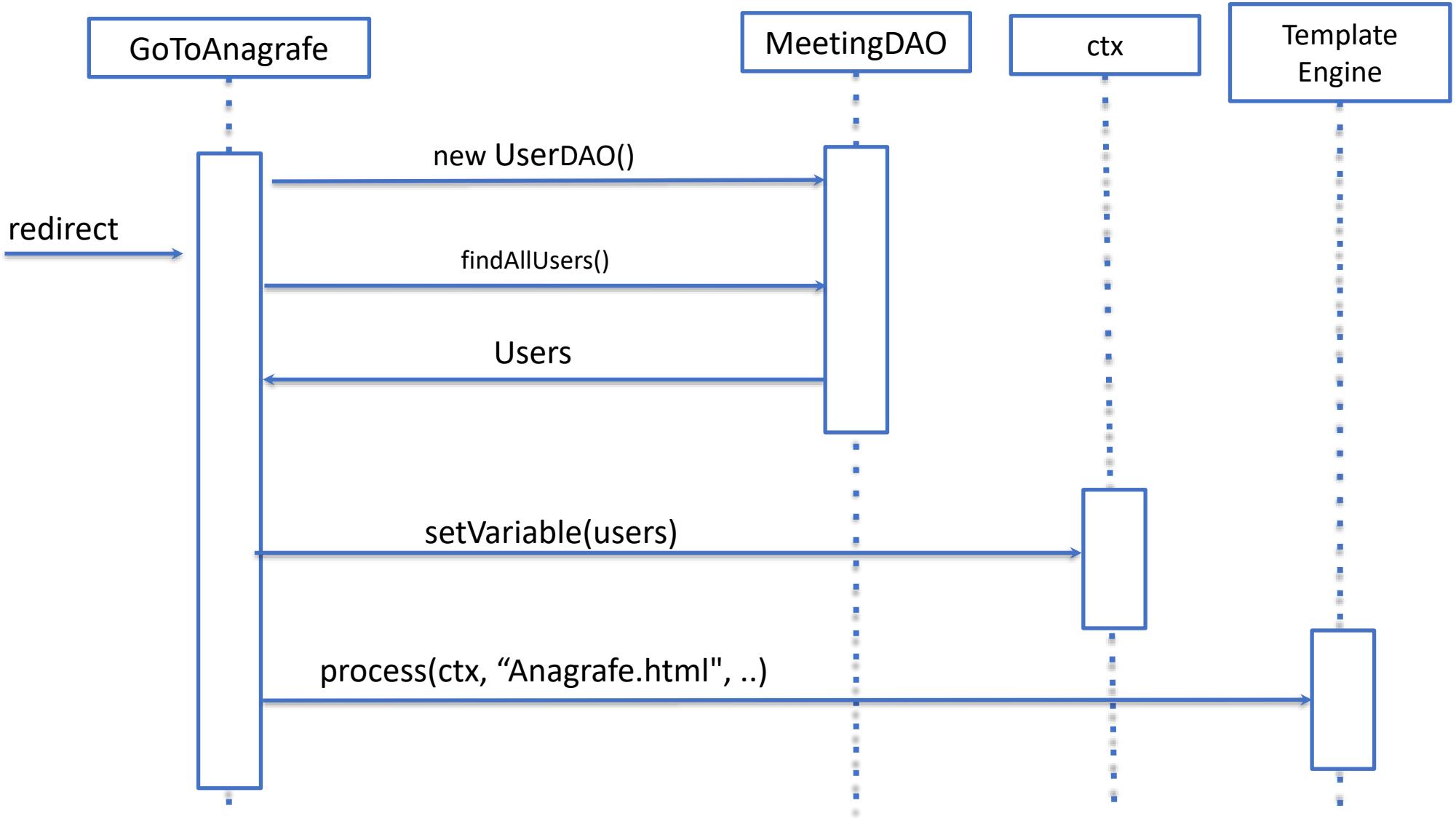
Event: Signup



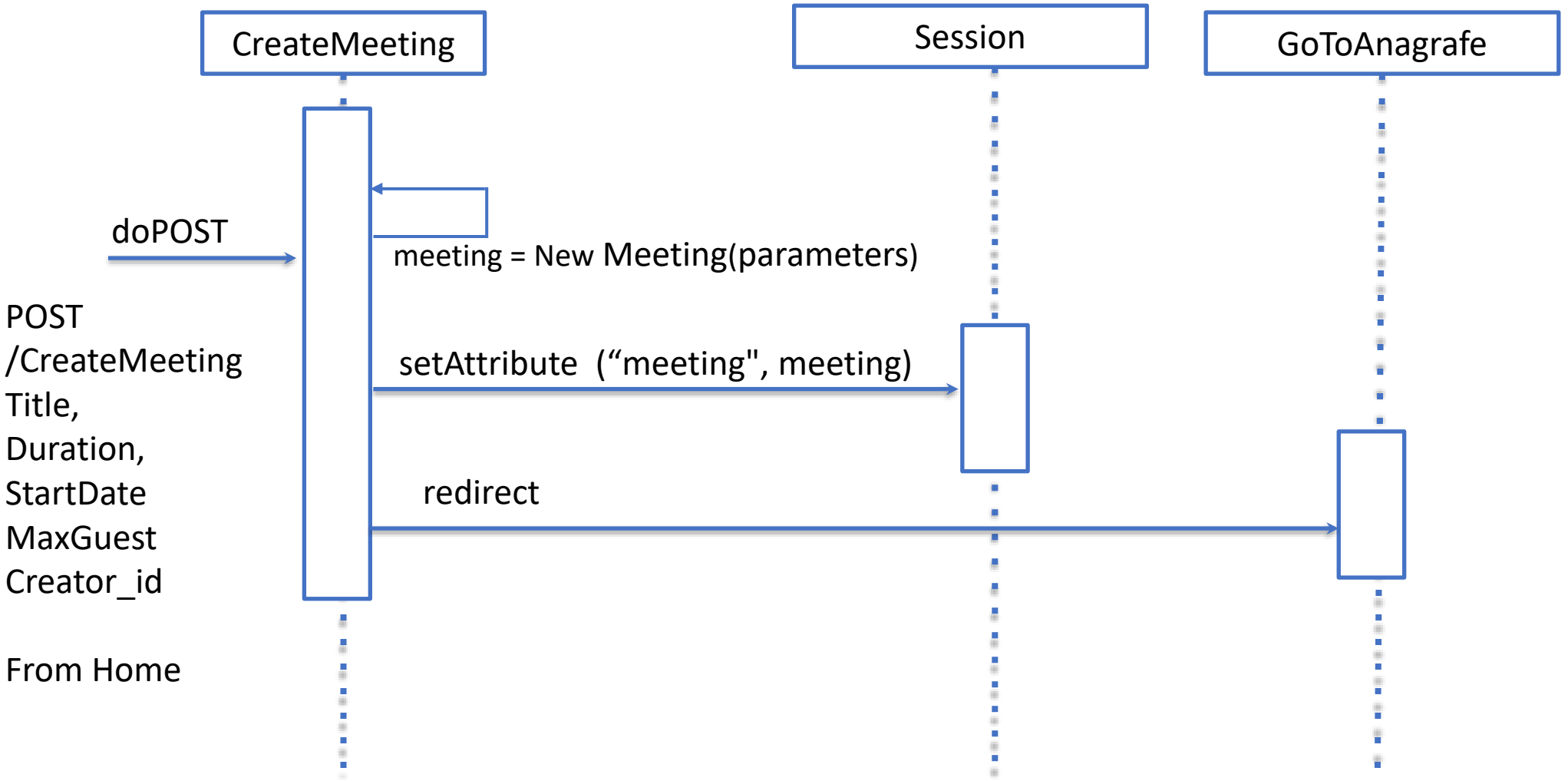
Event: go to Home



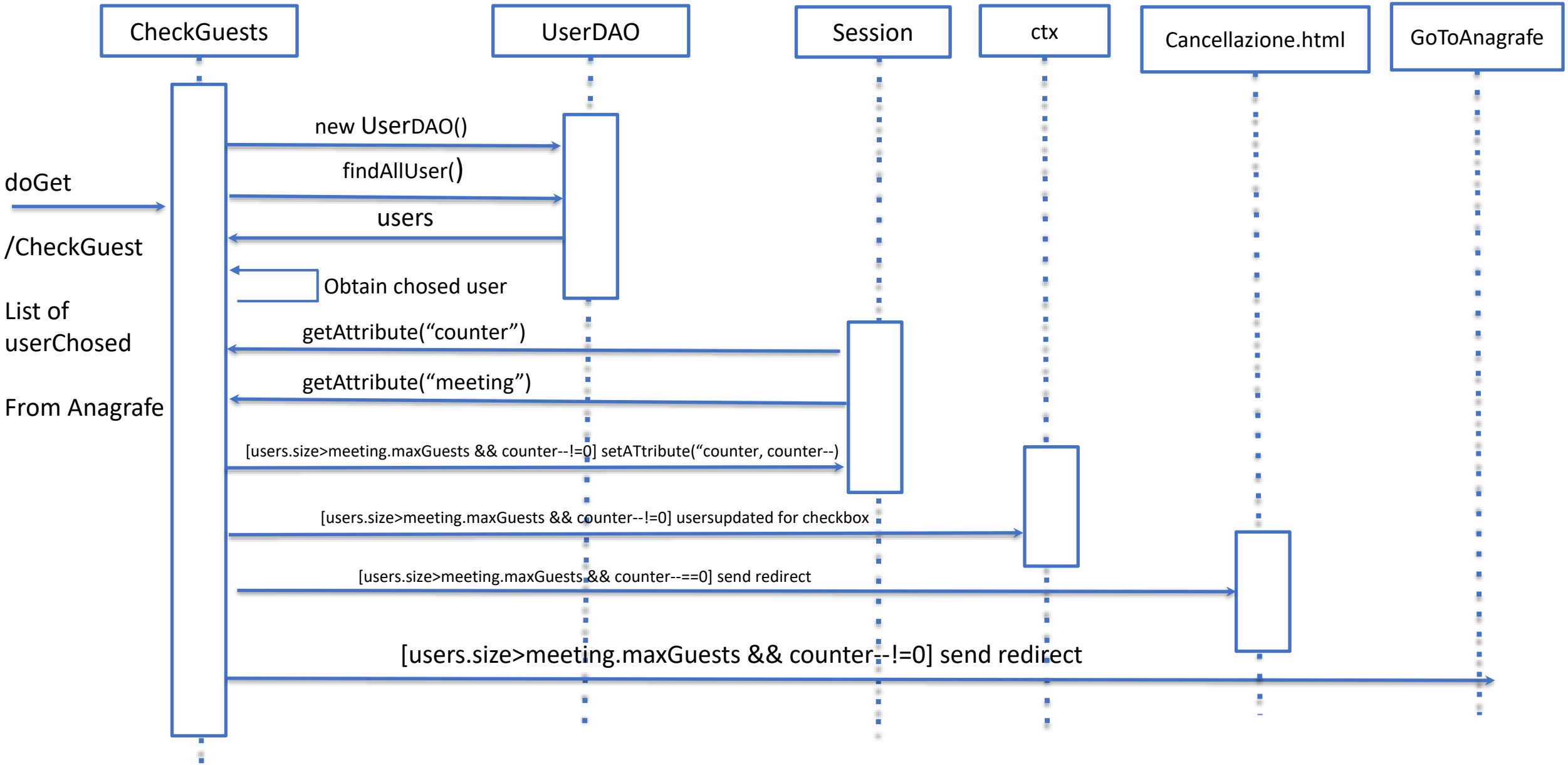
Event: go to Anagrafe



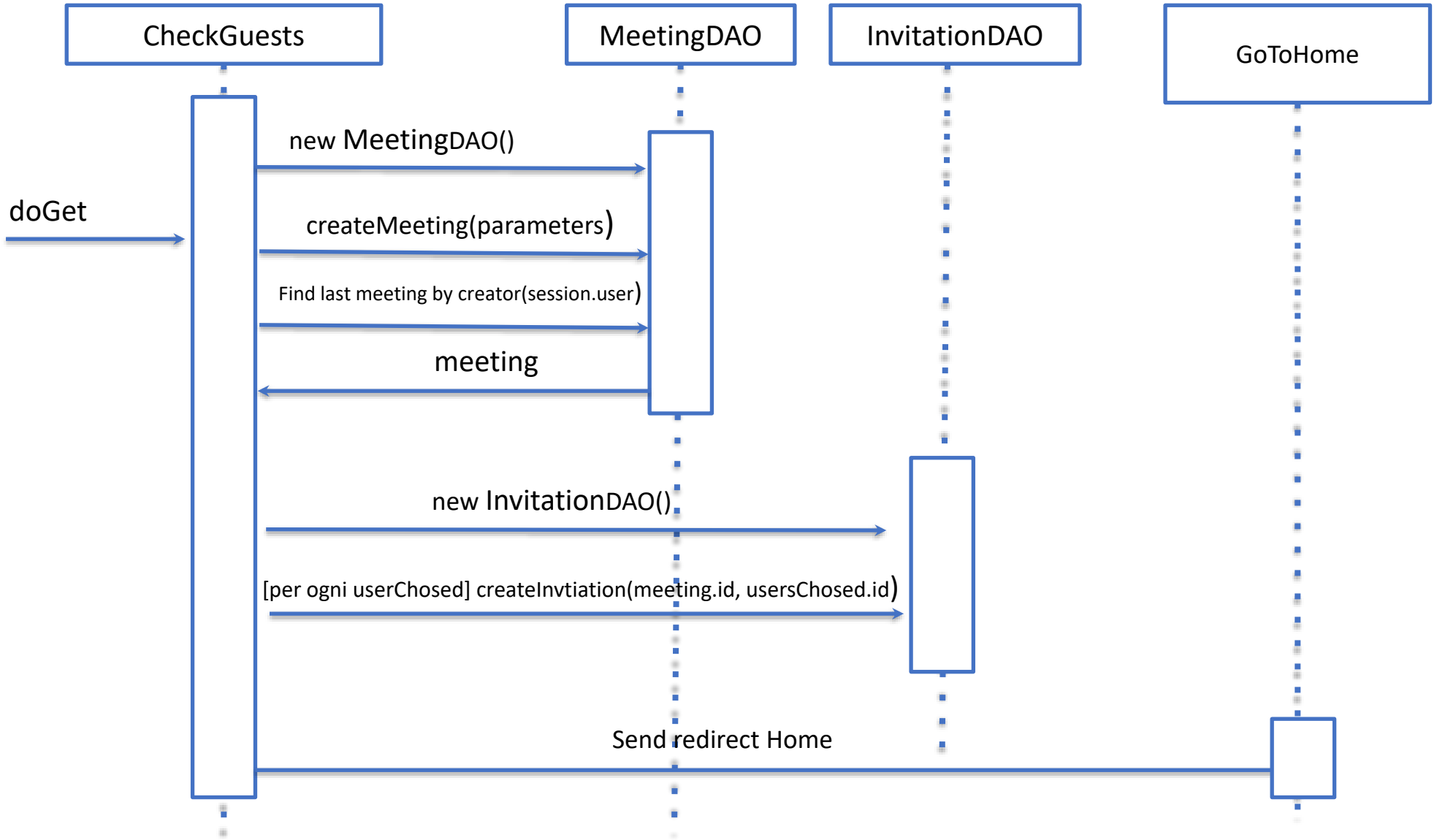
Event: CreateMeeting



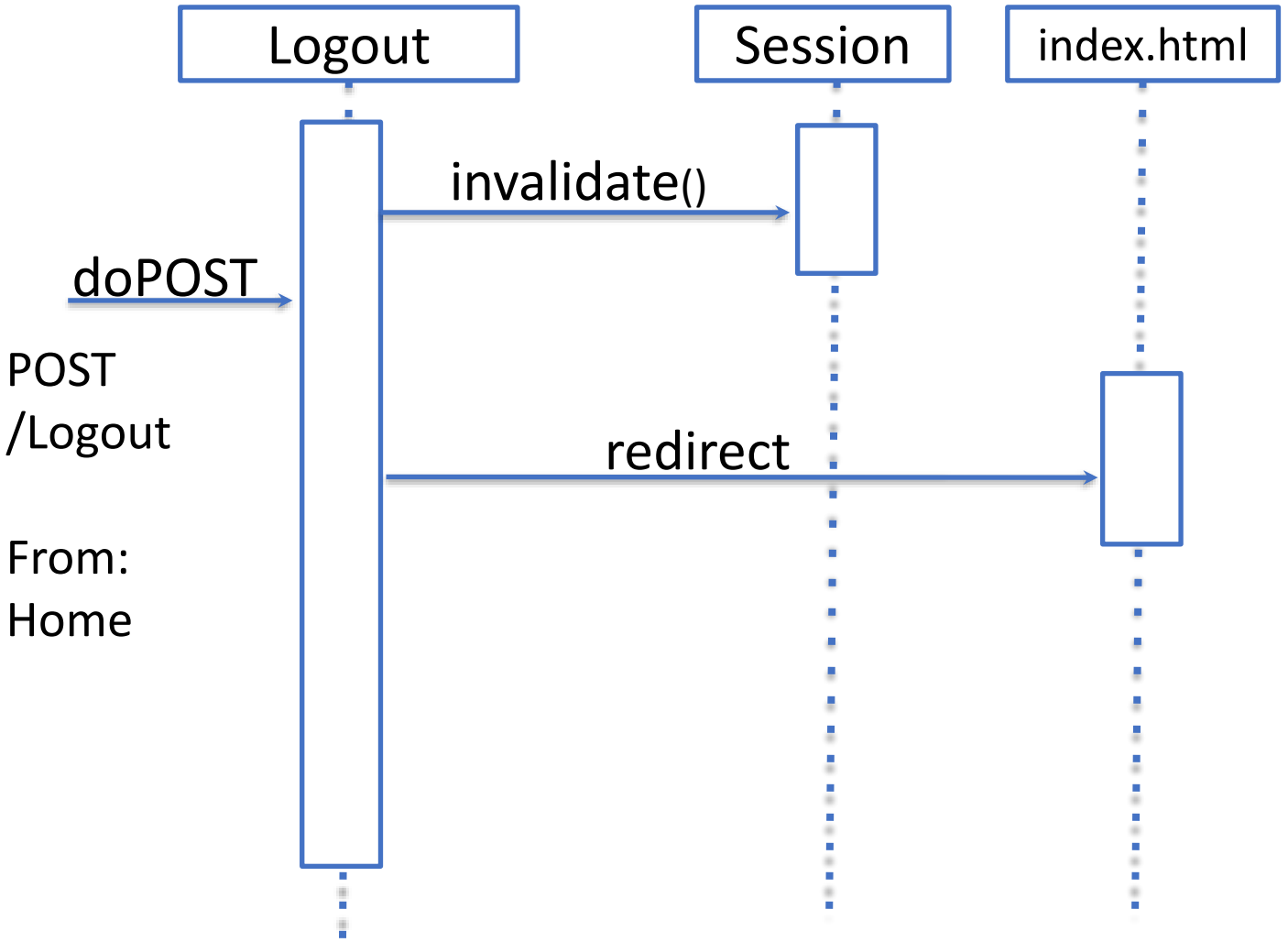
Event: CheckGuests



Event: CheckGuests if users.size<meeting.maxGuests



Event: logout



Versione con JavaScript

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.

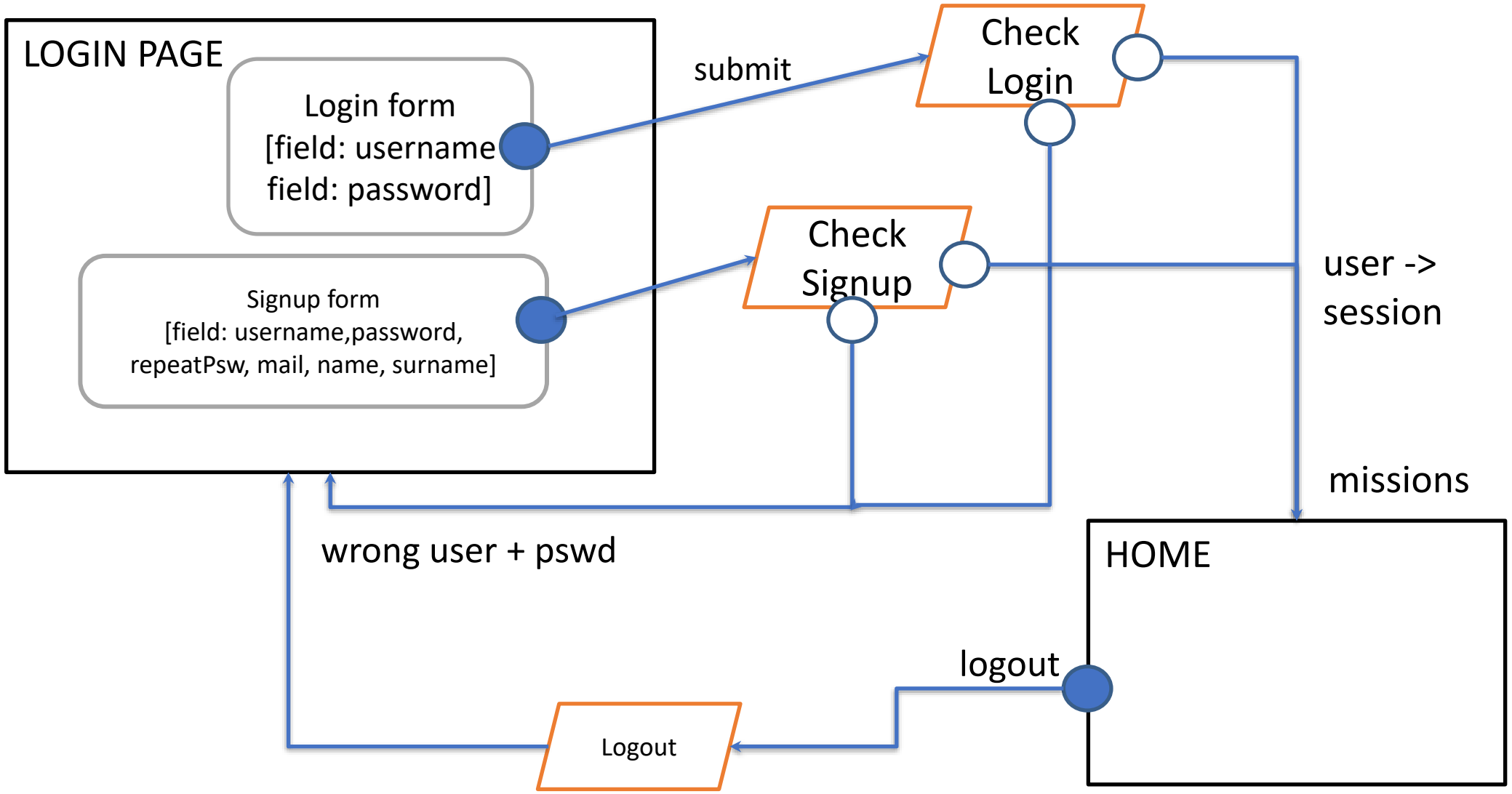
Components

- Model objects (Beans)
  - User
  - Meeting
  - Invitation

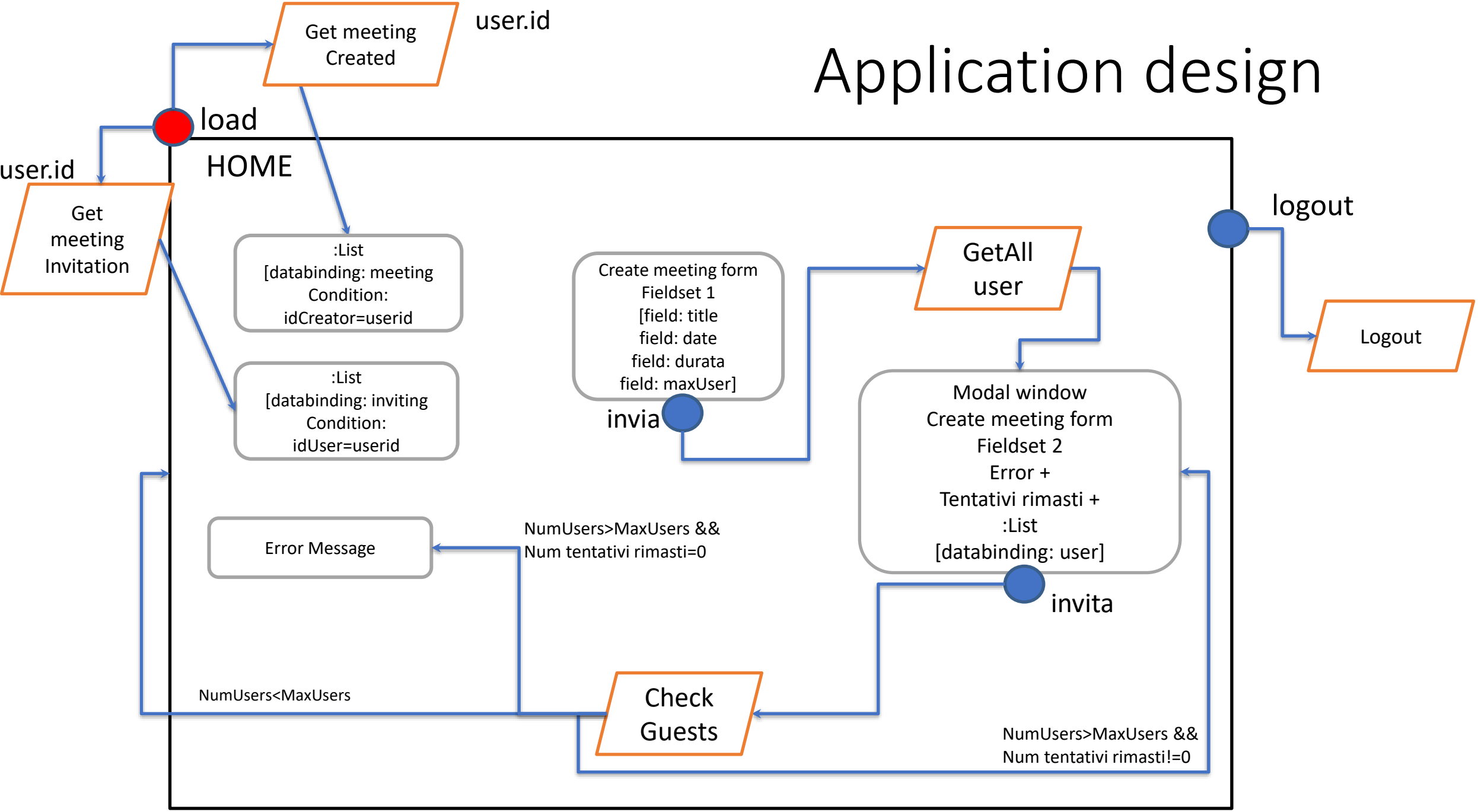
- Data Access Objects (Classes)
  - UserDAO
    - checkLoginCredentials(user, spw)
    - checkSignupCredentials(user)
    - findAllUsers()
    - getUserByUsername(user)
    - getUser(id)
  - MeetingDAO
    - findMeetingByCreator(id)
    - findLastMeetingByCreator(id)
    - findMeetingByGuest(id)
    - createMeeting(title, startDate, duration, maxGuests, creator)
  - InvitationDAO
    - createInvitation(idMeeting, idGuest);

- Controllers (servlets)
  - CheckLogin
  - CheckSignup
  - Logout
  - GetAllUser
  - GetMeetingCreated
  - GetMeetingInvitation
  - CheckGuest
- Views (Templates) & components
  - Login
    - Login form
  - Home
    - Created meeting list
    - Invited meeting list
    - Creation form
    - Modal Window
      - User list form
      - Max guest message
      - Number of attempts message

Application design

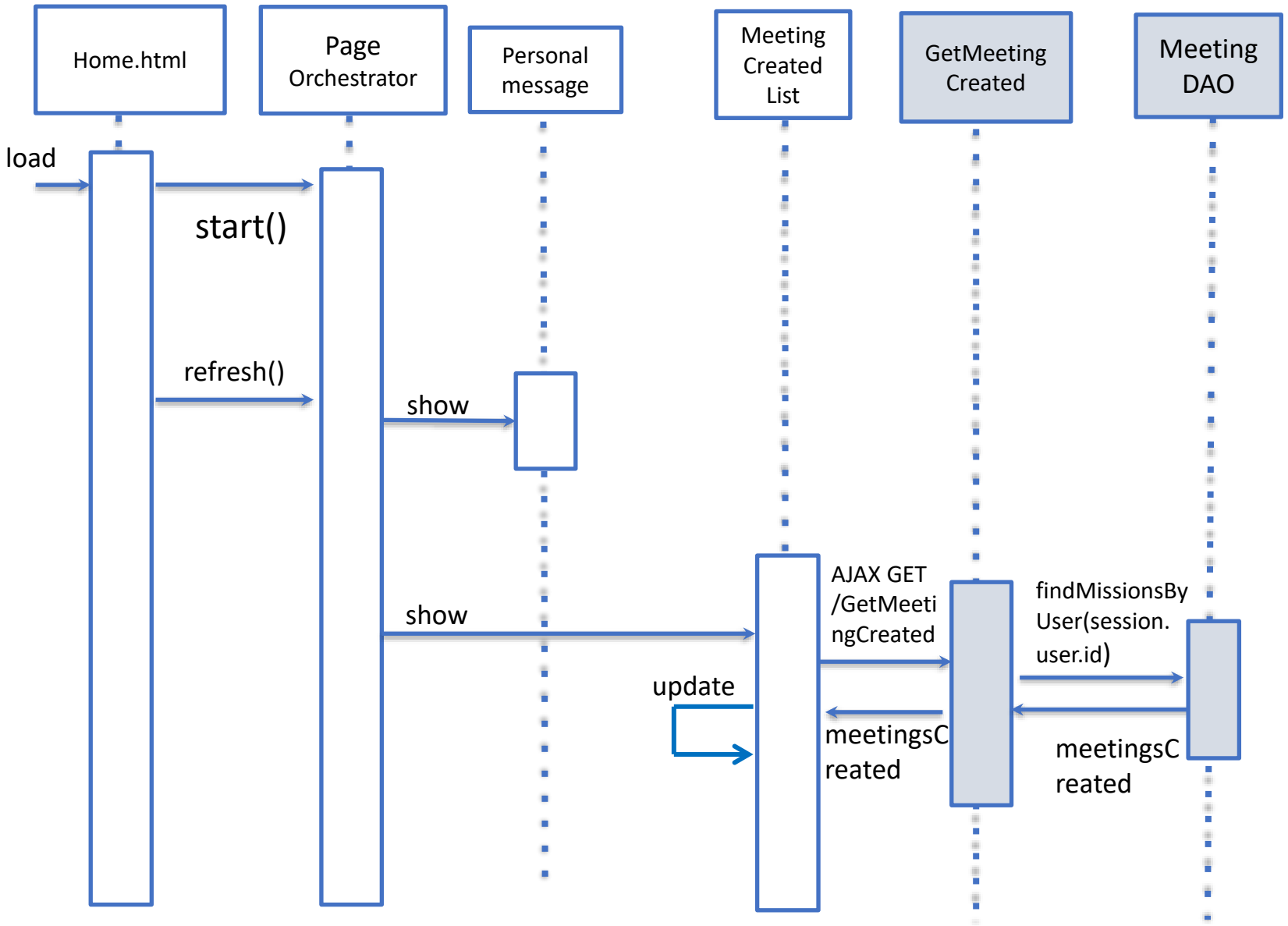


Application design

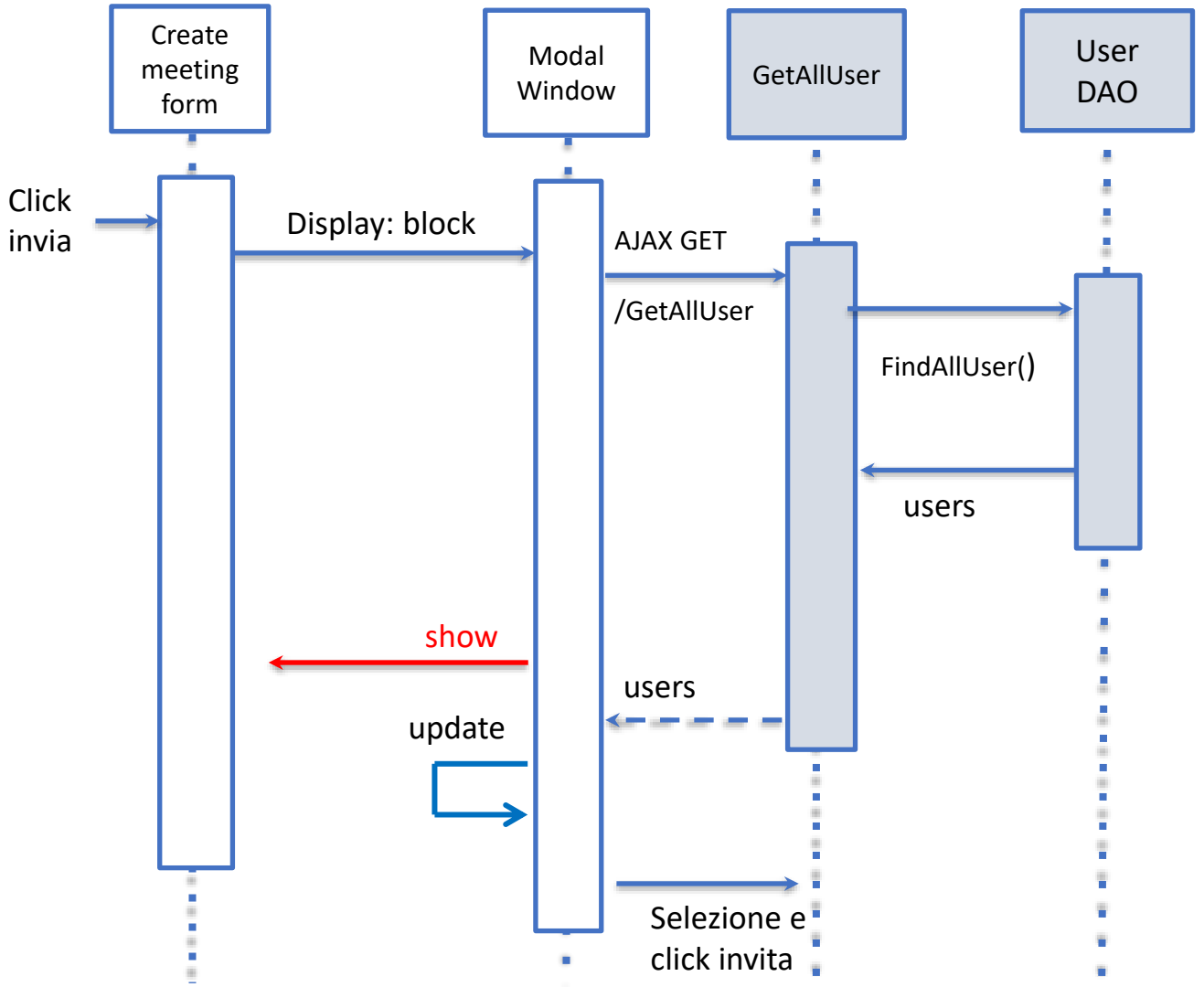


Evento: caricamento Home

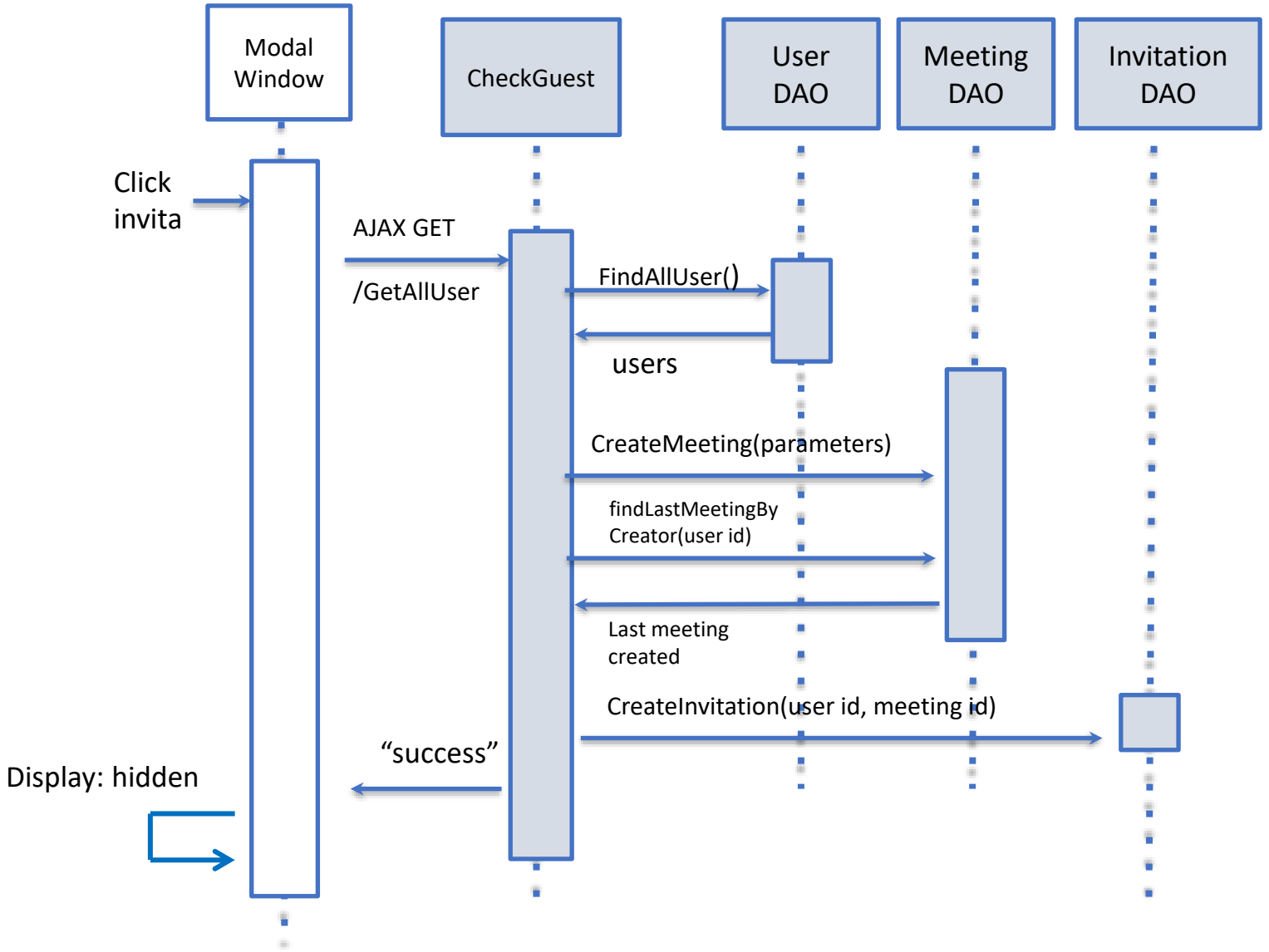
n.b. la visione della lista di meeting create e della lista degli inviti ai meeting viene visualizzata con la stessa logica, ma con servlet di nomi diversi



Evento: invia - create meeting form



Evento: invita - modal window



Evento: logout