

# Esercizio 5: riunioni online

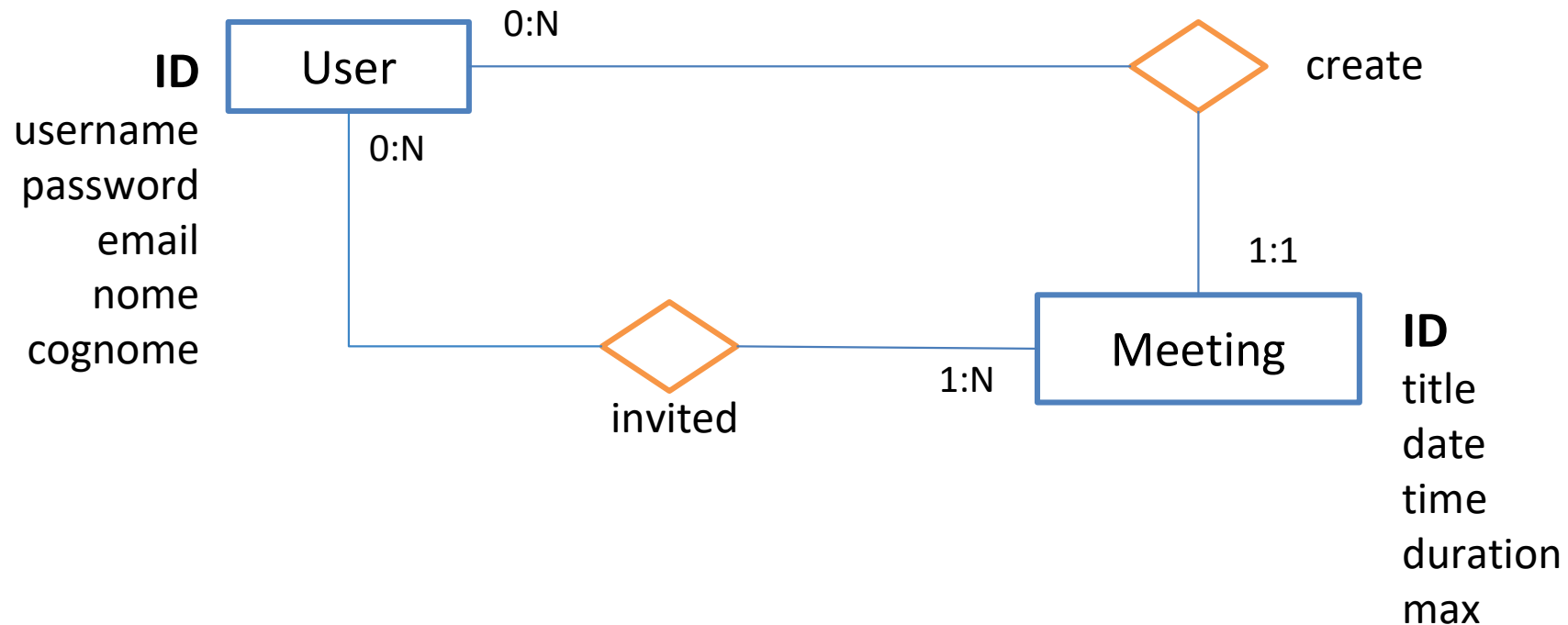
Matteo Gobbi Frattini

# Analisi dei dati

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.

**Entità, attributi, relazioni**

# Database design



# Local database schema

```
CREATE TABLE `user` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `username` VARCHAR(45) NOT NULL,  
  `email` VARCHAR(45) NOT NULL,  
  `password` VARCHAR(45) NOT NULL,  
  `name` VARCHAR(45) NOT NULL,  
  `surname` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id`));
```

```
CREATE TABLE `meeting` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `title` VARCHAR(45) NOT NULL,  
  `duration` INT NOT NULL,  
  `dateTime` DATETIME NOT NULL,  
  `max` INT NOT NULL,  
  `creator` INT NOT NULL, PRIMARY KEY  
  (`id`),  
  CONSTRAINT `creator`  
  FOREIGN KEY (`creator`)  
  REFERENCES `user`(`id`)  
  ON DELETE CASCADE ON UPDATE CASCADE);
```

# Local database schema

```
CREATE TABLE `invitation`  
  (`user_id` INT NOT NULL,  
   `meeting_id` INT NOT NULL,  
   PRIMARY KEY (`user_id`, `meeting_id`),  
   CONSTRAINT `user_id`  
   FOREIGN KEY (`user_id`)  
   REFERENCES `user` (`id`)  
   ON DELETE CASCADE      ON UPDATE CASCADE,  
   CONSTRAINT `meeting_id`  
   FOREIGN KEY (`meeting_id`)  
   REFERENCES `meeting` (`id`)  
   ON DELETE CASCADE      ON UPDATE CASCADE);
```

# Application requirements analysis

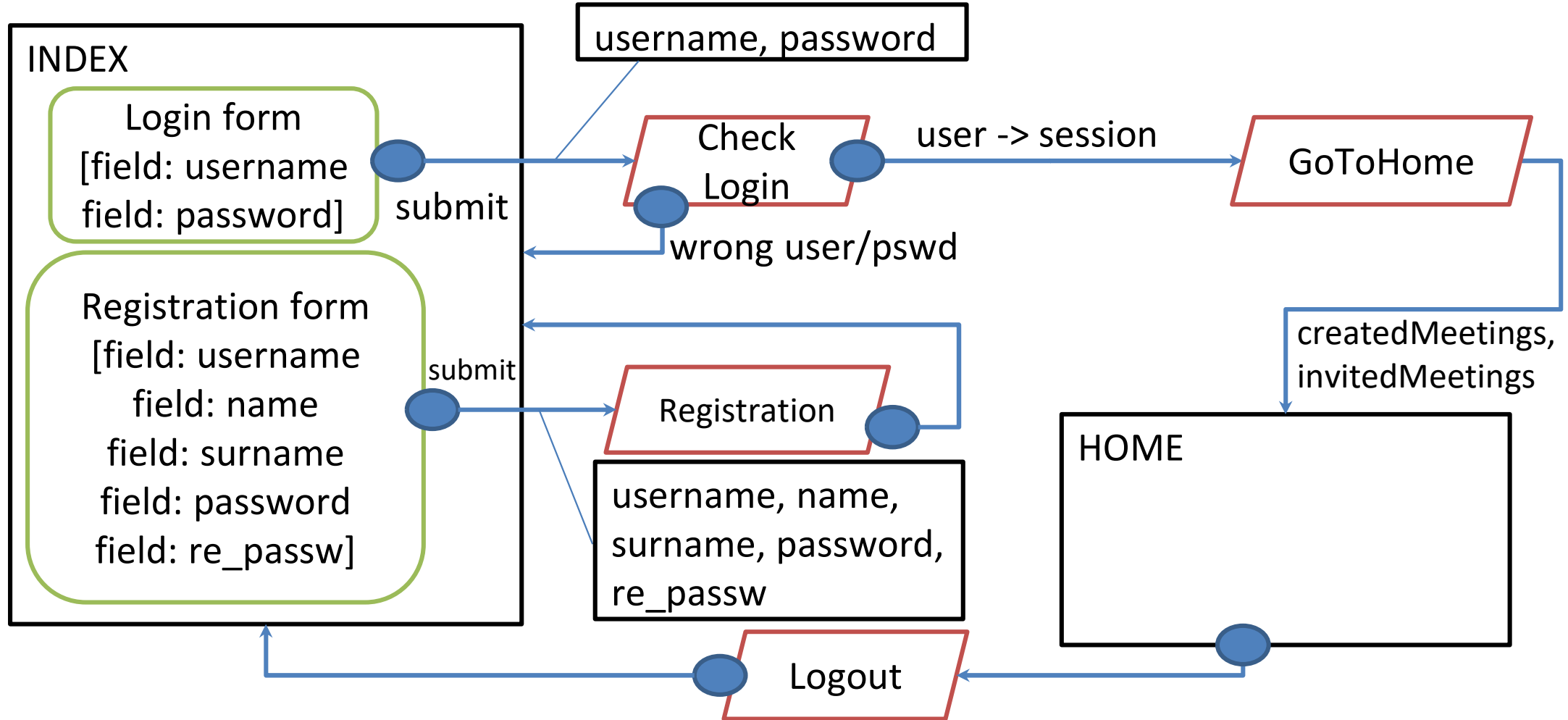
Un'applicazione web consente la gestione di riunioni online. L'applicazione supporta **registrazione(form)** e **login(form)** mediante una **pagina pubblica con opportune form**. La **registrazione** controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti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

# Completamento delle specifiche

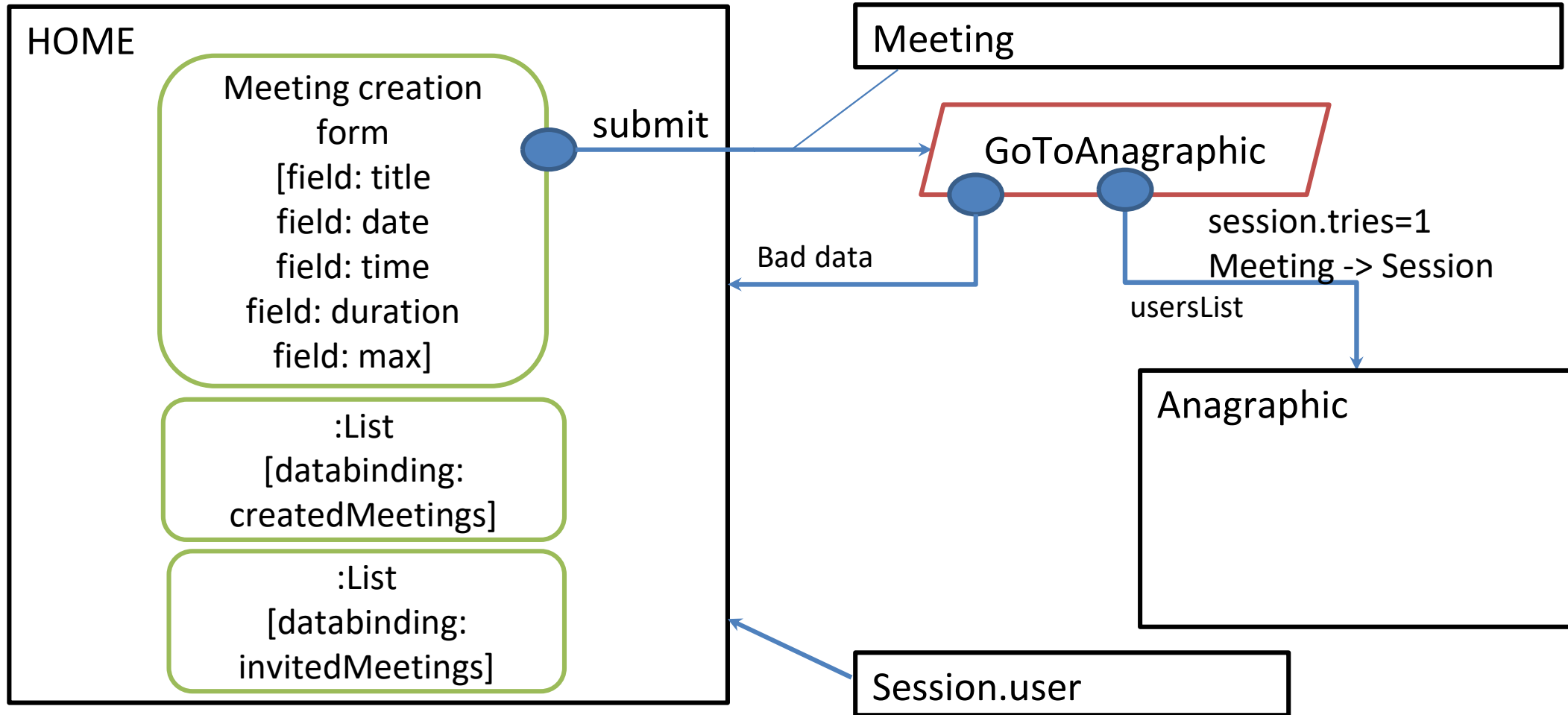
- La registrazione controlla anche l'unicità della mail
- Sia per la registrazione che per la creazione di una nuova riunione tutti i dati sono obbligatori
- Non è possibile creare una riunione con inizio antecedente al momento attuale
- La pagina ANAGRAFICA mostra tutti gli utenti registrati tranne l'utente loggato, che in quanto creatore non ha bisogno di essere invitato
- Essendo il campo “numero massimo di partecipanti” e non “numero massimo di invitati” al numero di invitati si aggiunge 1, il creatore

# Application design

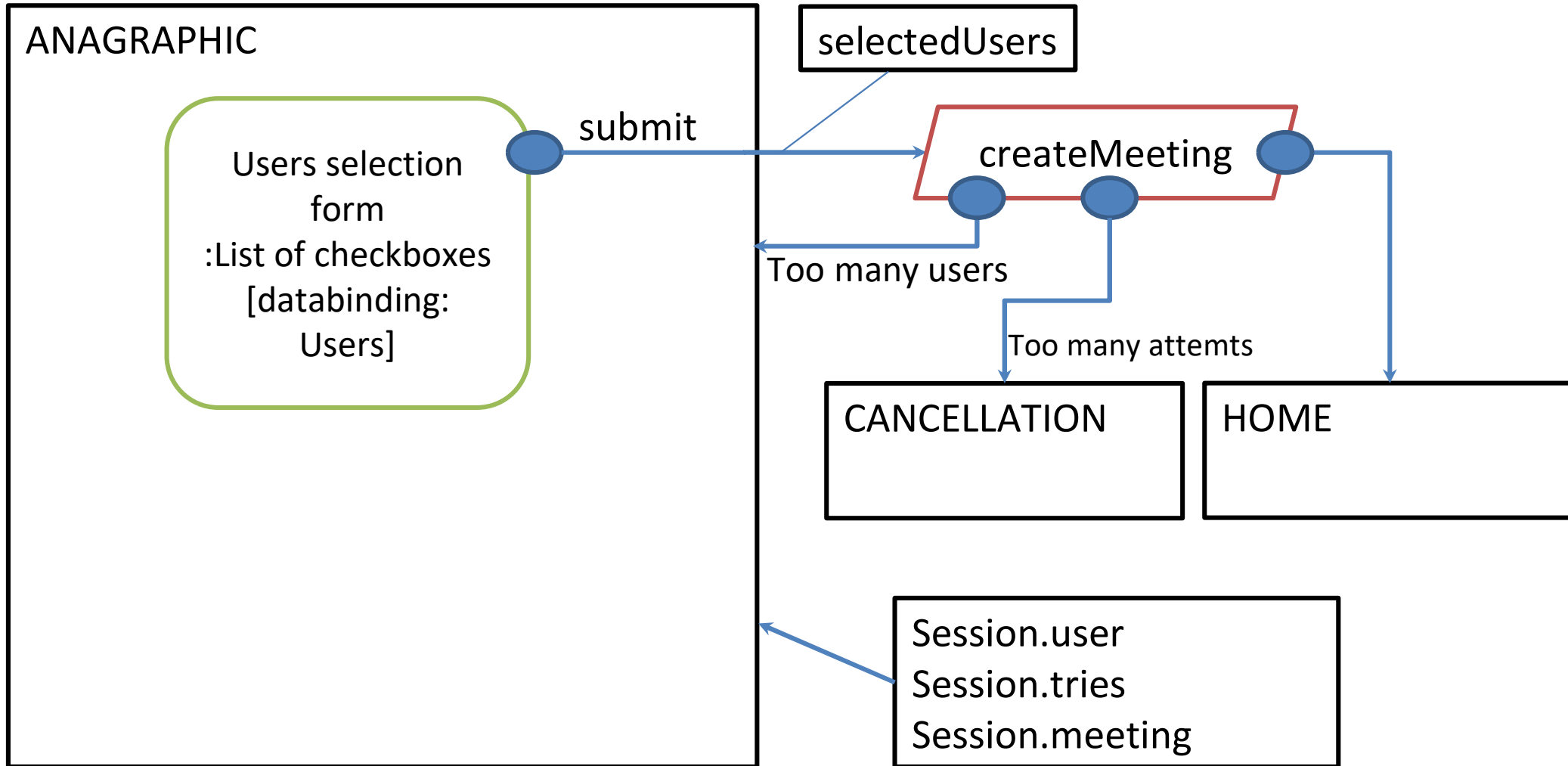




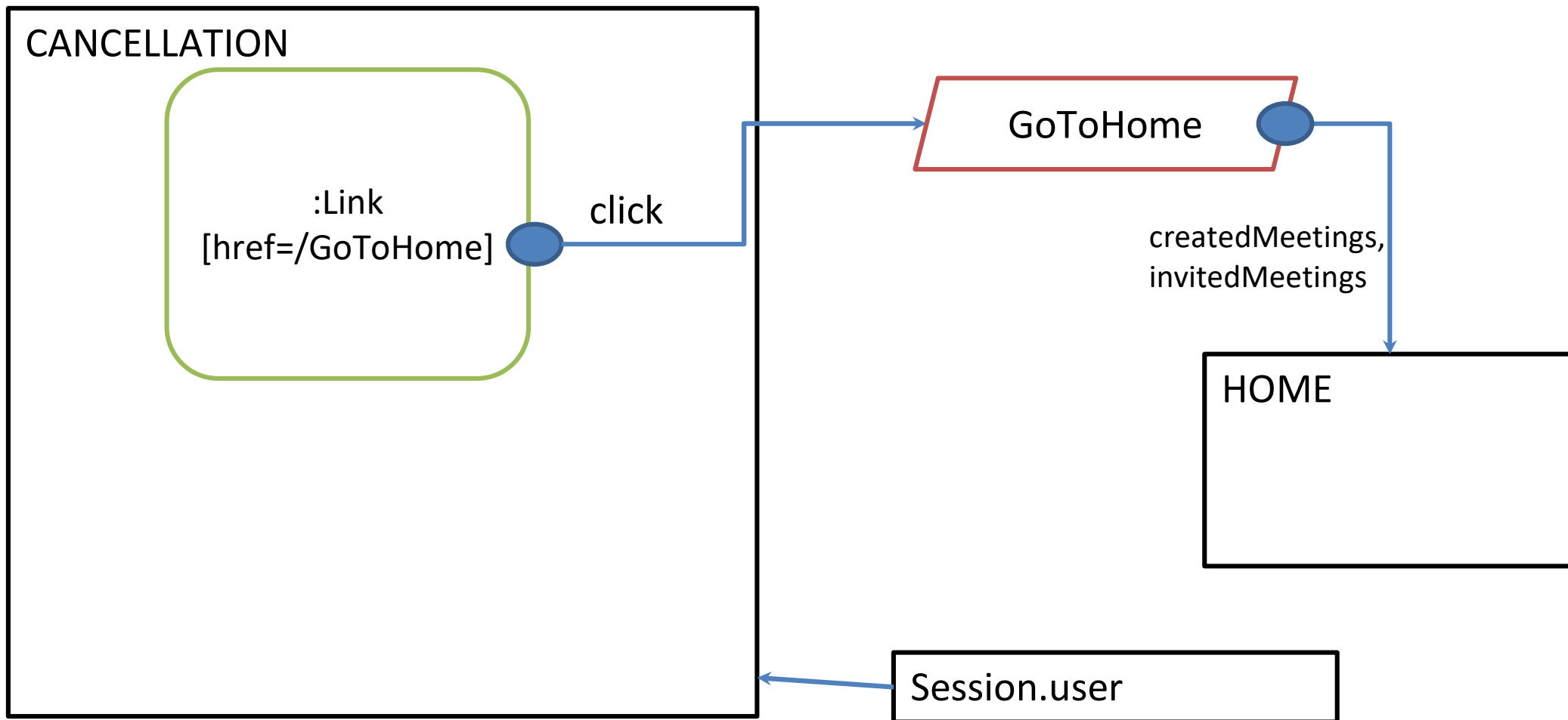
# Application design



# Application design



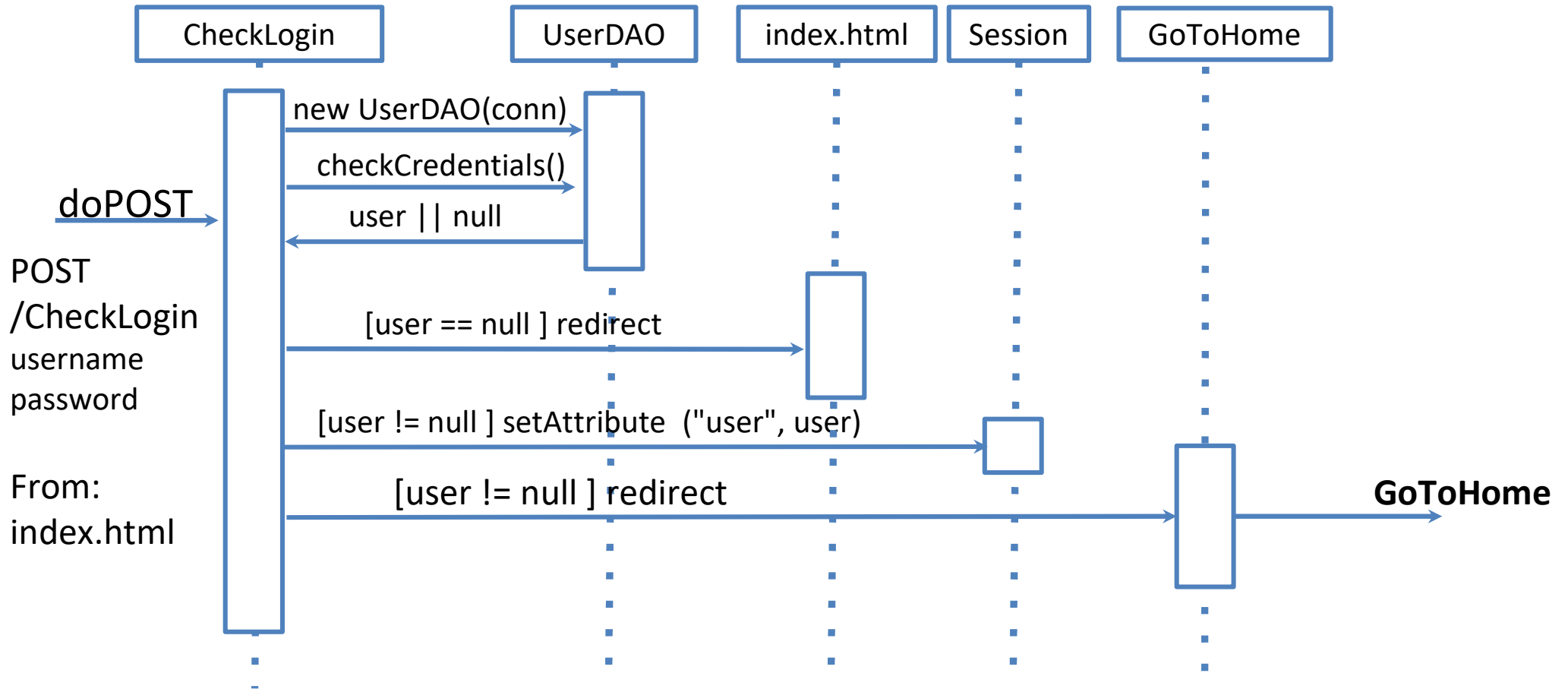
# Application design



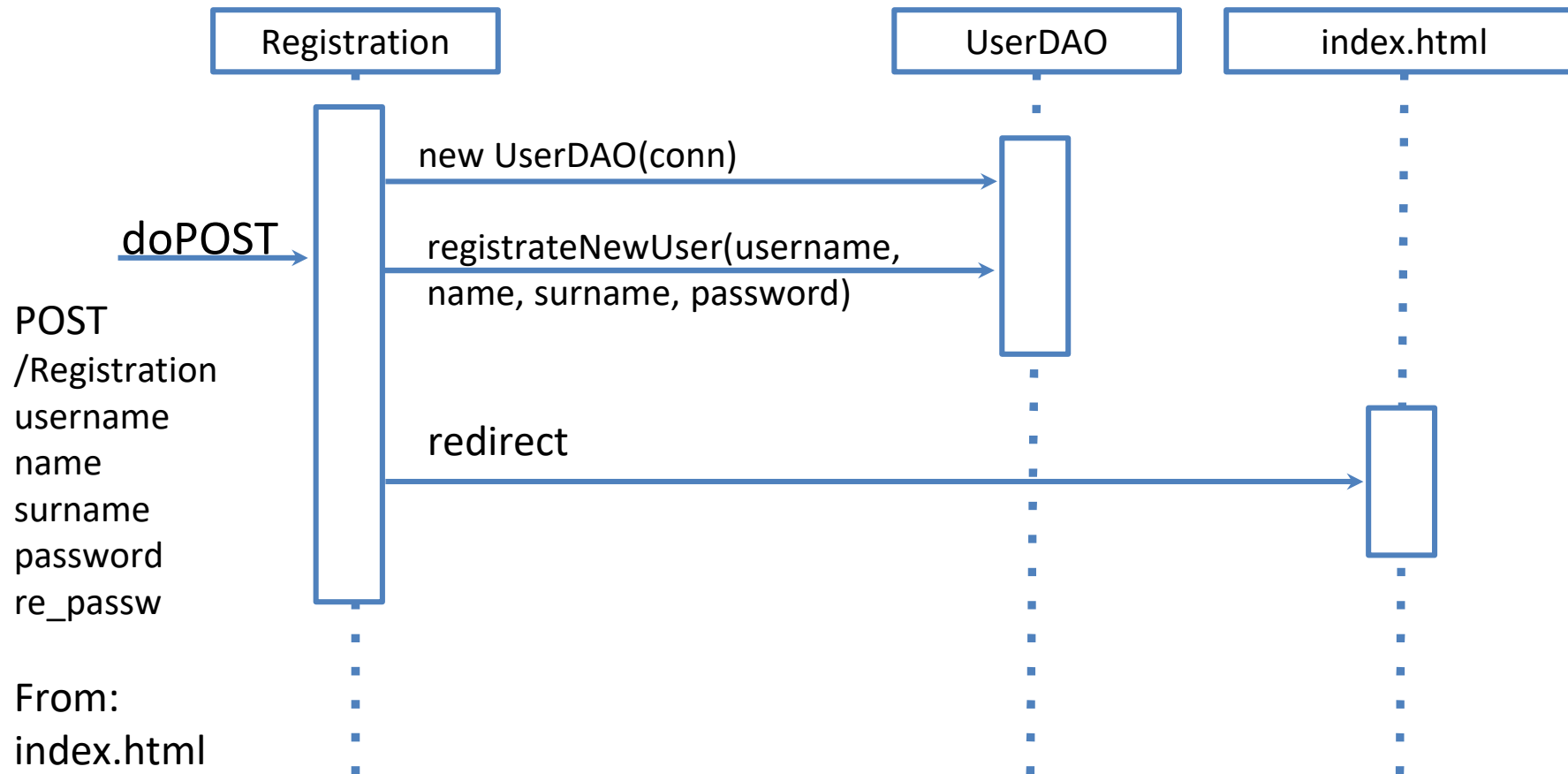
# Components

- Model objects (Beans)
  - User
  - Meeting
- Data Access Objects (Classes)
  - UserDao
    - checkCredentials(username, pwd)
    - isUsernameTaken(username)
    - isEmailTaken(email)
    - registerNewUser(user, password)
    - listAllUsersButOne(id)
  - MeetingDAO
    - createNewMeeting(meeting, creatorId, selectedUsers)
    - getFutureMeetingsByCreator(creatorId)
    - getFutureInvitationsByUser(userId)
  - InvitationDAO
    - addInvitation(userId, meetingId)
- Controllers (servlets)
  - CheckLogin
  - Registration
  - Logout
  - GoToHome
  - GoToAnagraphic
  - CreateMeeting
- Views (Templates)
  - index
  - Home
  - Anagraphic
  - Cancellation

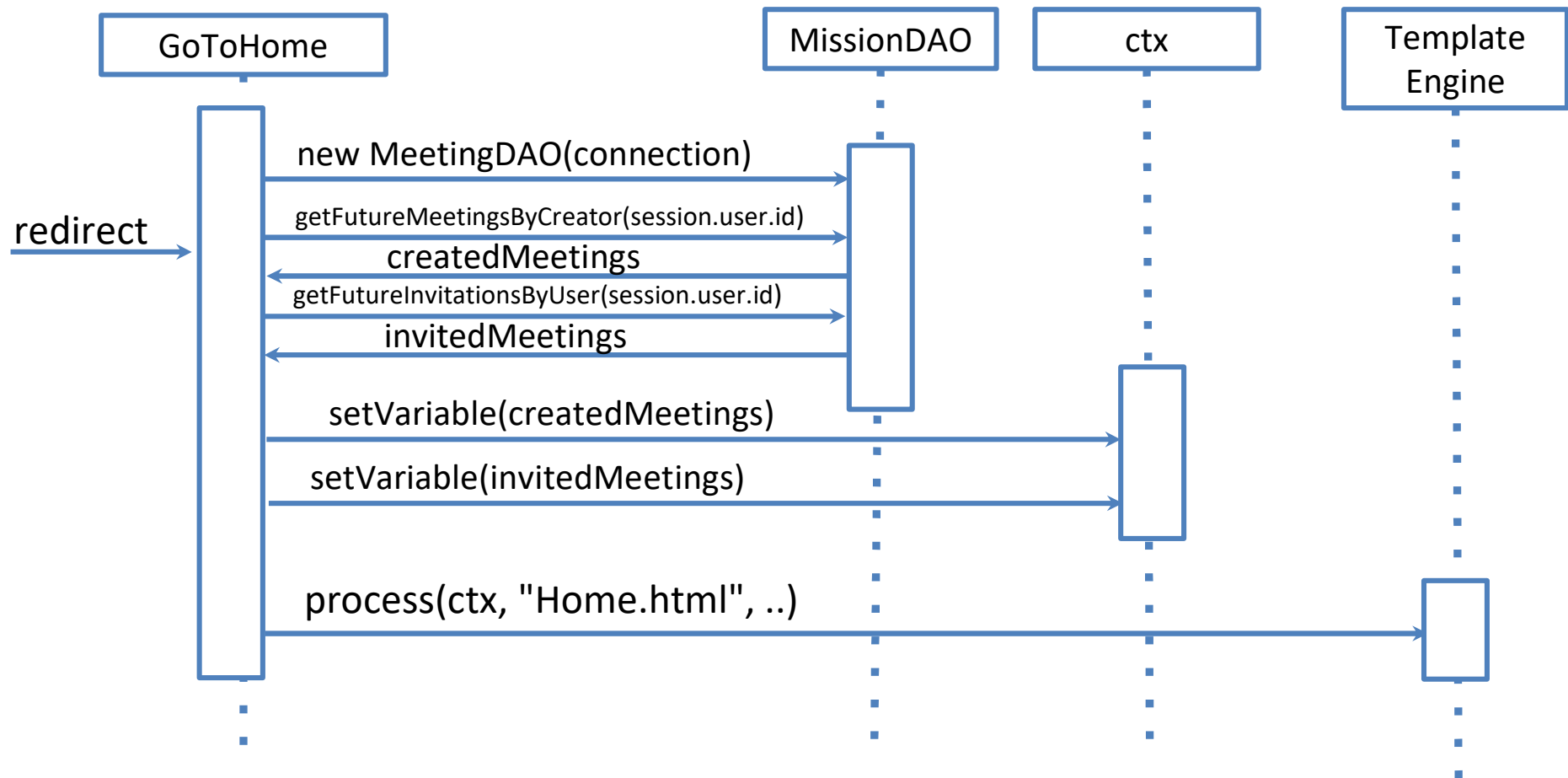
# Event: login



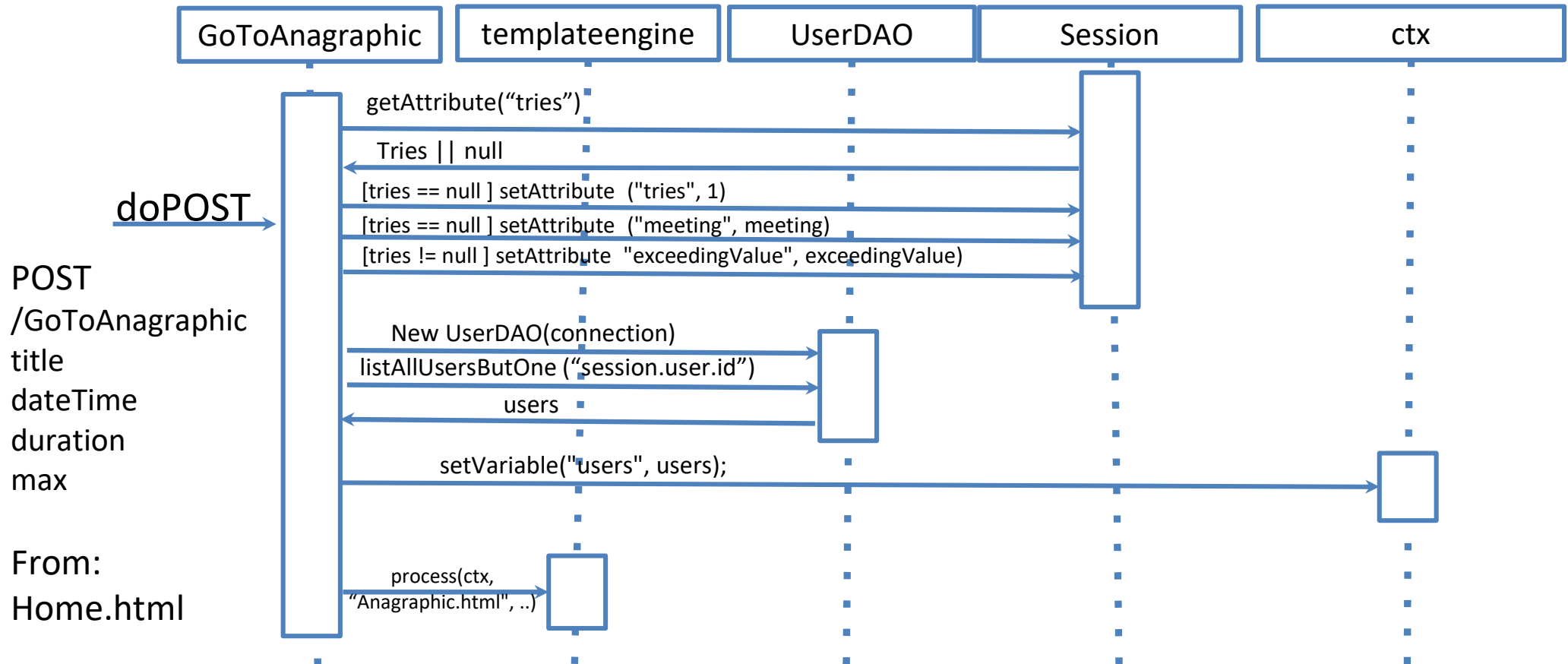
# Event: registration



# Event: go to home

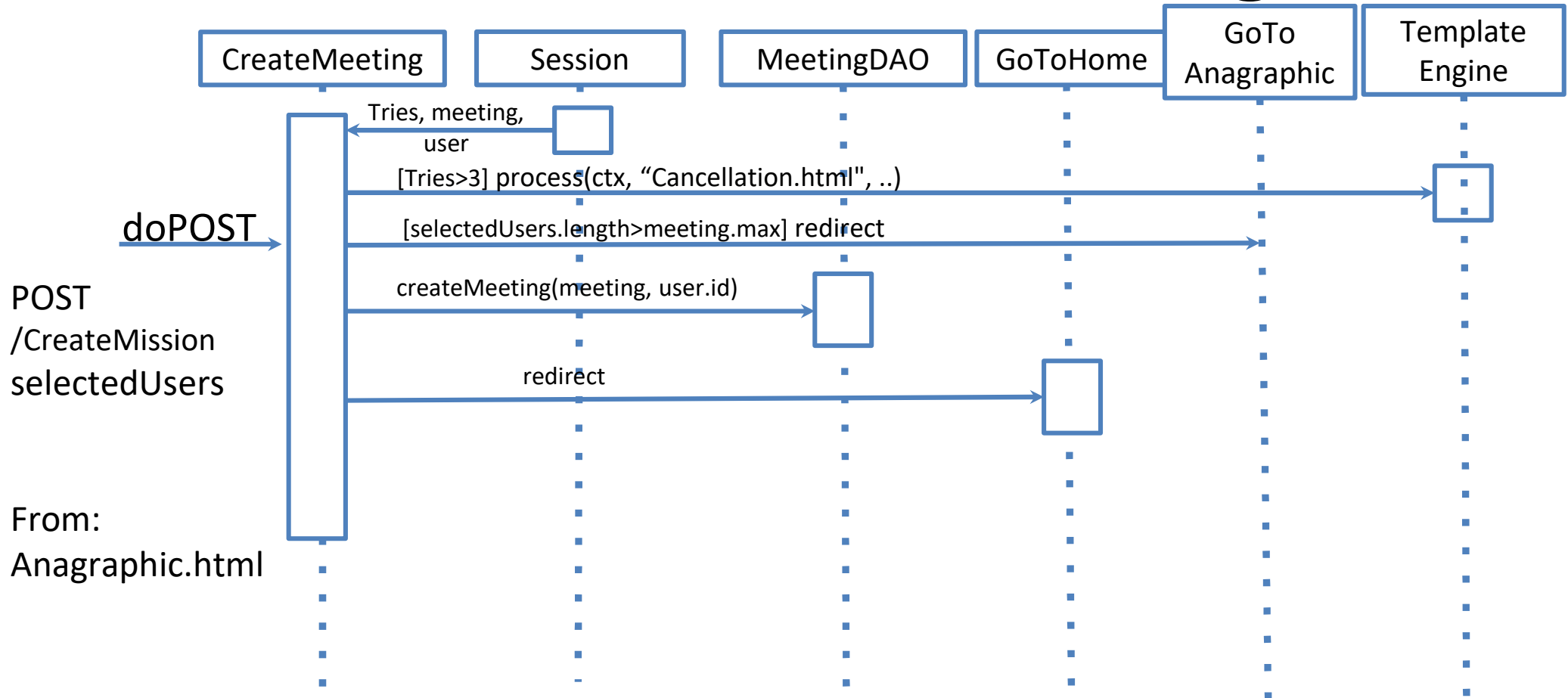


# Event: go to anagraphic





# Event: CreateMeeting



# Event: logout

