

Esercizio 5: RIUNIONI ONLINE

Tecnologie Informatiche per il Web - AA 2021/2022

Roberto Giandomenico
Francesca Grimaldi

PURE HTML version



Requirements

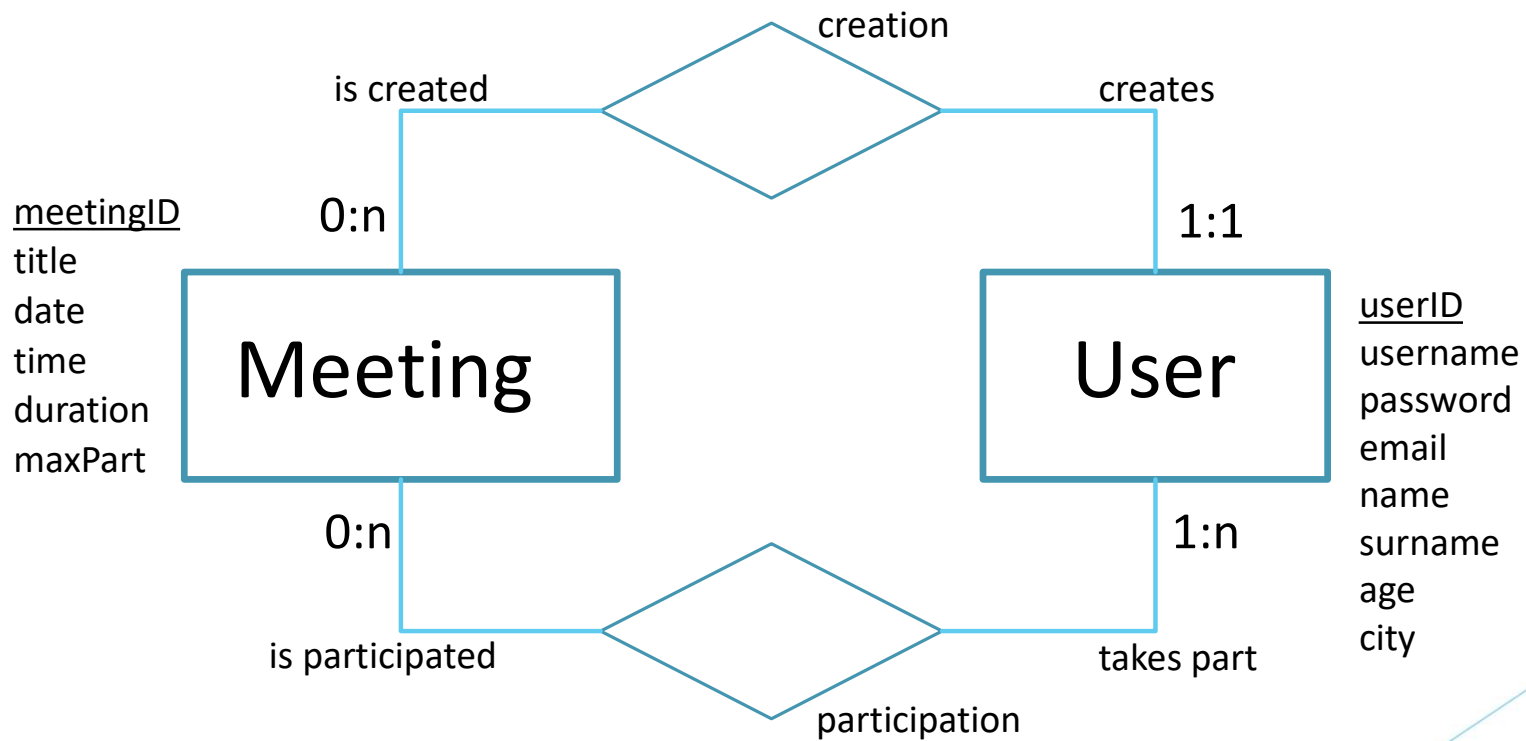
Un'applicazione web consente la gestione di riunioni online. L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password". La registrazione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

Data requirements analysis

Relationships, entities,
attributes

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



Local database schema (mySQL DDL 1/2)

```
CREATE TABLE `meeting` (  
  `meetingID` int NOT NULL  
  AUTO_INCREMENT,  
  `title` varchar(50) NOT NULL,  
  `date` date NOT NULL,  
  `time` time NOT NULL,  
  `duration` int NOT NULL,  
  `maxPart` int NOT NULL,  
  `creator` int NOT NULL,  
  PRIMARY KEY (`meetingID`),  
  UNIQUE KEY `meetingID_UNIQUE`  
  (`meetingID`),  
  KEY `creator_idx` (`creator`),  
  CONSTRAINT `creator` FOREIGN KEY  
  (`creator`) REFERENCES `user` (`userID`)  
  ON UPDATE CASCADE)
```

```
CREATE TABLE `user` (  
  `userID` int NOT NULL AUTO_INCREMENT,  
  `email` varchar(100) NOT NULL,  
  `username` varchar(50) NOT NULL,  
  `password` varchar(50) NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `surname` varchar(50) NOT NULL,  
  `age` int NOT NULL,  
  `city` varchar(50) NOT NULL,  
  PRIMARY KEY (`userID`),  
  UNIQUE KEY `email_UNIQUE` (`email`),  
  UNIQUE KEY `userID_UNIQUE` (`userID`),  
  UNIQUE KEY `username_UNIQUE` (`username`))
```

Local database schema (mySQL DDL 2/2)

We add a third table to represent the N:M relationship between user and meeting.

```
CREATE TABLE `participation` (  
    `meetingID` int NOT NULL,  
    `participantID` int NOT NULL,  
    PRIMARY KEY (`meetingID`,`participantID`),  
    KEY `participantID_idx` (`participantID`),  
    CONSTRAINT `participantID` FOREIGN KEY (`participantID`)  
REFERENCES `user` (`userID`) ON DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT `meetingID` FOREIGN KEY (`meetingID`) REFERENCES  
`meeting` (`meetingID`) ON DELETE CASCADE ON UPDATE CASCADE  
)
```

Application requirements analysis

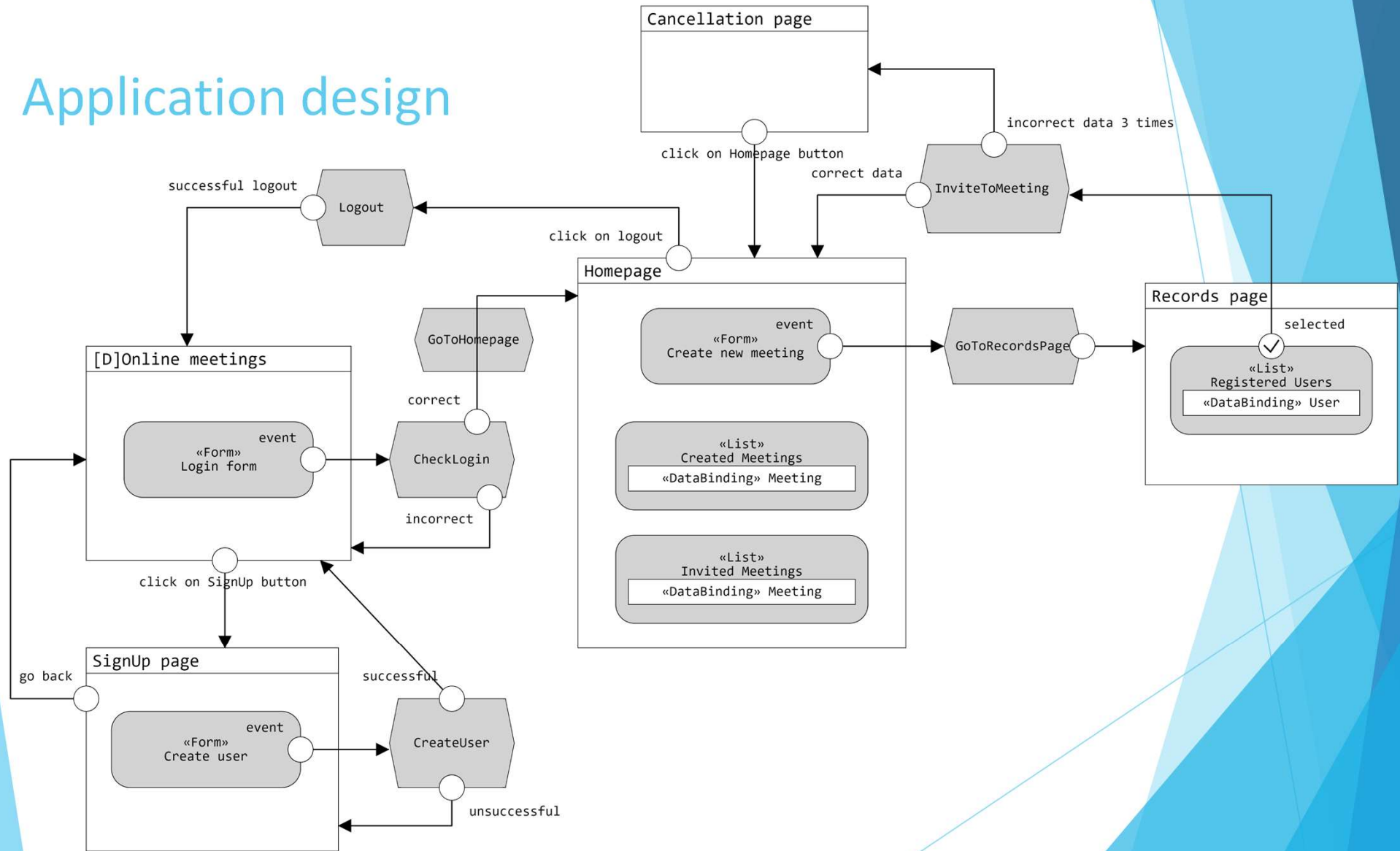
Pages (views), view
components, events, actions

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.

Completamento delle specifiche

- ▶ Si assume che durante la registrazione dell'utente vengano chiesti anche nome, cognome, città ed età.
- ▶ Si assume che tutti i campi delle form siano obbligatori.
- ▶ La **pagina di default** contiene la **form per il login** e un link alla **pagina che permette di registrarsi**.
- ▶ Nella **Homepage**, ossia nella pagina alla quale si accede dopo aver effettuato il login e che contiene l'elenco delle riunioni per quello specifico utente, è presente anche la **form per la creazione della riunione**.
- ▶ Si assume che l'utente possa creare una riunione scegliendo un numero massimo di partecipanti, che però non oltrepassi un massimo assoluto stabilito da noi.
- ▶ Inoltre, a seguito del login, si riconosce l'utente e un **messaggio di saluto** con il suo nome viene visualizzato.

Application design



Components

▶ Model objects (Beans)

- ▶ Meeting
- ▶ User
- ▶ MeetingForm (for support)
- ▶ UserForm (for support)

▶ Utilities

- ▶ DateChecker
- ▶ ConnectionHandler

▶ Data Access Objects (Classes)

▶ UserDao

getIDbyNick, getNickByID, createUser, checkCredentials, checkAvailability, getRegisteredUsers

▶ MeetingDAO

findCreatedMeetings, findInvitedMeetings, createMeeting, sendInvitation

▶ Controllers (servlets)

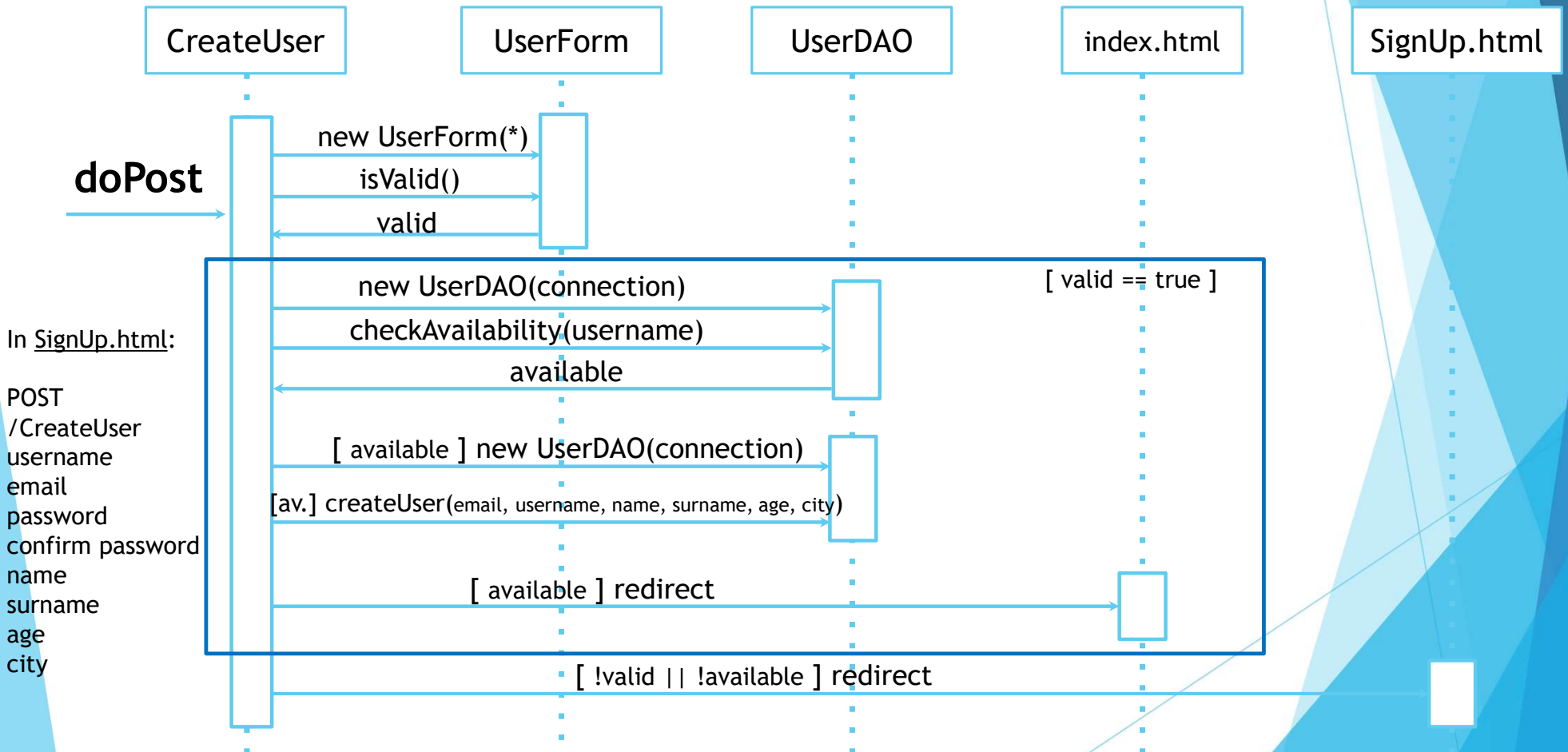
- ▶ CreateUser
- ▶ CheckLogin
- ▶ GoToHomepage
- ▶ GoToRecordsPage
- ▶ InviteToMeeting
- ▶ Logout

▶ Views (Templates)

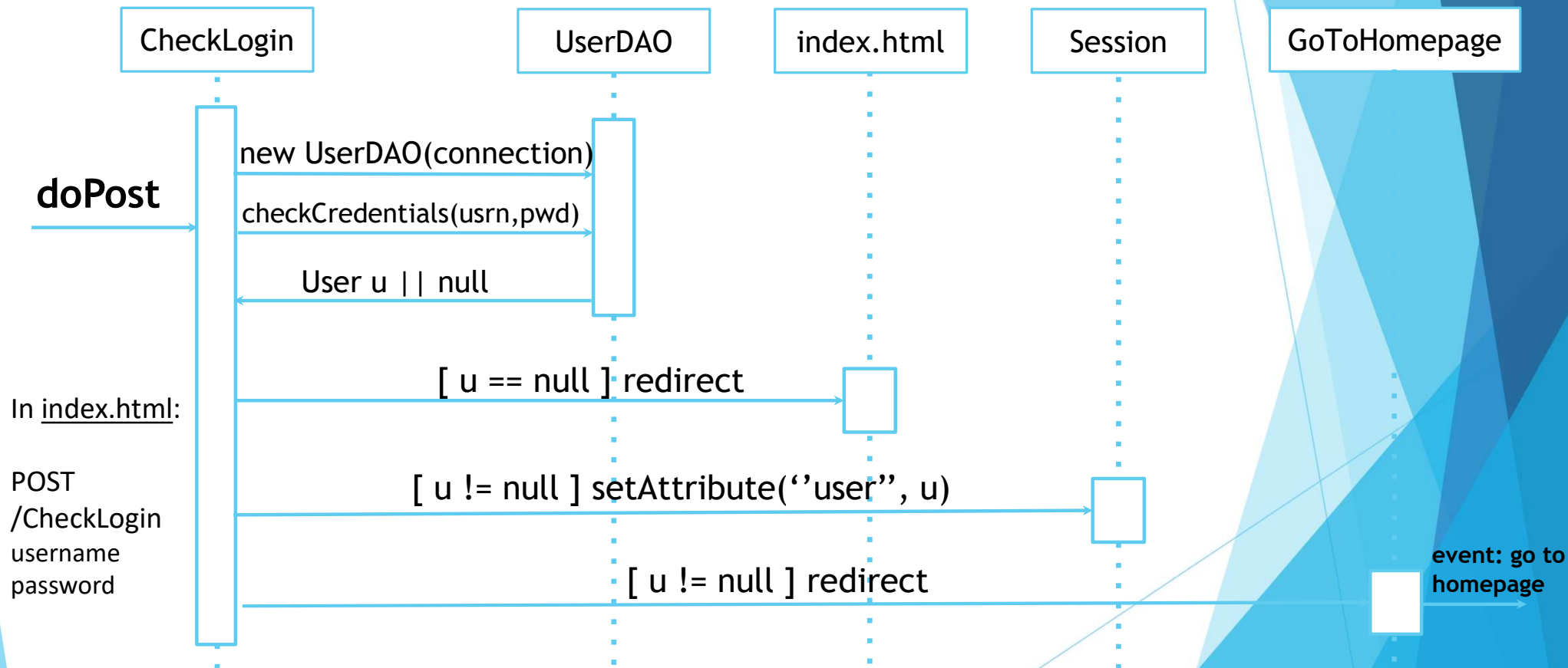
- ▶ index.html (*welcome page that contains the login form*)
- ▶ SignUp.html (*registration form*)
- ▶ Homepage.html (*user's homepage*)
- ▶ RecordsPage.html (*page from which select the users to invite to the meeting*)
- ▶ CancellationPage.html (*appears when too many attempts to create a meeting have been performed*)

* all the form fields are
set as parameters

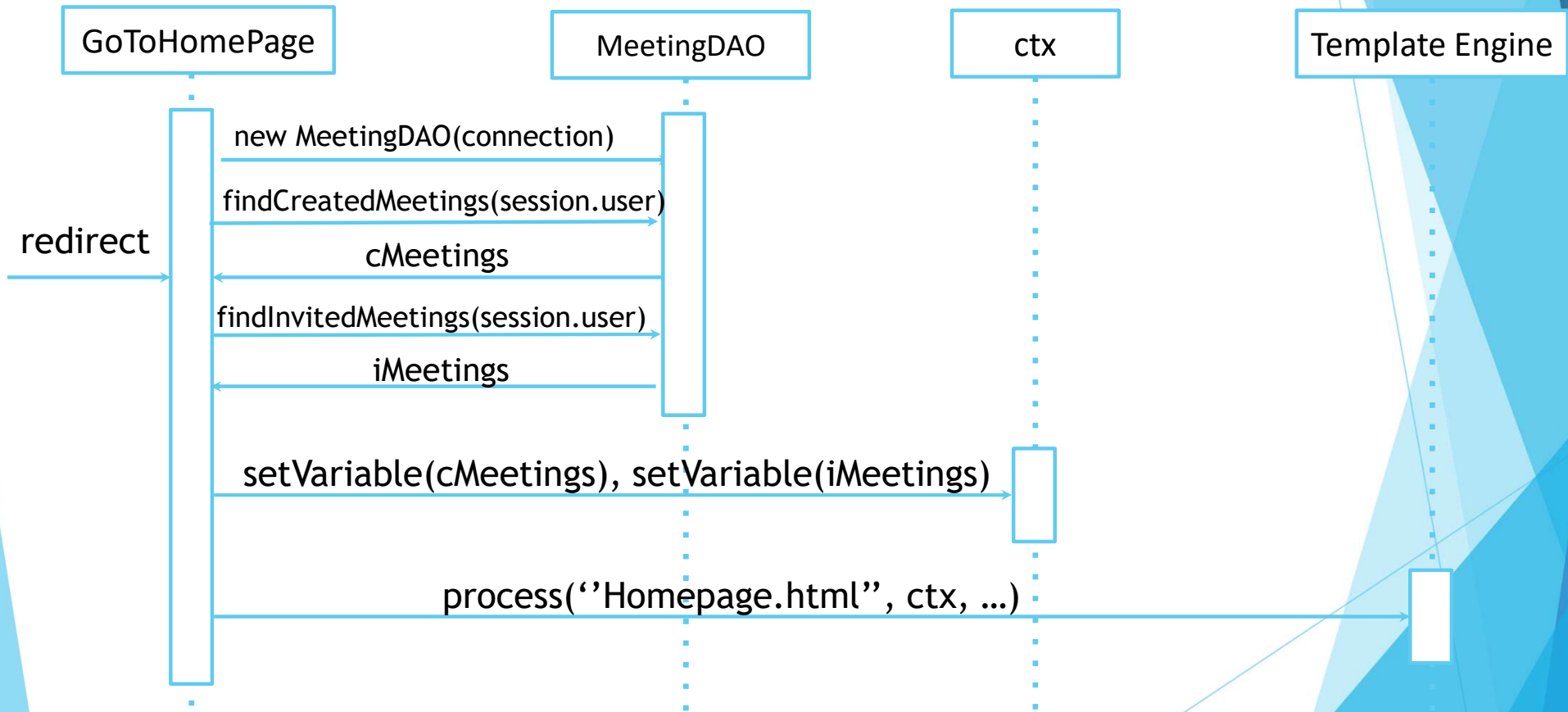
Event: register a user



Event: login



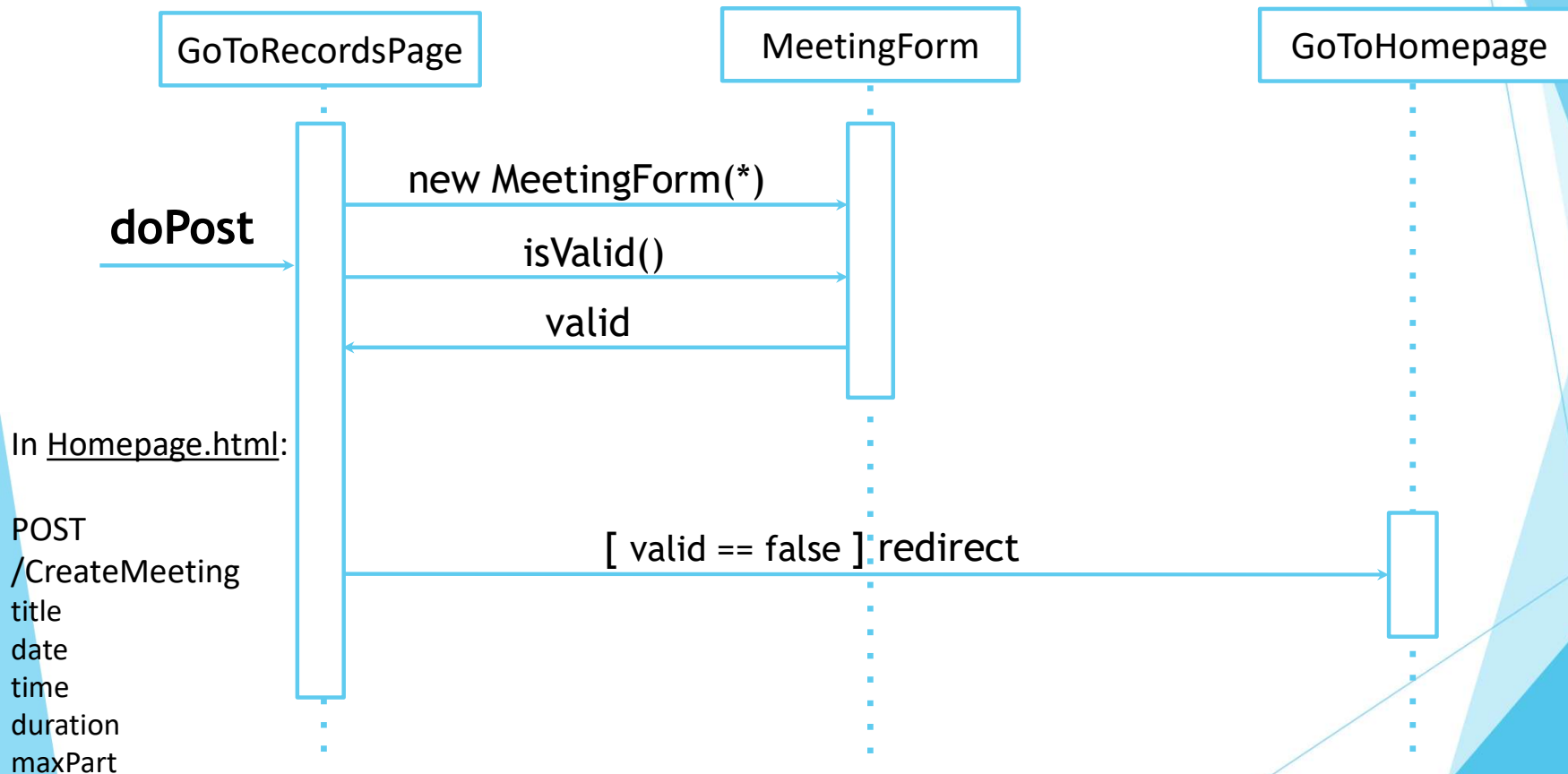
Event: go to homepage



Event: create a new meeting

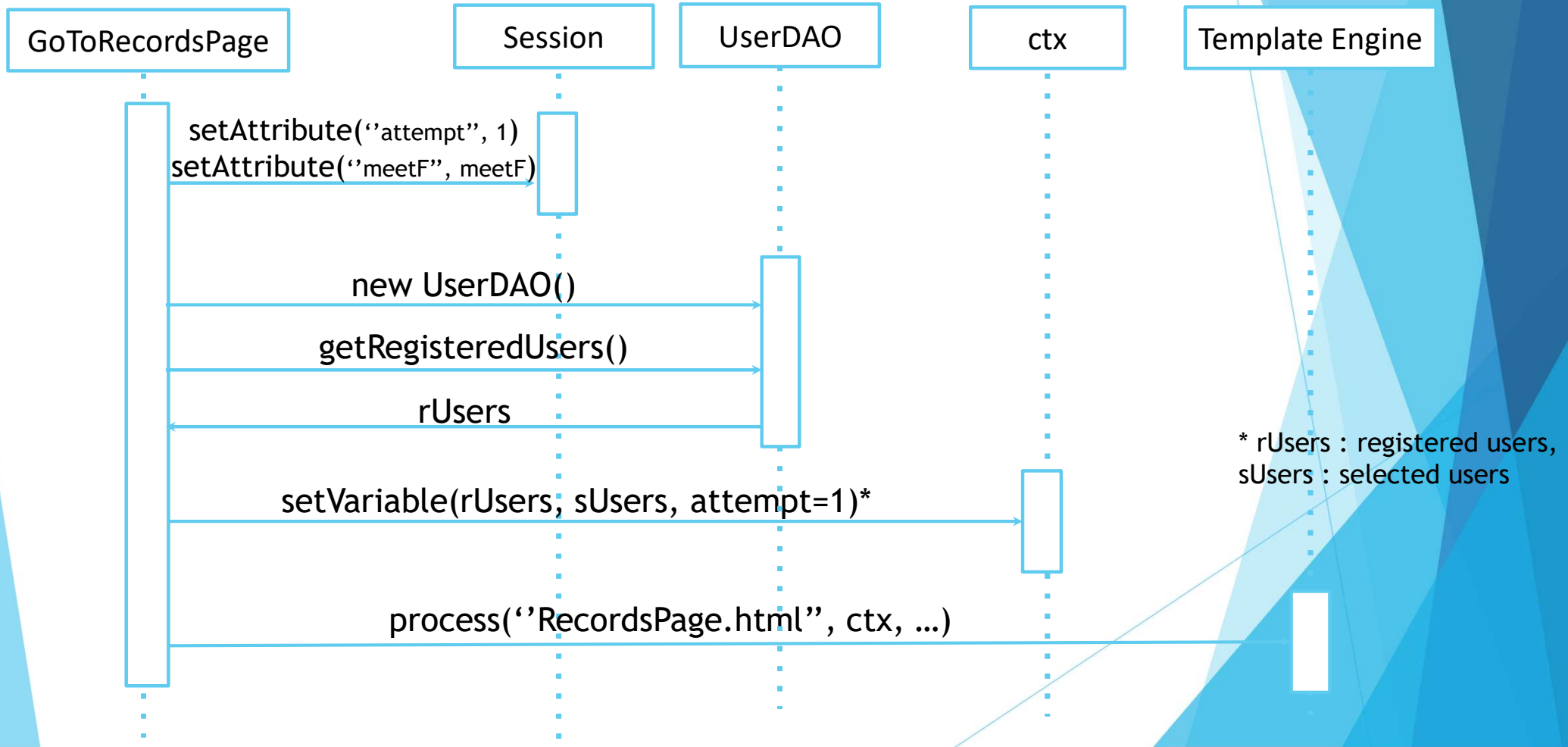
invalid input scenario

* all the form fields
are set as parameters



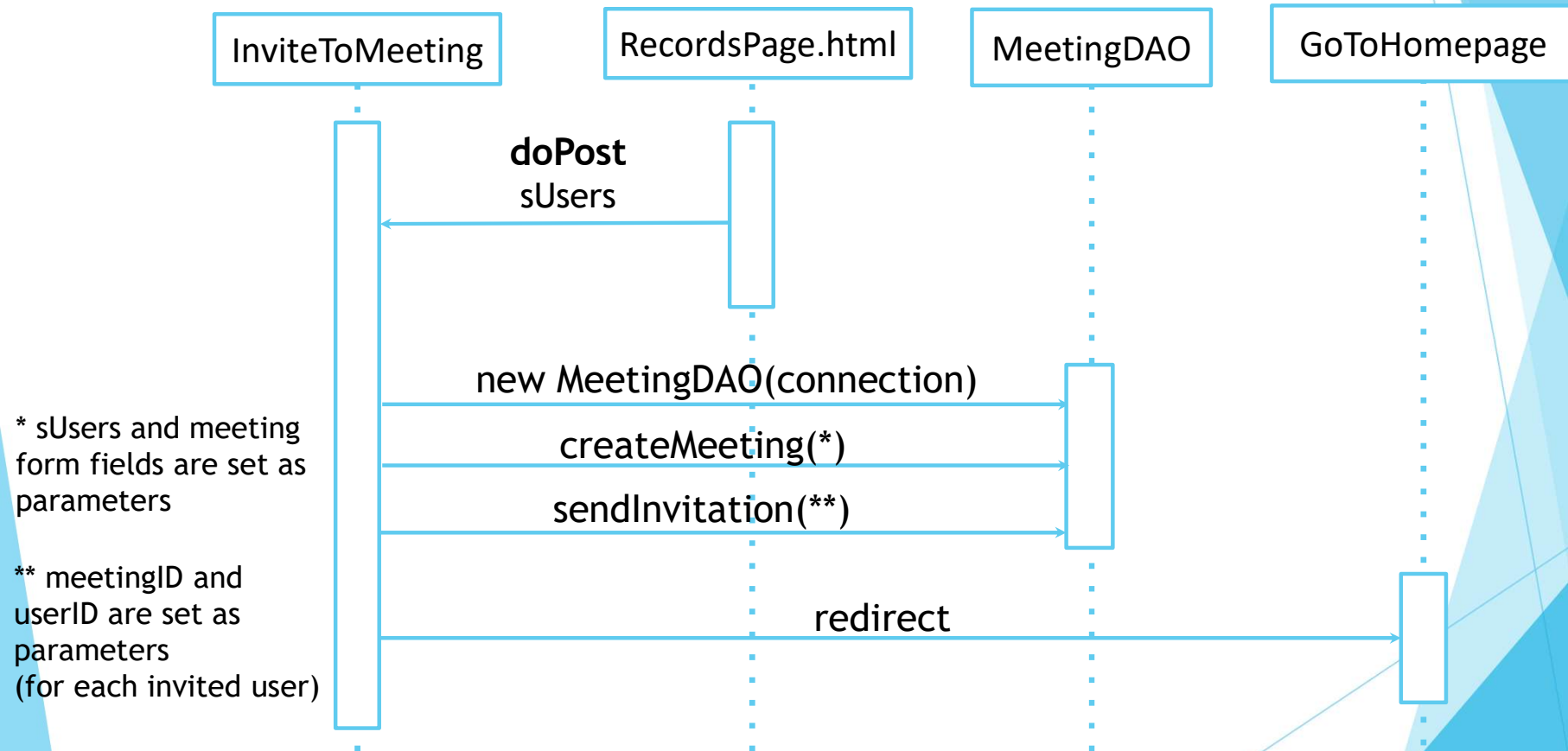
Event: create a new meeting

valid input scenario (getting information)



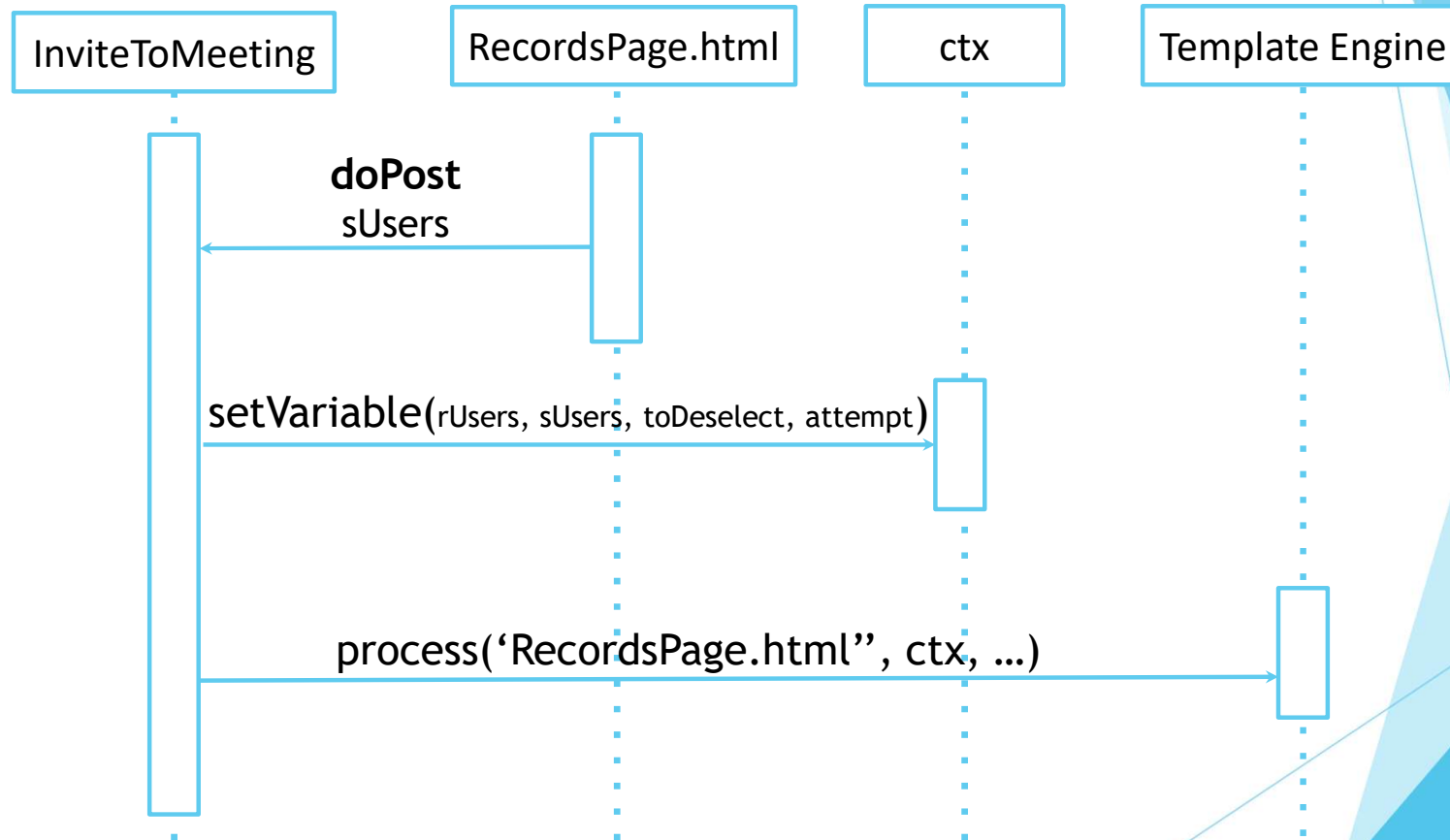
Event: create a new meeting

valid input scenario (selecting the correct number of people)



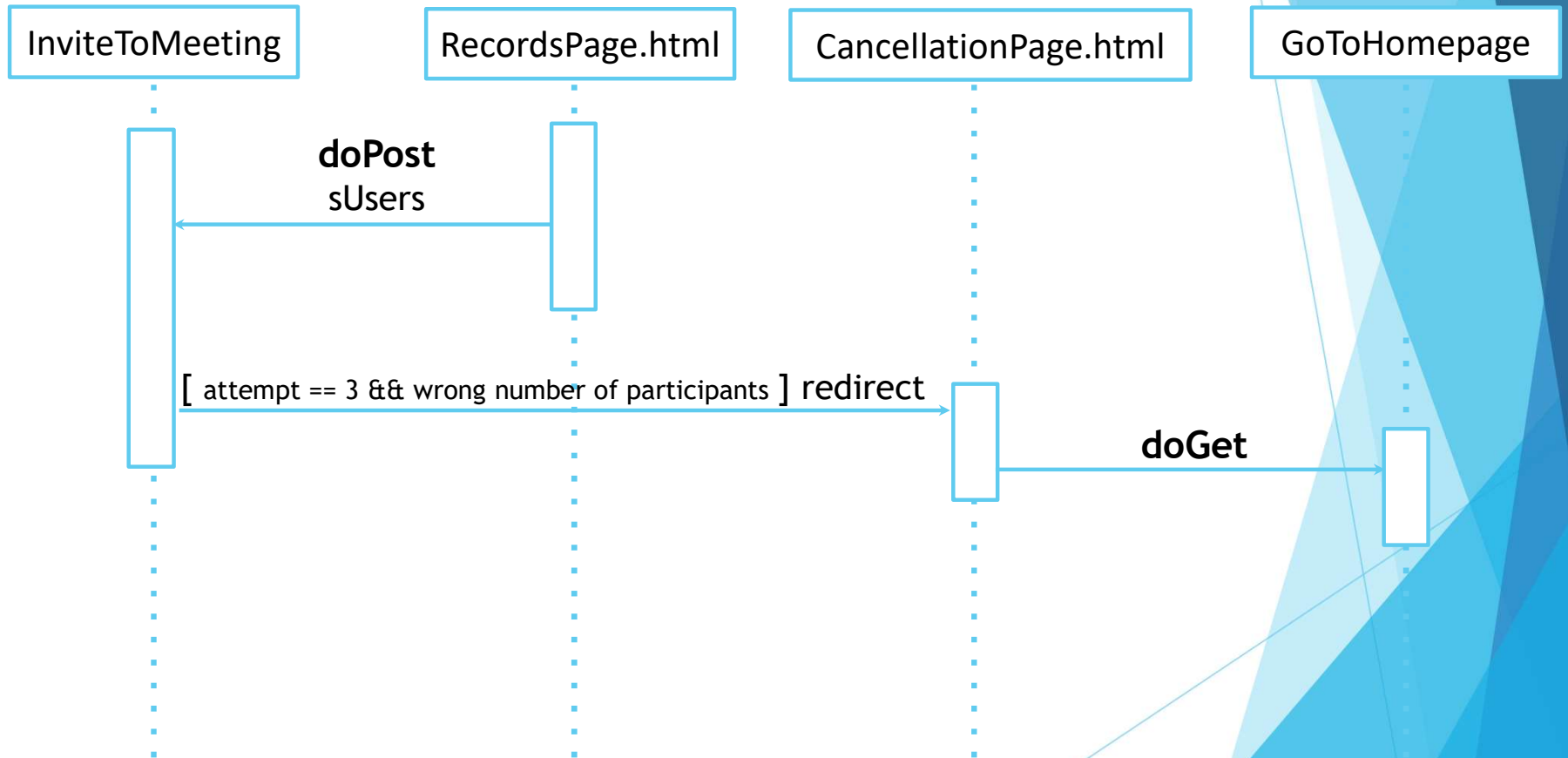
Event: create a new meeting

valid input scenario (but selecting the wrong number of people)

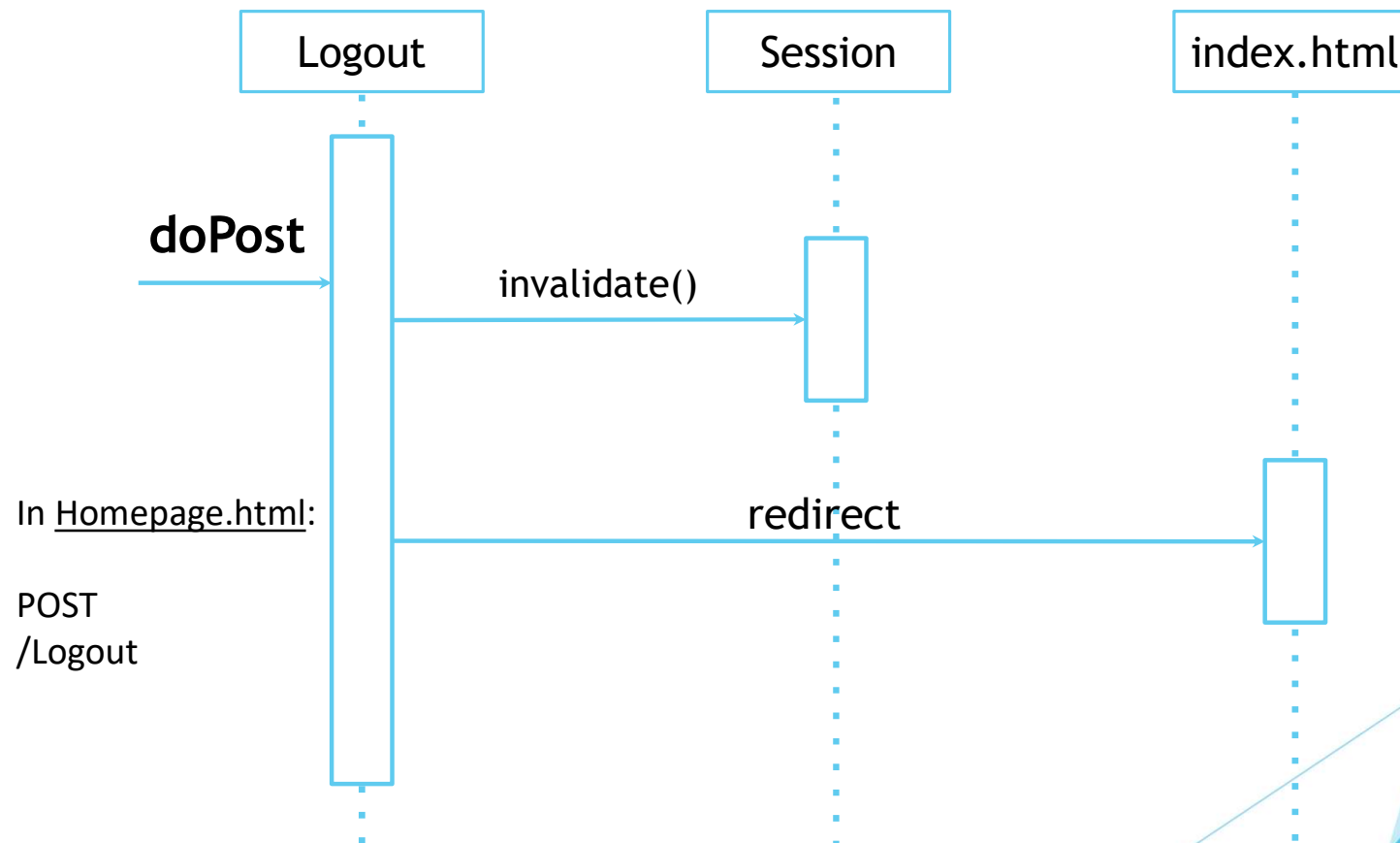


Event: create a new meeting

valid input scenario (but selecting the wrong number of people for the third time)



Event: logout



RIA version



JavaScript requirements

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

Pages (views), view
components, events

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.

Summary

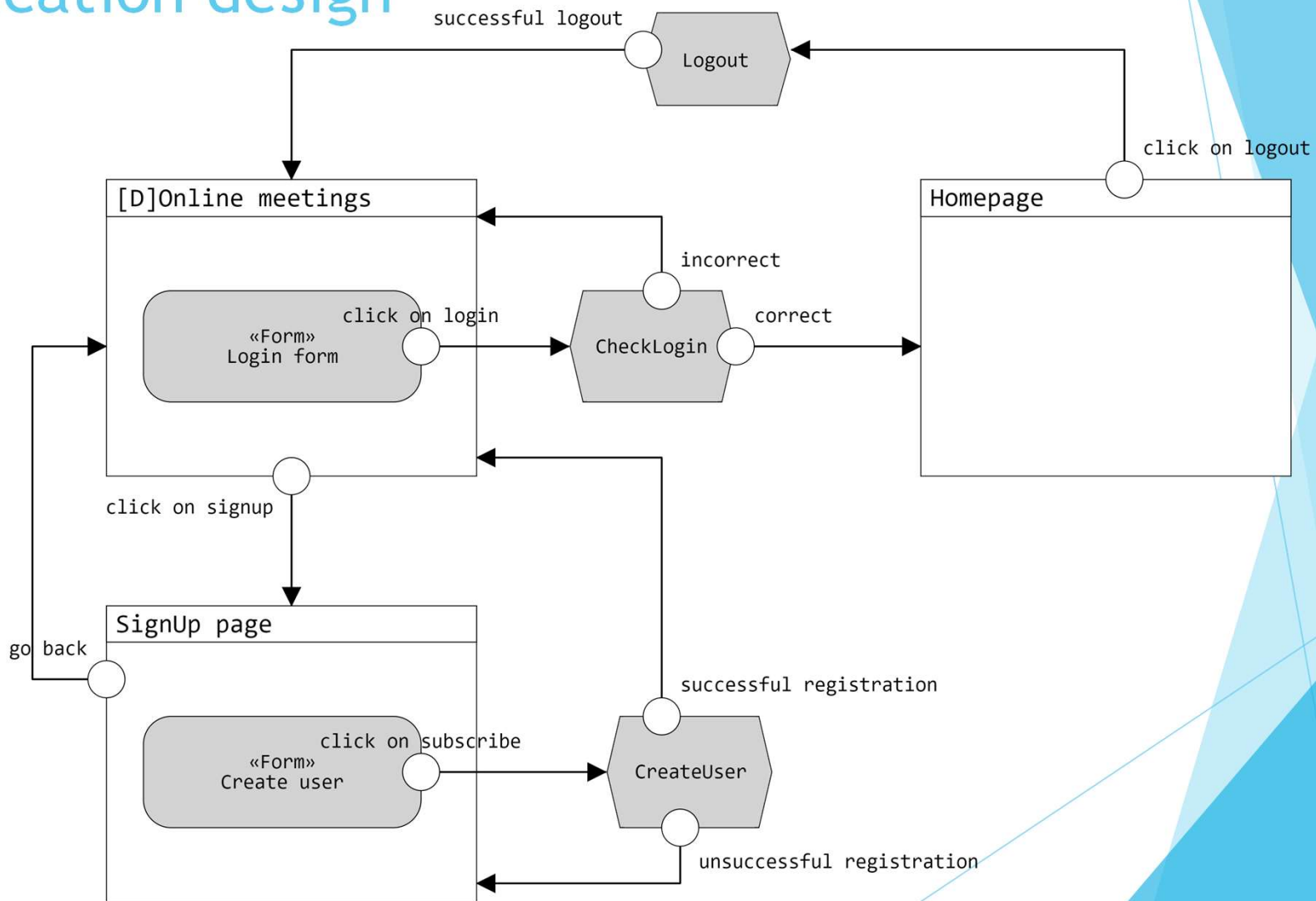
► Views and components

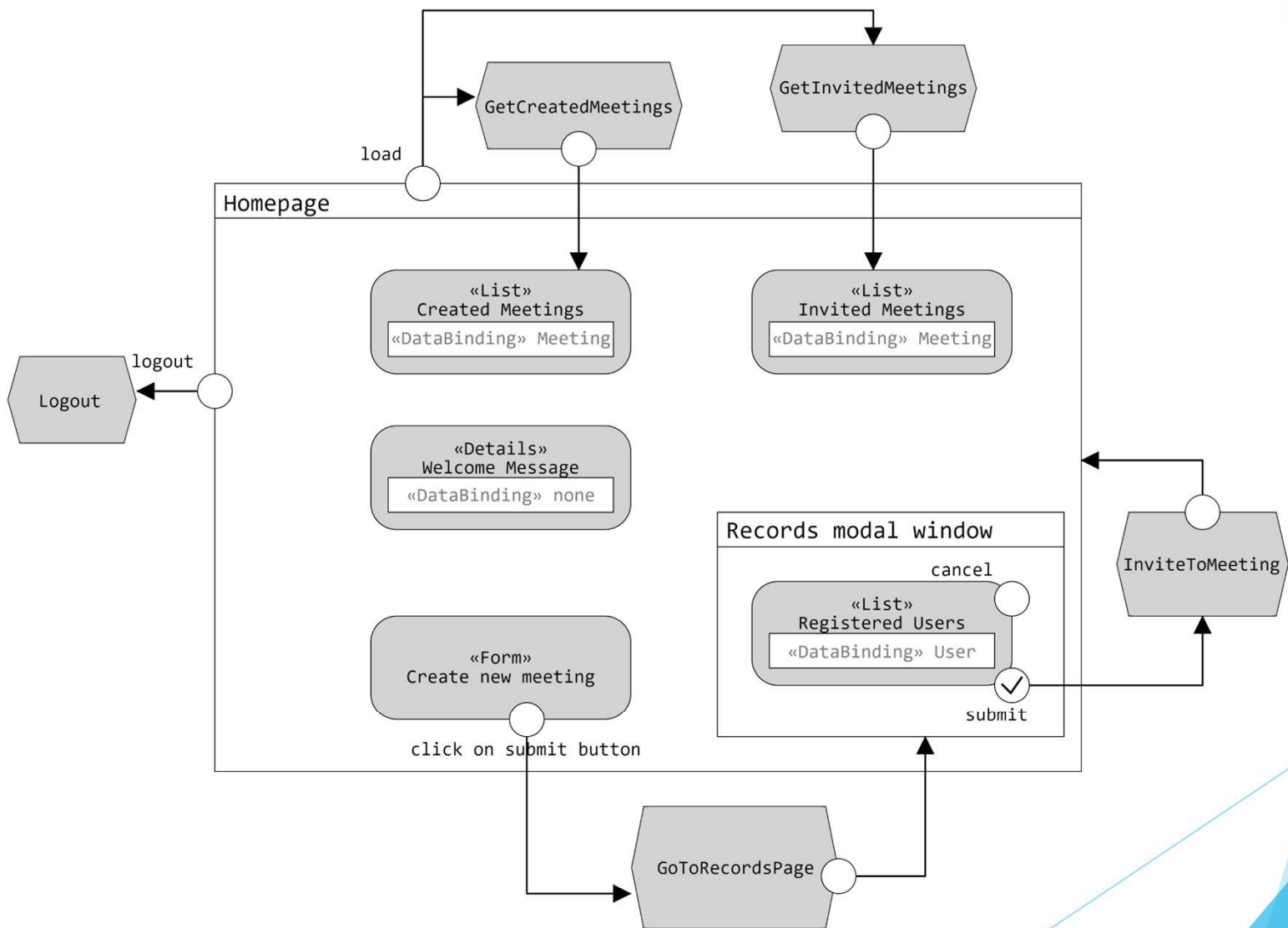
- Login
- SignUp
- Homepage
 - WelcomeMessage
 - CreatedMeetings
 - InvitedMeetings
 - CreateMeeting form
 - RecordsWindow (modal window to select users)

► Events and actions

- Login → *credentials check*
- Sign up → *check data and create user*
- Logout → *logout*
- Click on form 'submit' → *check data and open modal window*
- Click on 'cancel' from window → *empty form*
- Click on 'invite' from window → *check data and create meeting*

Application design





Server side: DAO & model objects

► Model objects (Beans)

- Meeting
- User
- MeetingForm (for support)
- UserForm (for support)

► Utilities

- DateChecker
- ConnectionHandler

► Data Access Objects (Classes)

- UserDao
- MeetingDAO

► Controllers (servlets)

- CreateUser
- CheckLogin
- GetCreatedMeetings
- GetInvitedMeetings
- GoToRecordsPage
- InviteToMeeting
- Logout

Client side: view & view component

▶ Login

Login form + submit and error handling

▶ Sign up

Signup form + submit and error handling

▶ Homepage

- PageOrchestrator

- start()
- refresh()

- WelcomeMessage

- show()

- InvitedMeetings

- reset()

- show()

- update()

- CreatedMeetings

- reset()

- show()

- update()

- RegisteredUsers

- reset()

- show()

- update()

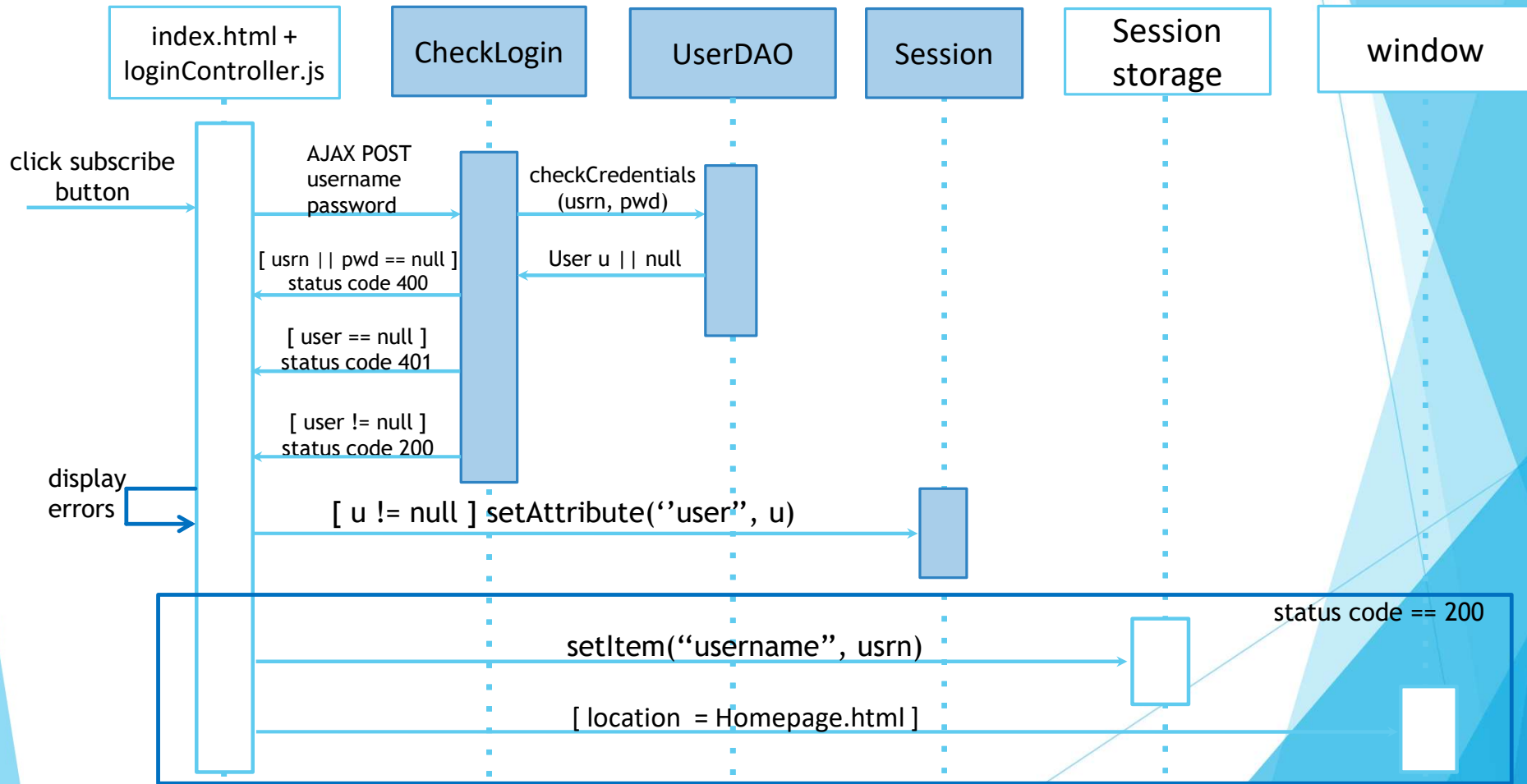
- Utilities

- E.g. setMinDate(), eventListeners for buttons...

Event: login

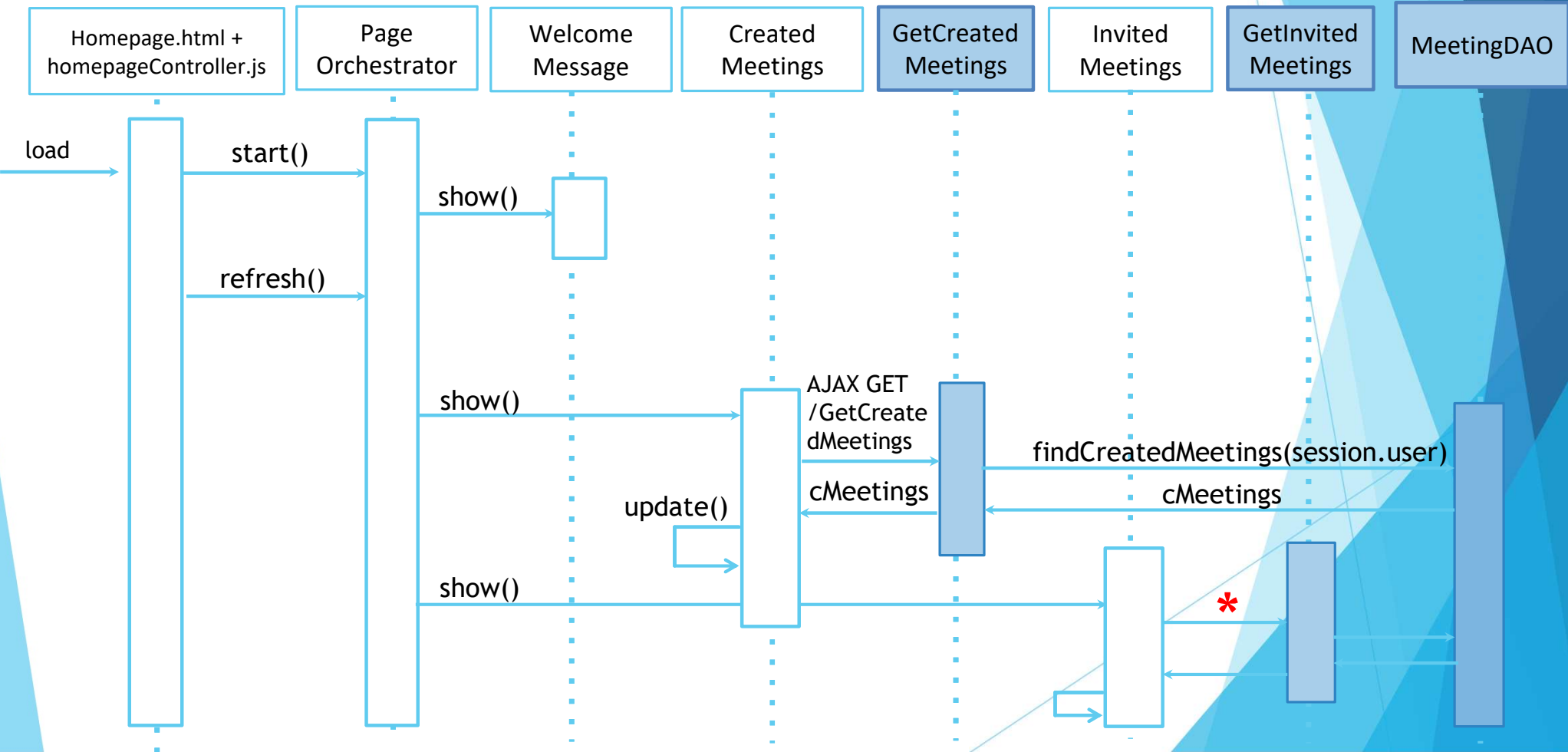
Client side

Server side

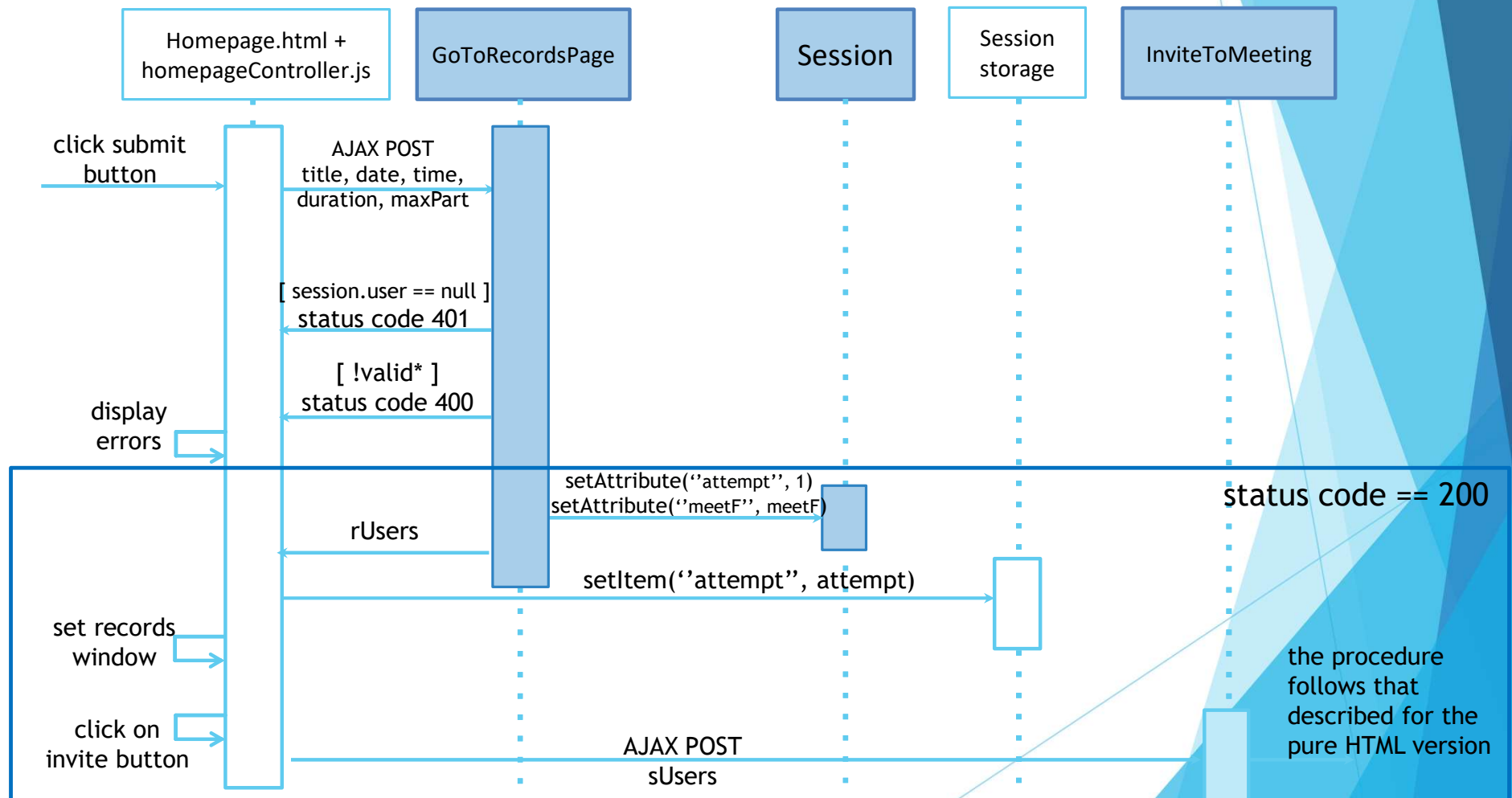


* messages to get invited meetings are analogous to the ones regarding created meetings

Event: access homepage



Event: create meeting



Event: logout

