

Esercizio 5: riunioni online

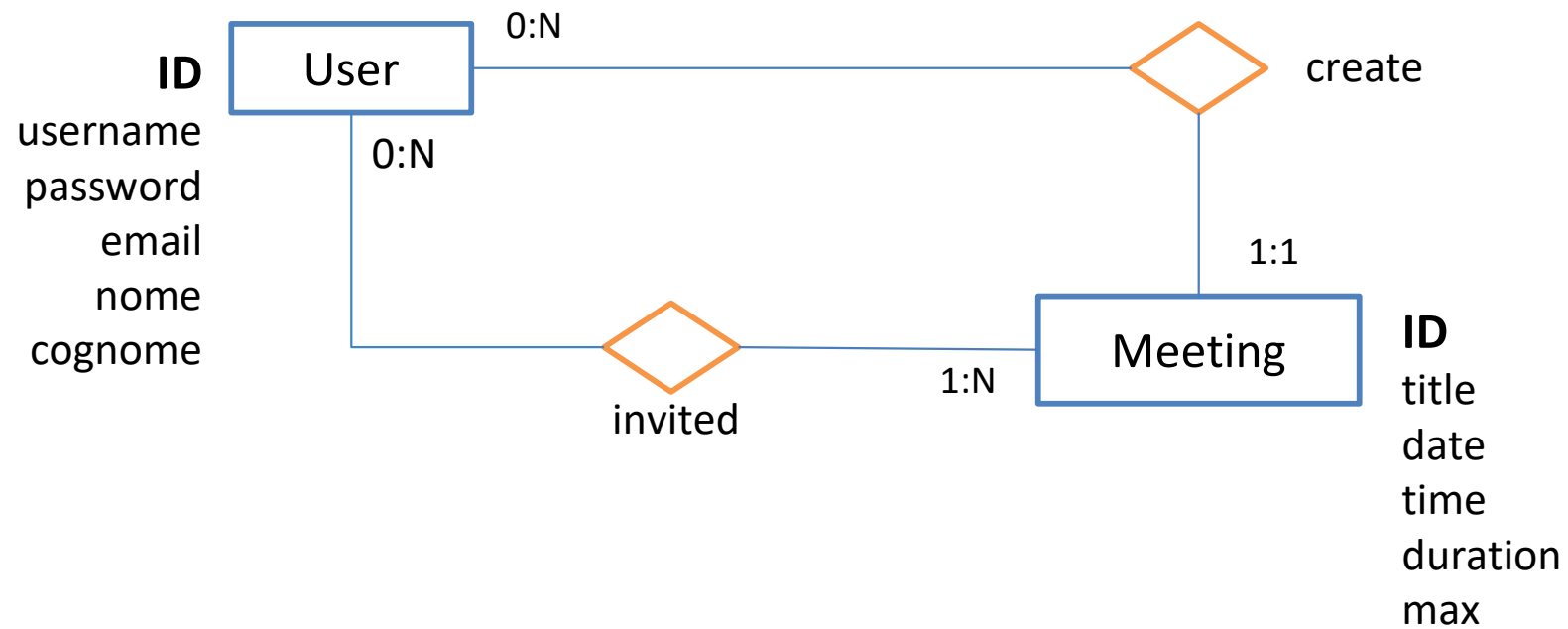
Matteo Gobbi Frattini

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", anche a lato client. 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**, dove l'intera applicazione è realizzata con un'unica pagina 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**. 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. Quando l'utente **inoltra la form** con il **bottone INVIA**, **appare l'elenco degli utenti registrati**. La scelta dall'anagrafica deve essere realizzata con una **pagina modale con i bottoni invia e cancella**. **L'utente può scegliere uno o più partecipanti dall'elenco e premere** il bottone INVIA per invitarli alla riunione. Se il numero d'invitati è superiore di X unità rispetto al massimo ammissibile, **appare 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. Al terzo tentativo scorretto di assegnare troppi invitati a una riunione **appare un messaggio "Tre tentativi di definire una riunione con troppi partecipanti, la riunione non sarà creata"**. In questo caso la riunione NON è memorizzata nella base di dati. I controlli di correttezza del numero massimo 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. 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

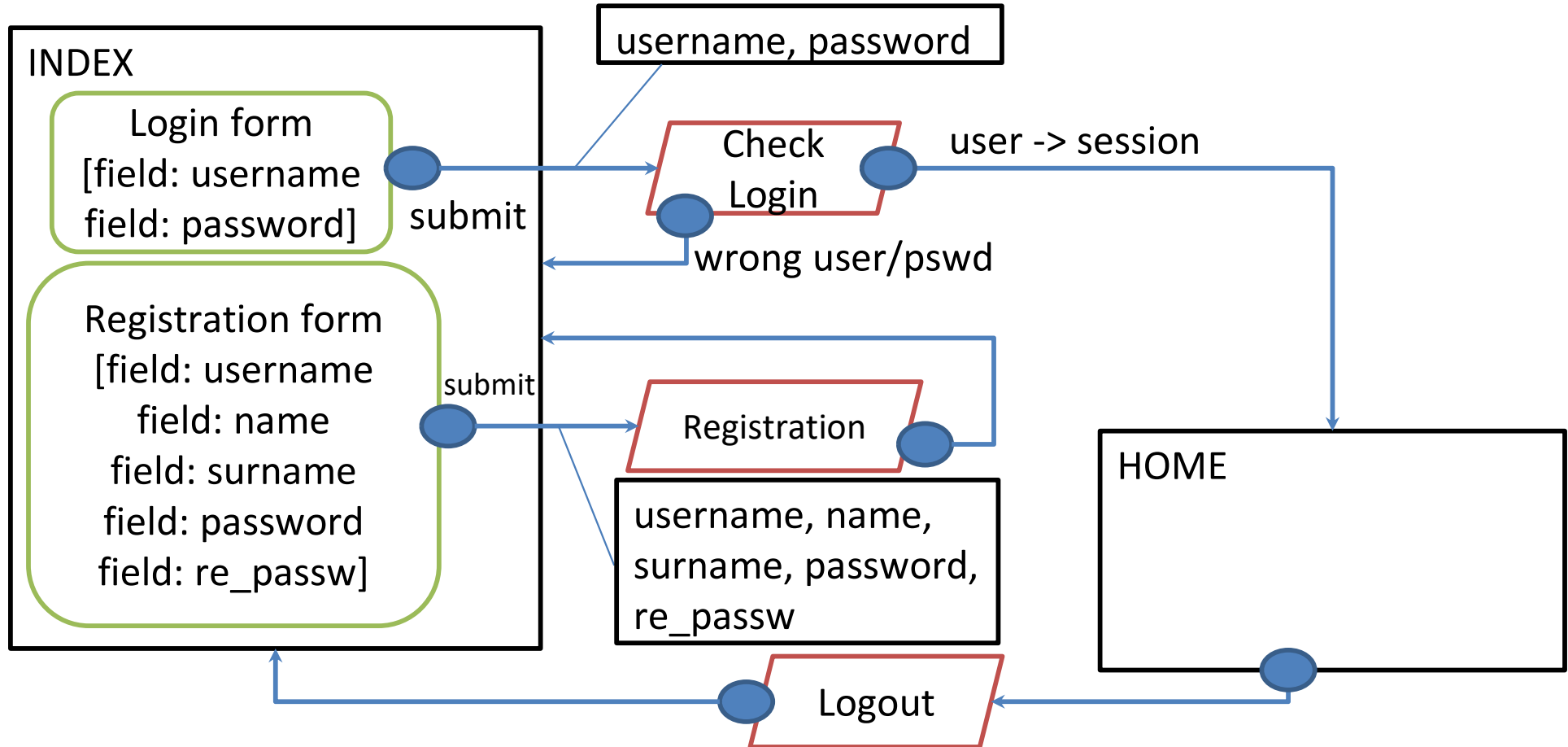
Completamento delle specifiche

- 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
- Le spese non possono avere saldi negativi.
- Non è possibile creare una riunione con inizio antecedente al momento attuale
- La finestra 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

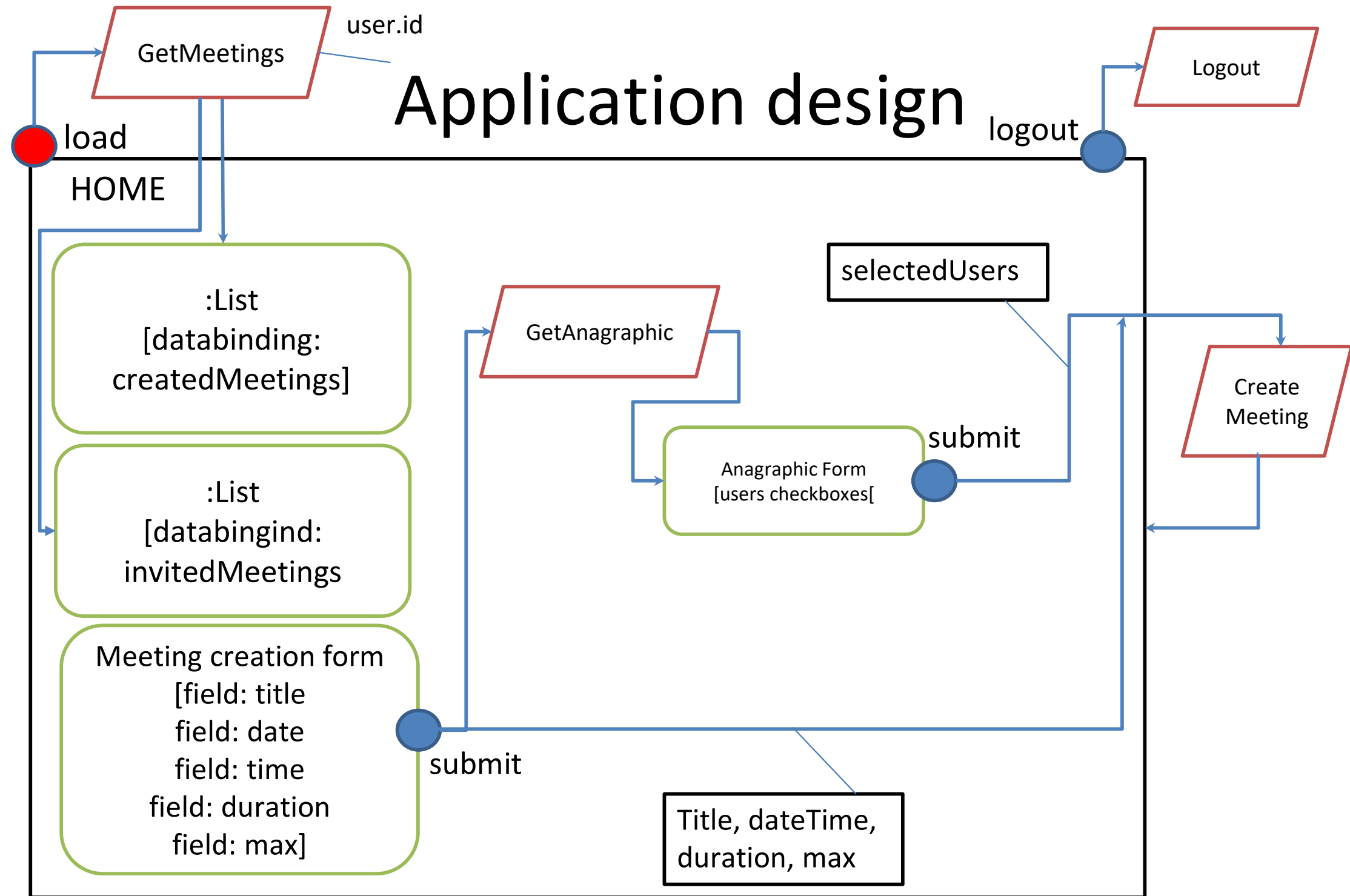
Sommario

- Viste e componenti
 - Pagina index
 - Login form
 - Registration form
 - Pagina Home
 - Elenco riunioni create e «invitate»
 - Form creazione riunione
 - Finestra modale anagrafica con bottoni «invita» e «cancella»
 - Bottone «Crea riunione»
 - Bottone logout
- Eventi e azioni
 - Login
 - Verifica credenziali
 - Registrazione
 - Creazione nuovo utente
 - Click su bottone Crea Riunione
 - Apertura finestra anagrafica
 - Click su bottone Cancella
 - Chiusura finestra anagrafica
 - Click su bottone Invita
 - Creazione nuova riunione
 - Logout
 - logout

Application design



Application design



Eventi & azioni

Client side		Server side	
Evento	Azione	Evento	Azione
index → login form → submit	Controllo dati	POST username password	Controllo credenziali
Index → registration form → submit	Controllo dati, svuotamento form	POST username, name, surname, email, password	Registrazione utente
Home → load	Aggiorna view con prossime riunioni	GET (nessun parametro)	Estrazione riunioni future utente
Home → create meeting form → submit	Aggiorna finestra modale anagrafica con tutti gli utenti e la fa comparire	GET (nessun parametro)	Estrazione utenti
Home → anagraphic → submit	Controllo dati, svuotamento createMeetingForm	POST (title, dateTime, duration, max, selectedUsers)	Creazione nuova riunione
Home → anagraphic → cancel	Scomparsa finestra modale anagrafica, reset stato (tentativi)	-	-
Home → Logout	Redirect index	GET (nessun parametro)	Terminazione sessione

Controller / event handler

Client side		Server side	
Evento	Controllore	Evento	Controllore
index → login form → submit	Function makeCall	POST (username, password)	CheckLogin (servlet)
Index → registration form → submit	Function makeCall	POST (username, name, surname, email, password)	Registration (servlet)
Home → load	Function checkLoggedInUser Function fillTables	GET (nessun parametro)	GetMeetings (servlet)
Home → create meeting form → submit	Function anagraphicHandler	GET (nessun parametro)	GetAnagraphic (servlet)
Home → anagraphic → submit	Function inviteButtonFunction	POST (title, dateTime, duration, max, selectedUsers)	CreateMeeting (servlet)
Home → anagraphic → cancel	Function cancelButtonFunction	-	-
Home → logout	Function logout	GET (nessun parametro)	Logout (servlet)

Server side: DAO & model objects

- Controllers
 - CheckLogin
 - Registration
 - GetMeetings
 - GetAnagraphic
 - CreateMeeting
 - Logout
- 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)
 - getFutureMeetingsByUser(userId)
 - InvitationDAO
 - addInvitation(userId, meetingId)

Client side: views & functions

Il valore dei seguenti elementi viene calcolato lato client:

- Index
 - Login form
 - loginErrorMessage
 - Registration form
 - registrationMessage
- Home
 - welcomeMessage
 - tableErrorMessage
 - Created meetings table
 - createdMeetingsTableBody
 - noCreatedMeetingsMessage
 - Invited meetings table
 - invitedMeetingsTableBody
 - noInvitedMeetingsMessage
 - Meeting form
 - meetingFormMessage
 - Anagraphic modal
 - maxInvitationsMessage
 - anagraphicErrorMessage
 - triesMessage
 - anagraphicServerErrorMessage
 - anagraphicTableBody

In particolare, vengono calcolati all'interno delle seguenti funzioni javascript:

- Index
 - loginErrorMessage: makecall(Checklogin) callback function
 - registrationMessage: makeCall(Registration) callback function
- Home
 - welcomeMessage: checkLoggedUser()
 - Meetings tables and messages: fillTables()
 - Meeting form message: makeCall(CreateMeeting) callback function
 - Anagraphic modal
 - anagraphicTableBody: showAnagraphic()
 - Everything else: anagraphicHandler()

Event Listeners

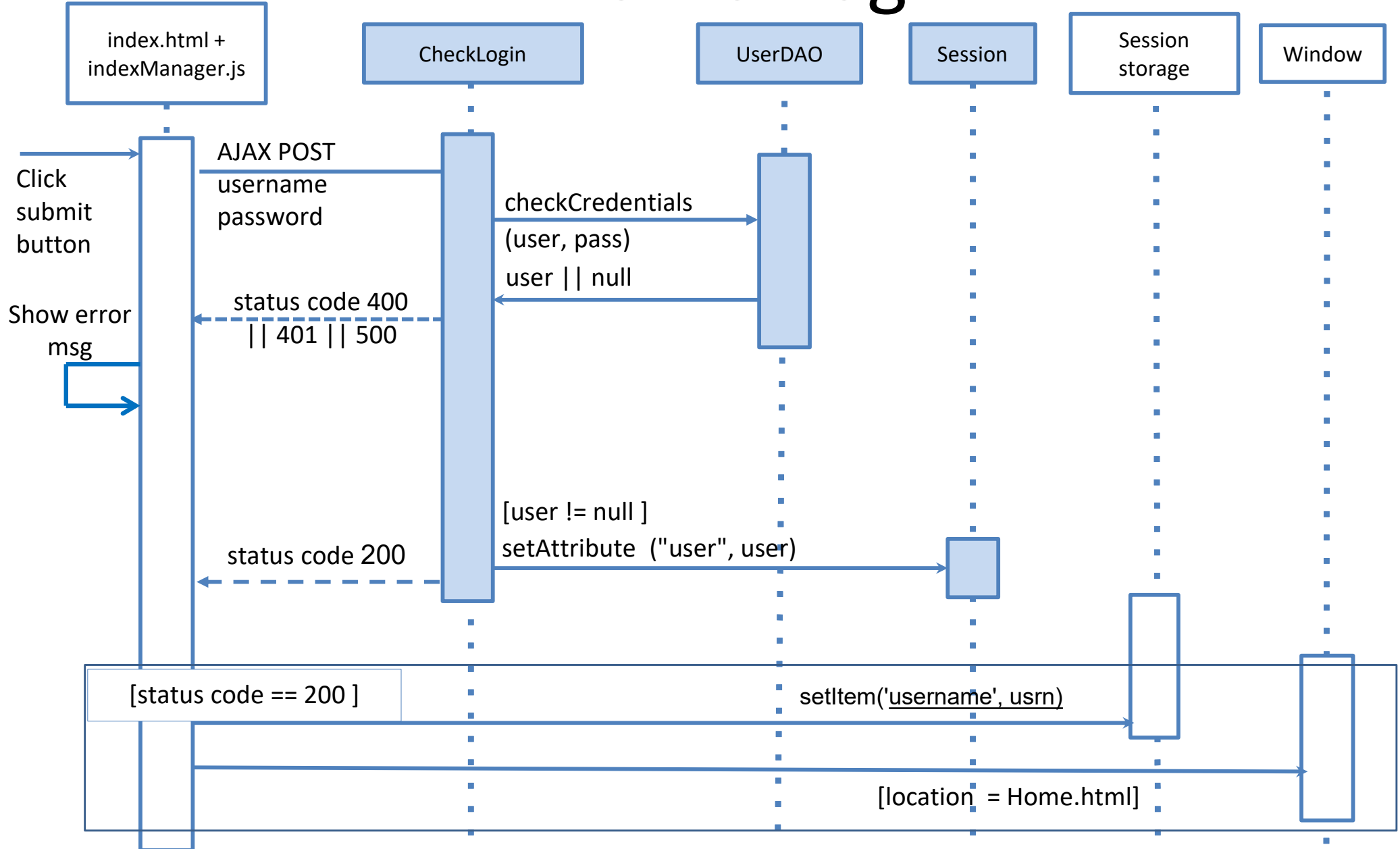
Elenco degli eventListener e delle funzioni che evocano:

- Index
 - loginButton("click") -> makeCall(CheckCredentials)
 - registrationButton("click") -> makeCall(Registration)
- Home
 - window("load") -> checkLoggedUser, fillTables
 - meetingFormButton("click") -> anagraphicHandler
 - inviteButton("click") -> inviteButtonFunction
 - cancelButton("click") -> cancelButtonFunction
 - logoutButton("click") -> logout

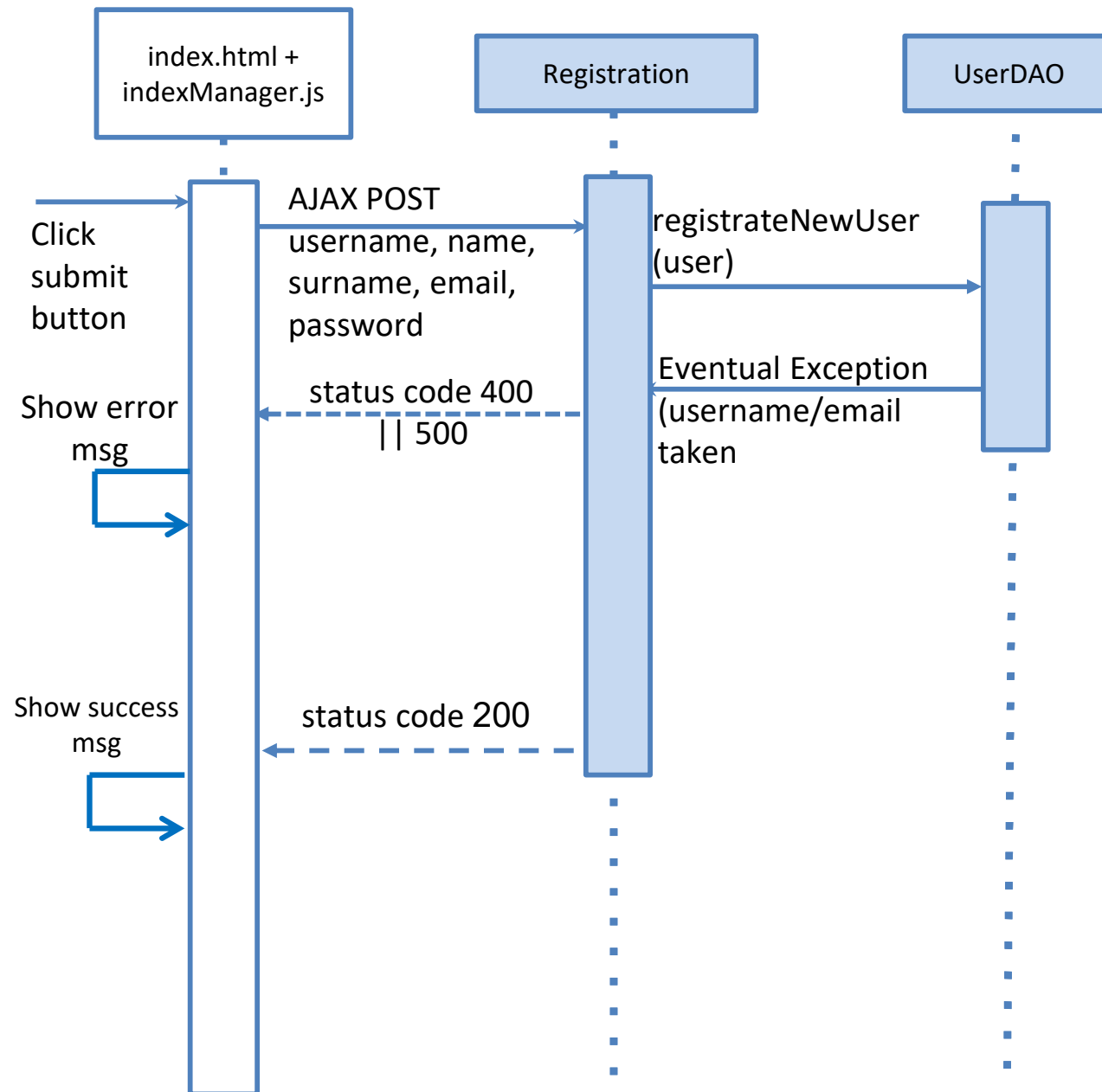
Client side

Server side

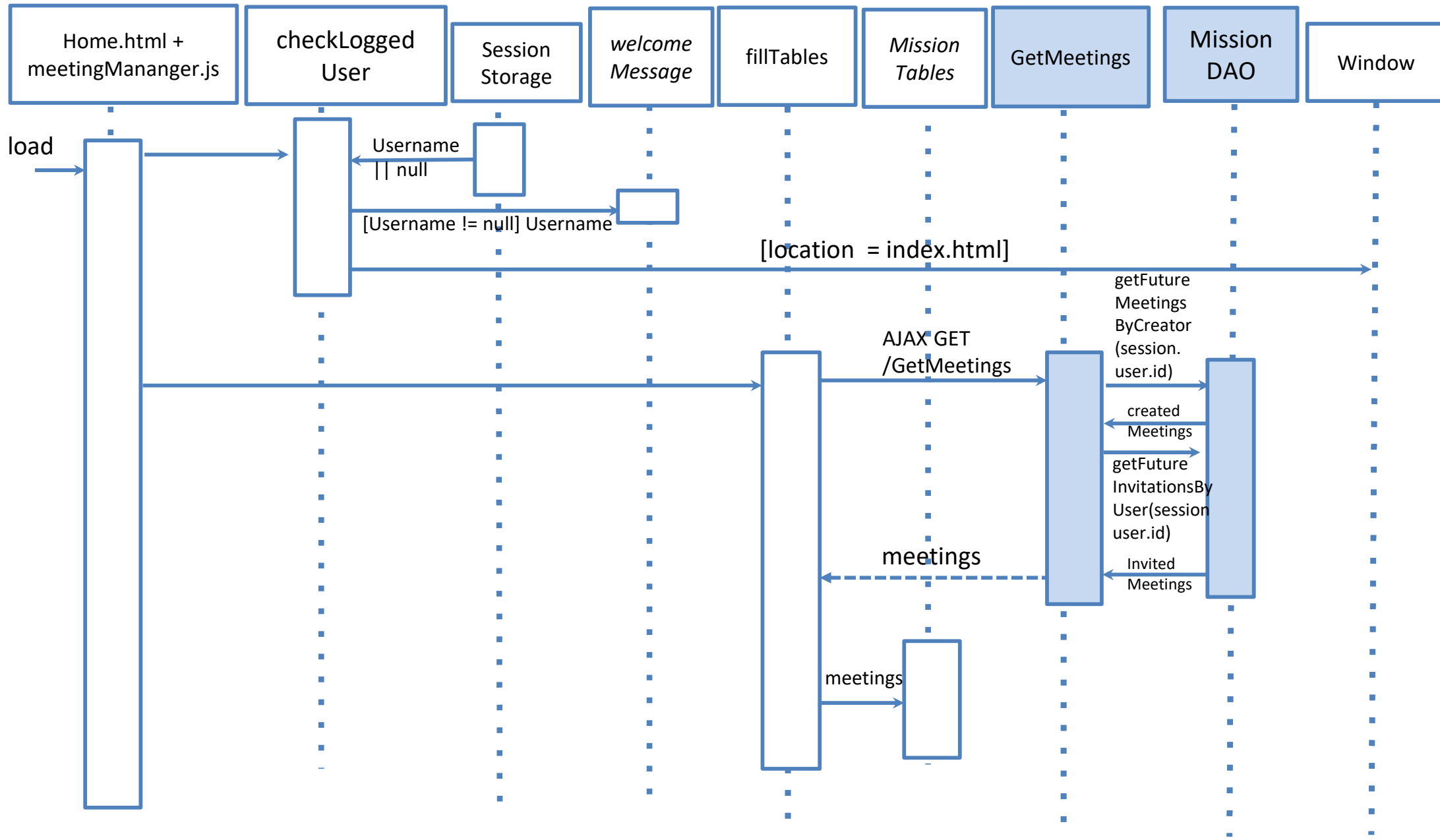
Evento: login



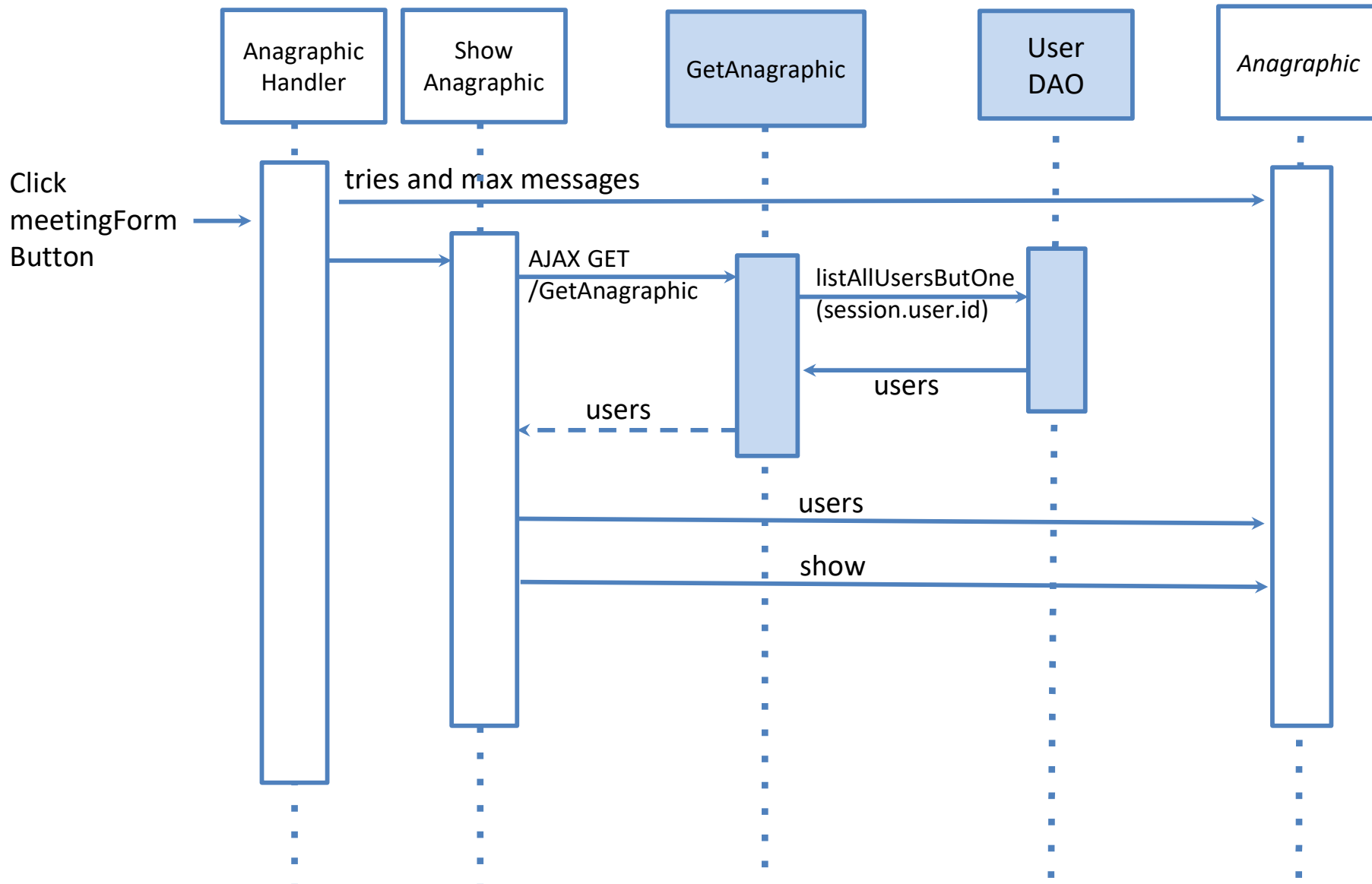
Evento: Registrazione



Evento: caricamento Home page



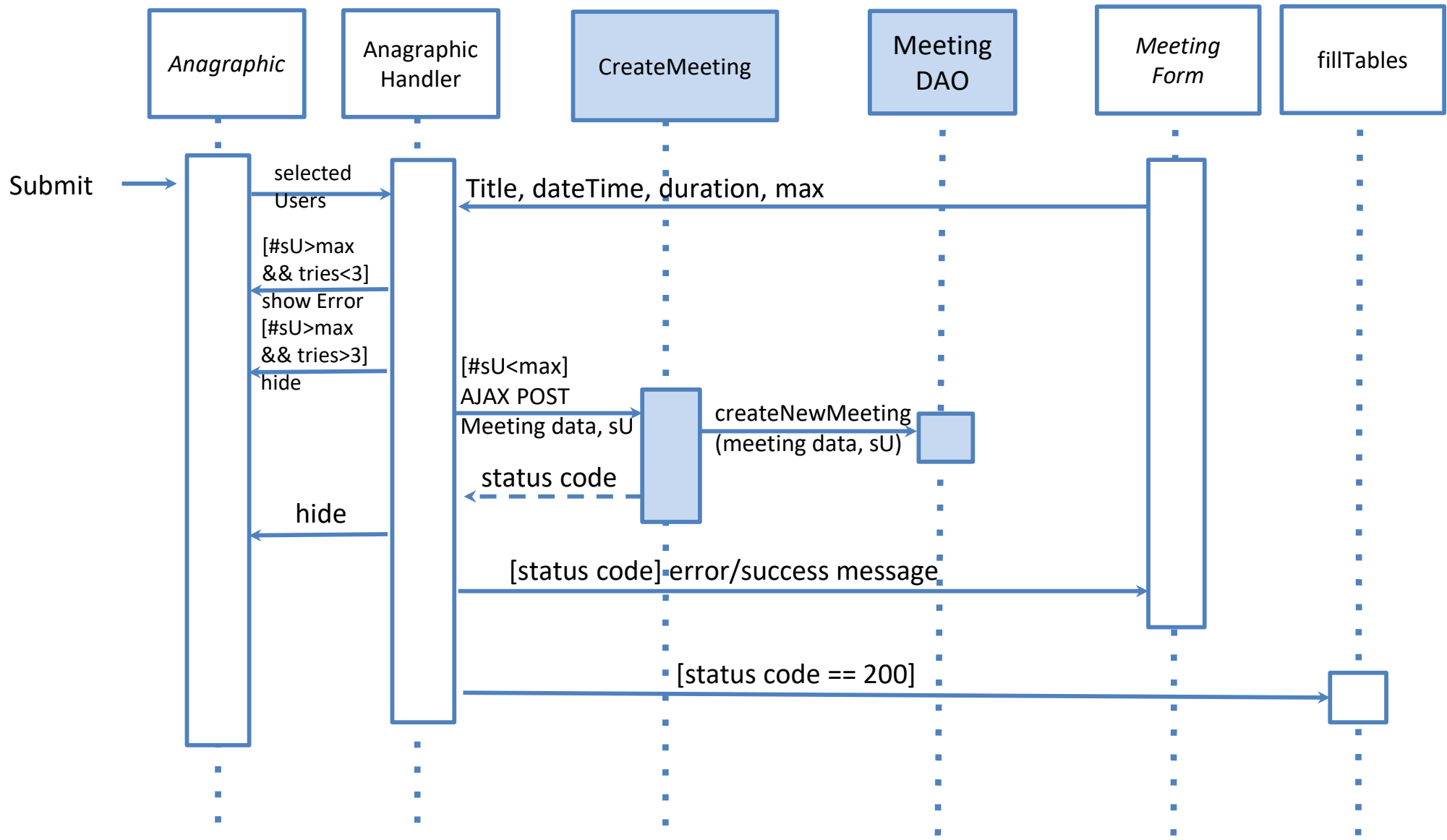
Evento: apertura anagrafica



sU = selectedUsers

#sU = selectedUsers.length

Evento: creazione di una riunione



Evento: logout

