

TIW Project #5

Riunioni Online

Versione HTML Pure

Sironi Alessandro, 10680296 - 932033

Riunioni Online

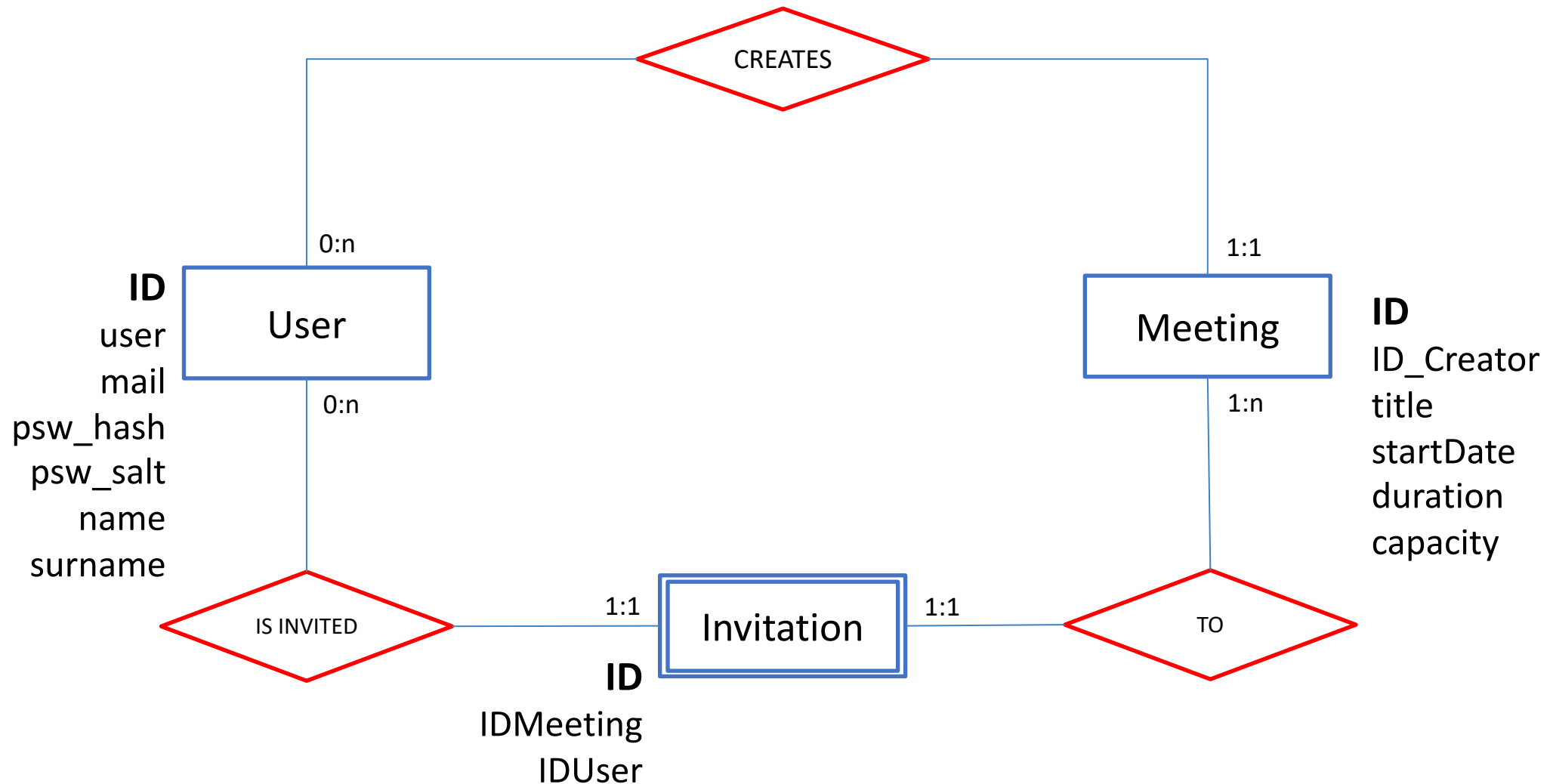
Un'applicazione web consente la gestione di riunioni online. L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password". La registrazione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

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

Design Database – Modello ER



Database Schema

```
CREATE TABLE User (  
  ID INT PRIMARY KEY NOT NULL  
  AUTO_INCREMENT,  
  mail VARCHAR(100) NOT NULL,  
  user VARCHAR(100) NOT NULL,  
  psw_hash VARBINARY(128) NOT NULL,  
  psw_salt VARBINARY(16) NOT NULL,  
  name VARCHAR(100) NOT NULL,  
  surname VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Meeting (  
  ID INT PRIMARY KEY NOT NULL  
  AUTO_INCREMENT,  
  ID_Creator INT NOT NULL,  
  title VARCHAR(100) NOT NULL,  
  startDate DATETIME NOT NULL,  
  duration INT NOT NULL,  
  capacity INT NOT NULL,  
  
  FOREIGN KEY (ID_Creator) REFERENCES User(ID)  
  ON UPDATE CASCADE  
  ON DELETE CASCADE  
);
```

Database Schema

```
CREATE TABLE Invitation (  
  ID INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
  IDMeeting INT NOT NULL,  
  IDUser INT NOT NULL,  
  
  FOREIGN KEY (IDMeeting) REFERENCES Meeting(ID)  
  ON UPDATE CASCADE  
  ON DELETE CASCADE,  
  FOREIGN KEY (IDUser) REFERENCES User(ID)  
  ON UPDATE CASCADE  
  ON DELETE CASCADE  
)
```

Application requirements analysis

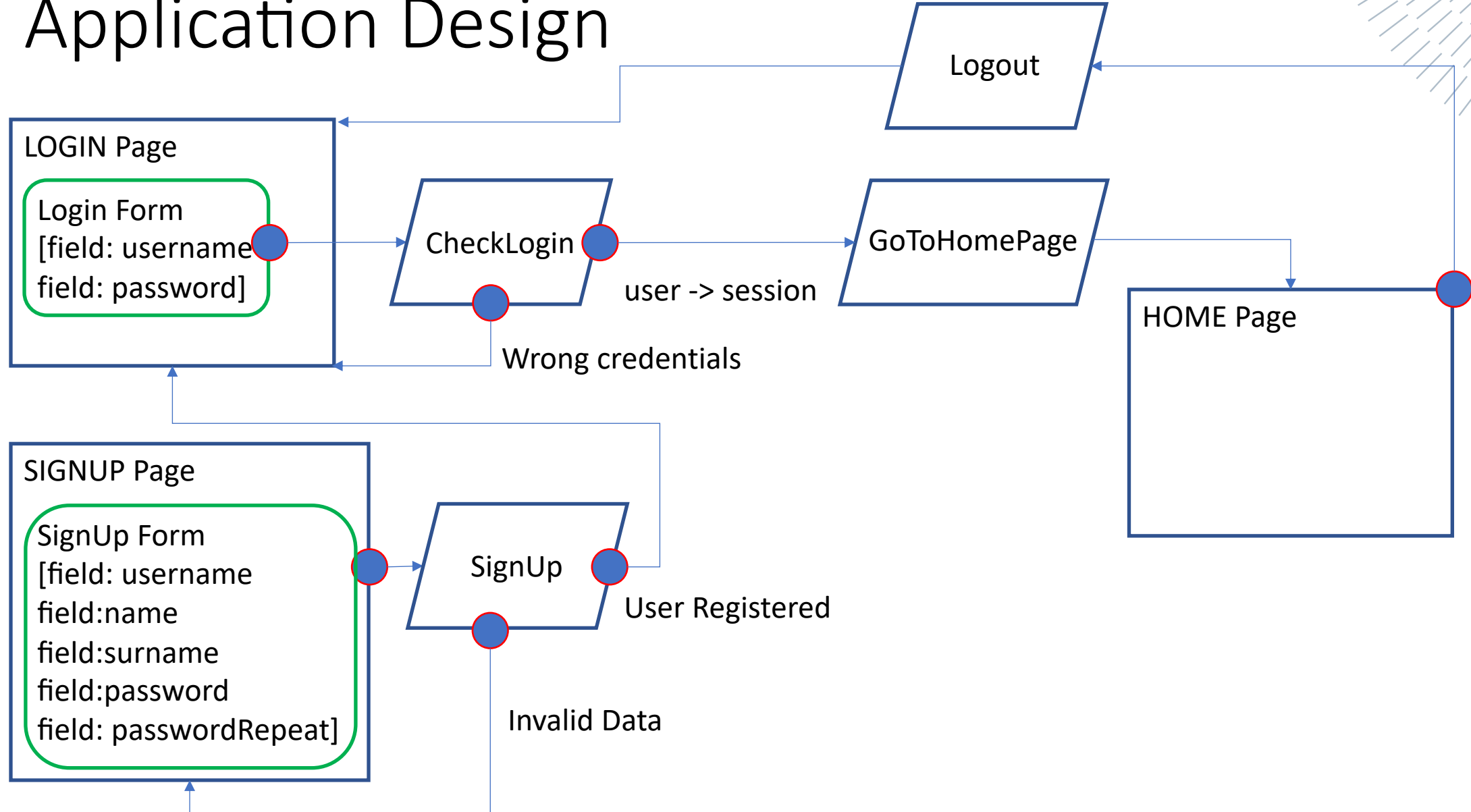
Un'applicazione web consente la gestione di riunioni online. L'applicazione supporta **registrazione** e **login** mediante una pagina pubblica con opportune **form**. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password".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

Views View components Events Actions

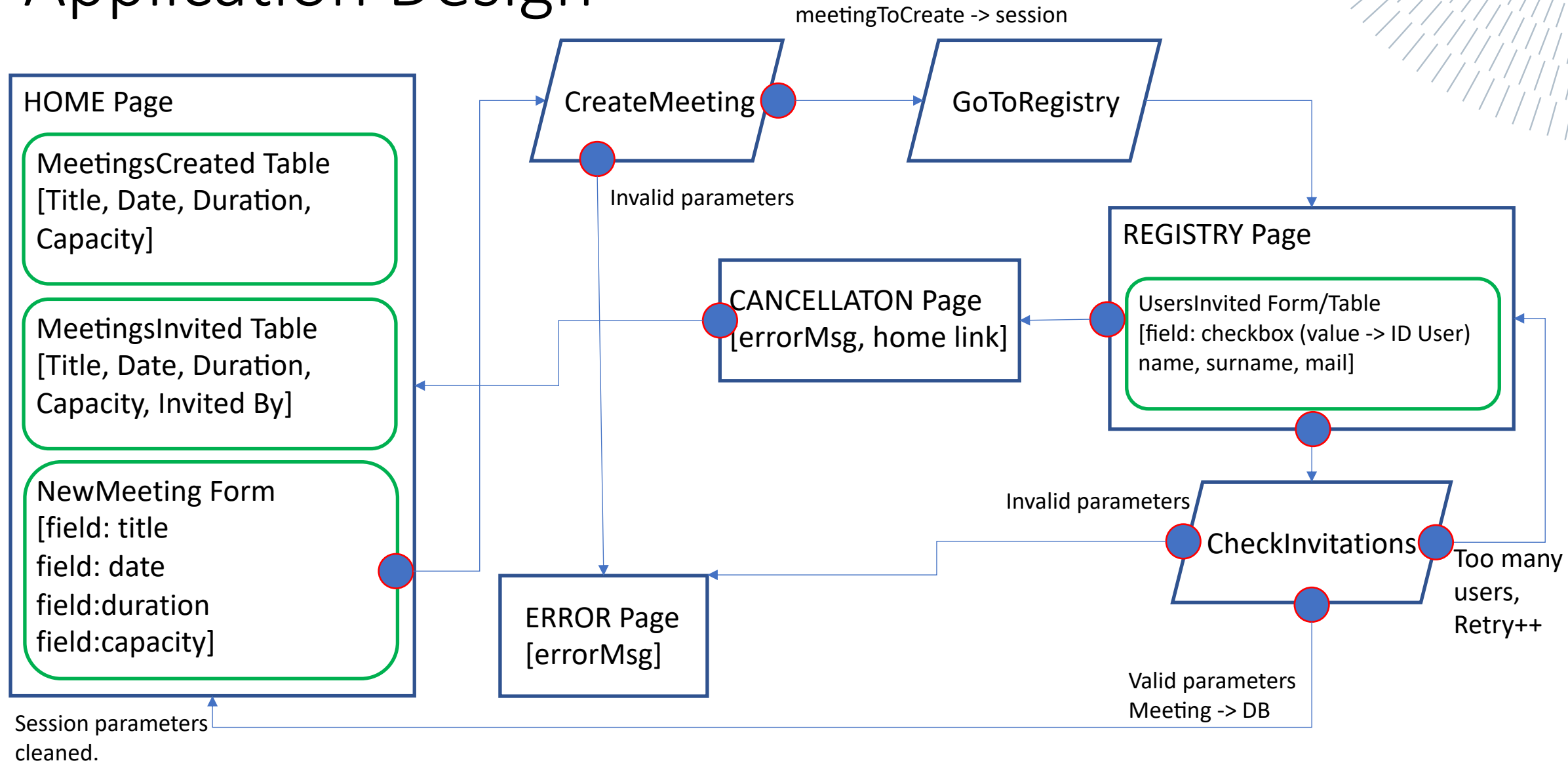
Completamento delle Specifiche

- La pagina di default contiene la form di login.
- Lo Username può essere un nickname oppure la mail; nel progetto si è scelto di utilizzare un campo aggiuntivo.
- A seguito del login, personalizzare la pagina con un messaggio all'utente.
- Tutti i dati delle form sono obbligatori e devono rispettare il formato consono.
- Le riunioni non possono essere create con date antecedenti a quella attuale.

Application Design



Application Design



Components

- Model Objects (Beans)

- User
- Meeting
- Invitation

- DAOs

- UserDao
 - checkCredentials (usr, pwd)
 - registerUser (mail, usr, pwd, name, surname)
 - getOtherUsers (int iduser) //SELECT ... WHERE ID != iduser
 - checkUserExists (usr)
 - checkUserIDExists (ID)
- MeetingDAO
 - createMeeting (ID_Creator, title, startDate, duration, capacity)
 - findMeetingsByOwner(ID_Owner)
 - getMeetingID (ID_Creator, title, startDate, duration, capacity)
 - getMeetingOwnerUsername (IDMeeting)
- InvitationDAO
 - inviteUser (IDMeeting, IDUserToInvite)
 - findMeetingsByInvitation (IDUser)

- Controllers (Servlets)

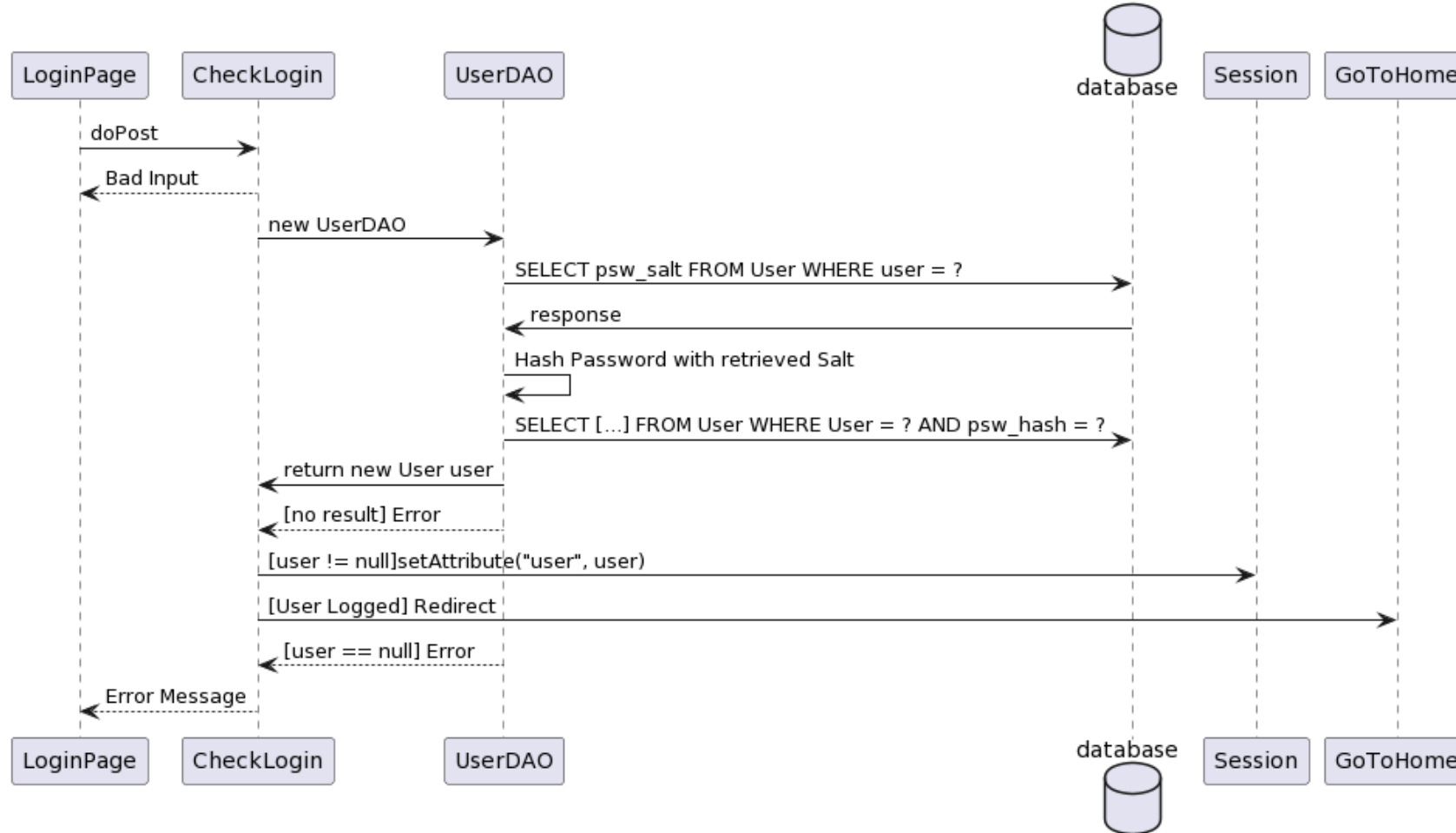
- CheckInvitations
- CheckLogin
- CreateMeeting
- GoToErrorCreationMeeting
- GoToHomePage
- GoToRegistry
- Logout
- SignUp
- (filter) AuthenticatedCheck (Filter check will be omitted from Events UML for brevity)
- (utils) ConnectionHandler

Components

- Views
 - index.html (login)
 - signup.html
 - home.html
 - registry.html
 - error.html

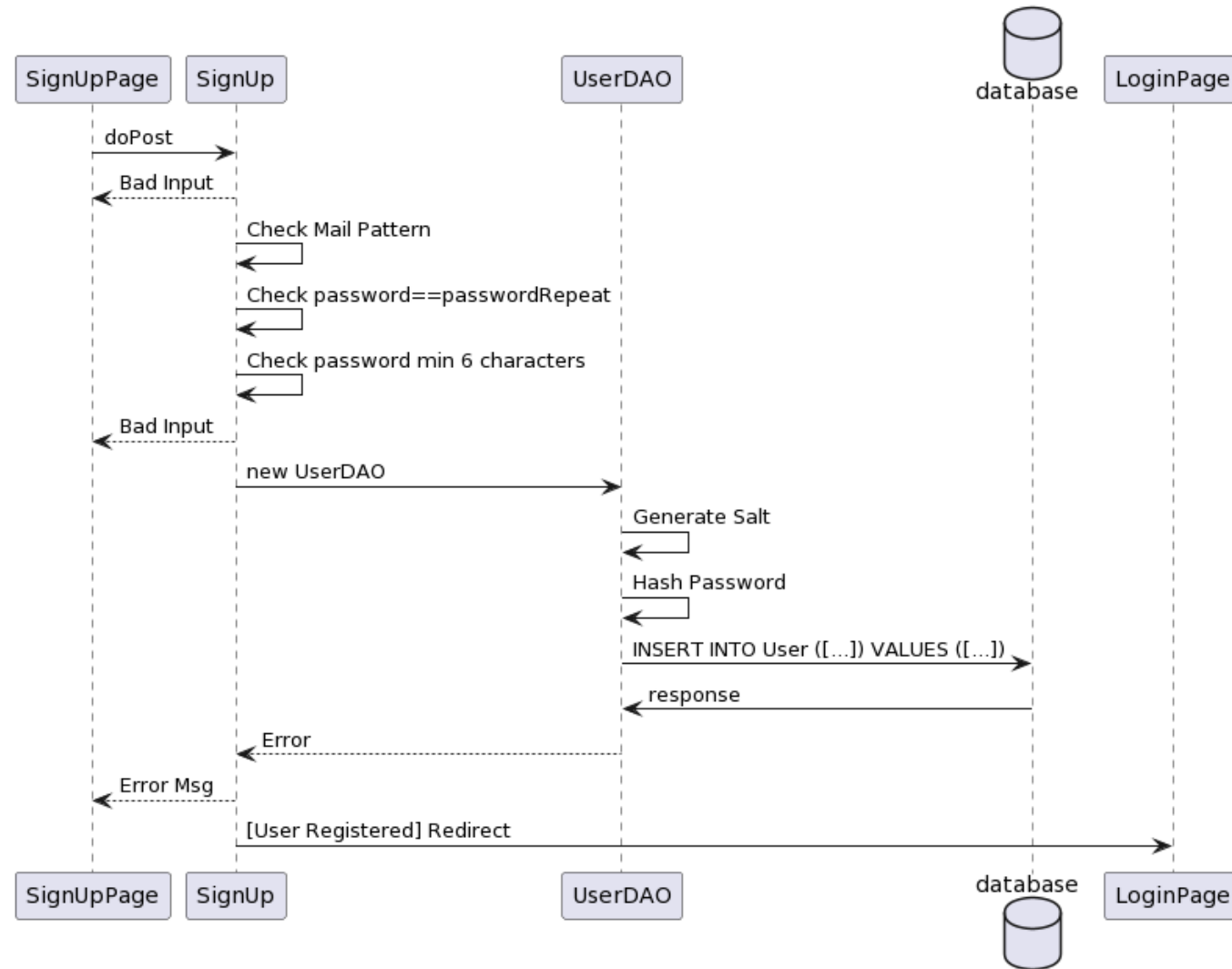
Event: Login

Form:
username,
password

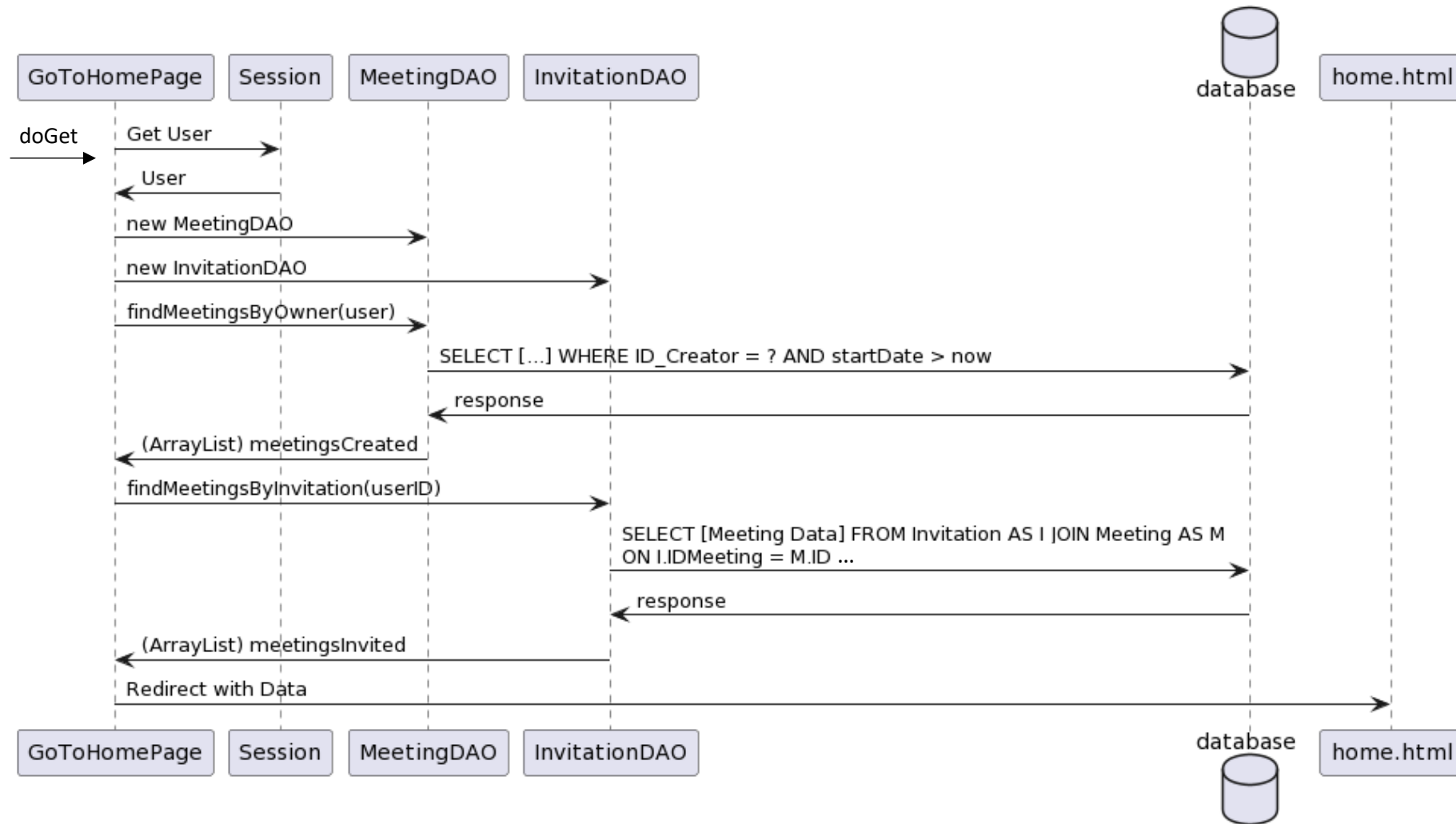


Event: SignUp

Form:
username,
mail
name
surname
password
passwordRepeat

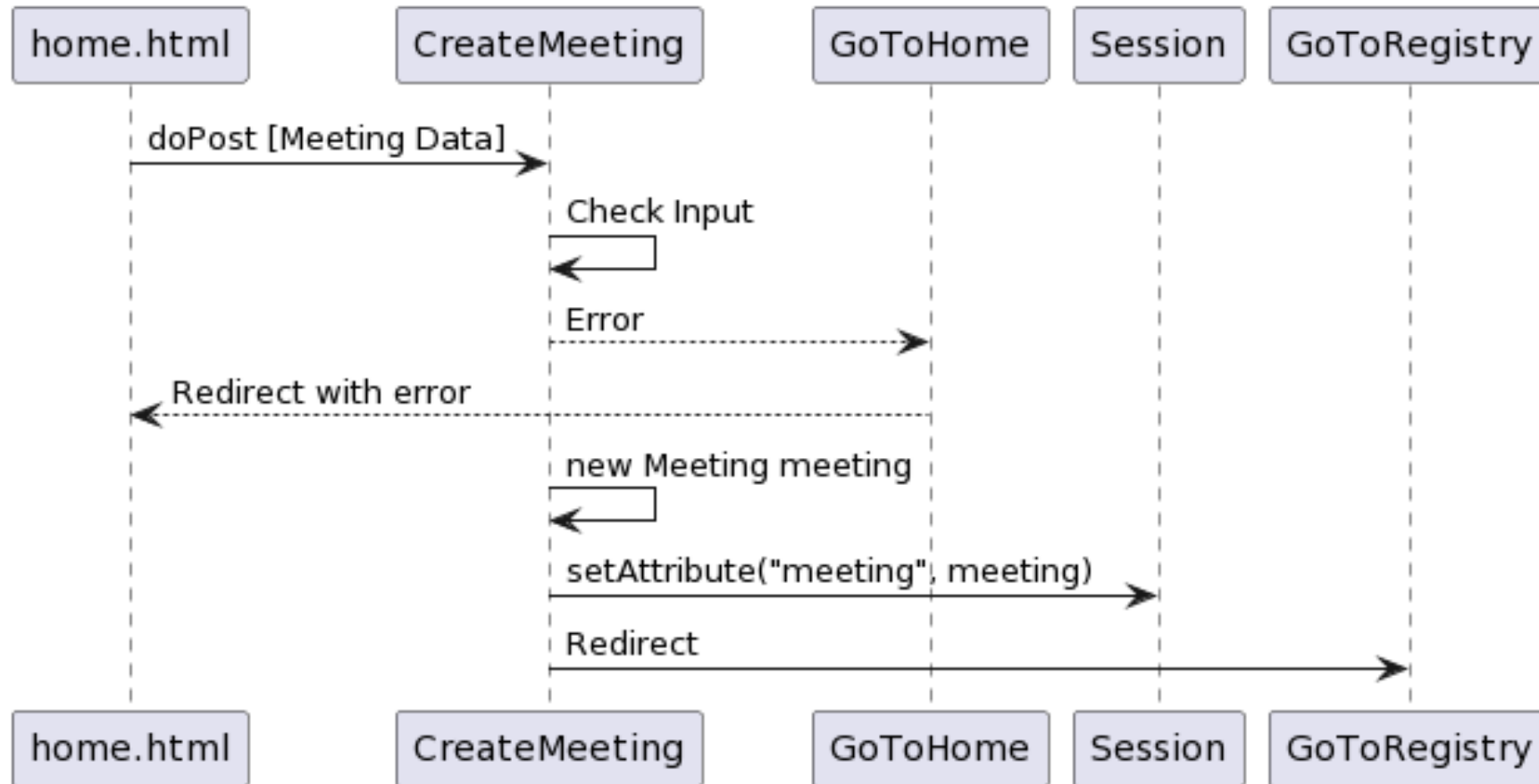


Event: GoToHomePage

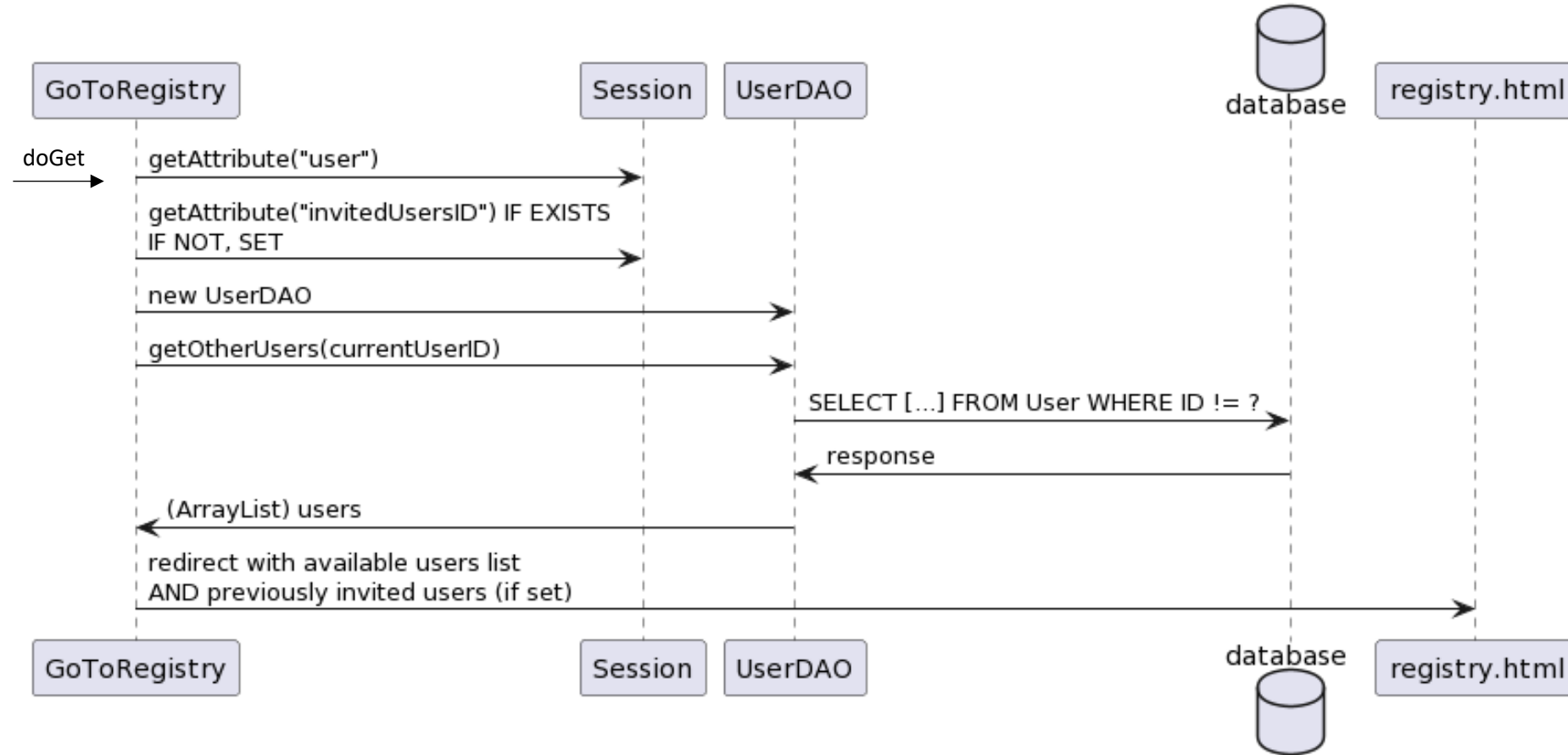


Event: CreateMeeting

Form:
title,
startDate,
duration
capacity

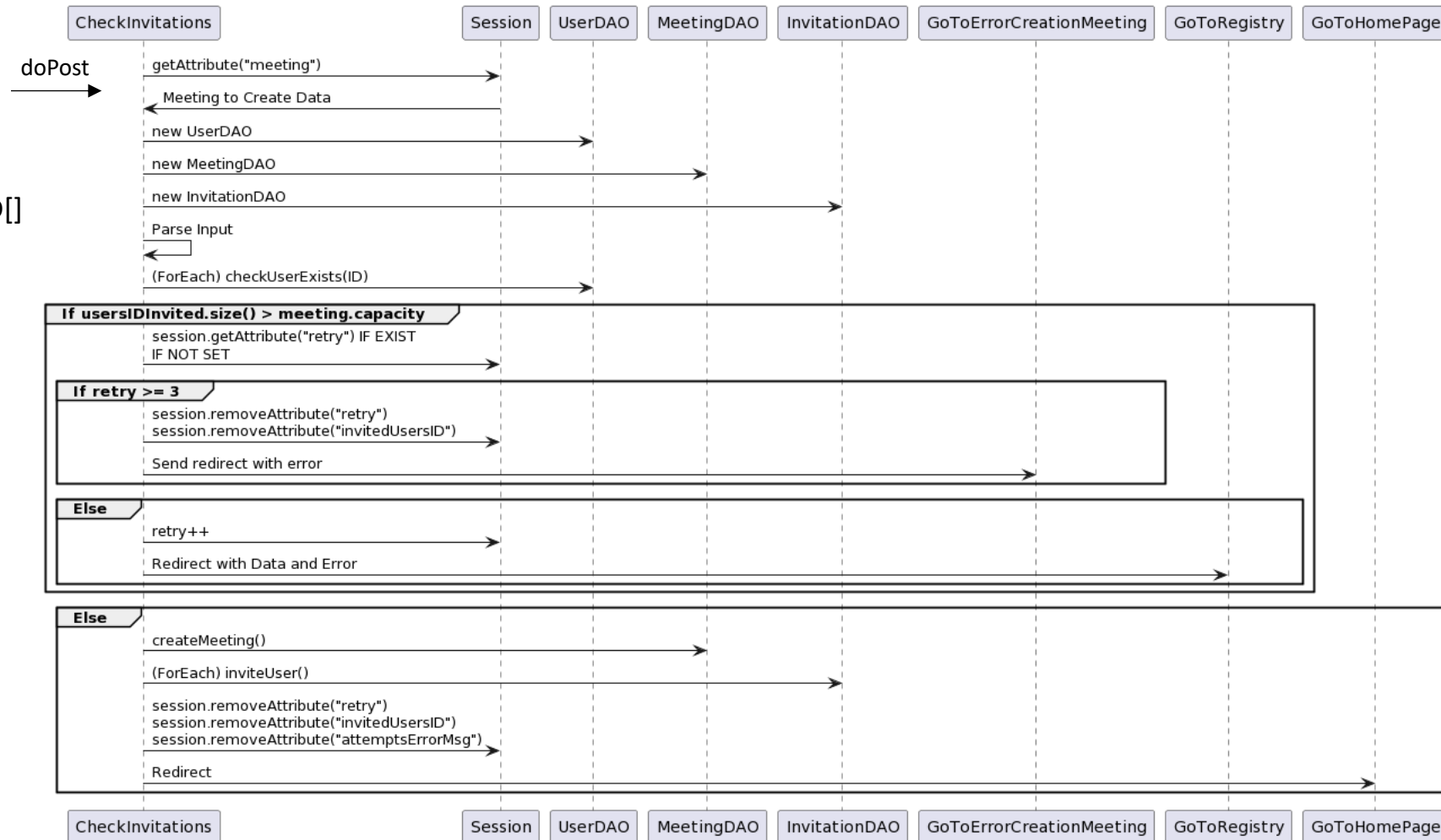


Event: GoToRegistry



Event: CheckInvitations

Form:
(Array) usersInvitedID[]



* Responses and Error checking omitted for brevity

TIW Project #5

Riunioni Online

Versione RIA

Sironi Alessandro, 10680296 - 932033

Modifiche

- 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.

Views View components Events Actions

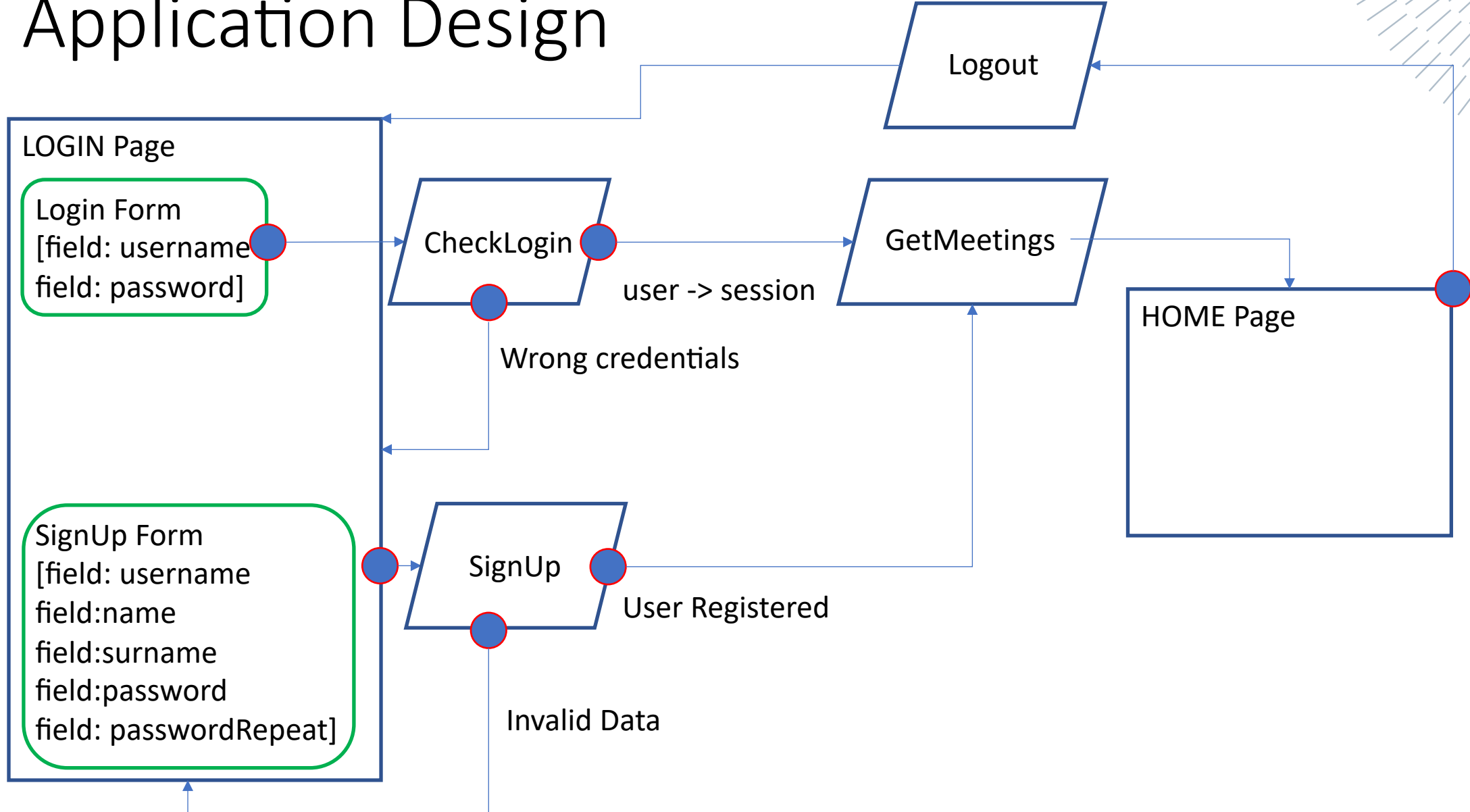
Components (RIA)

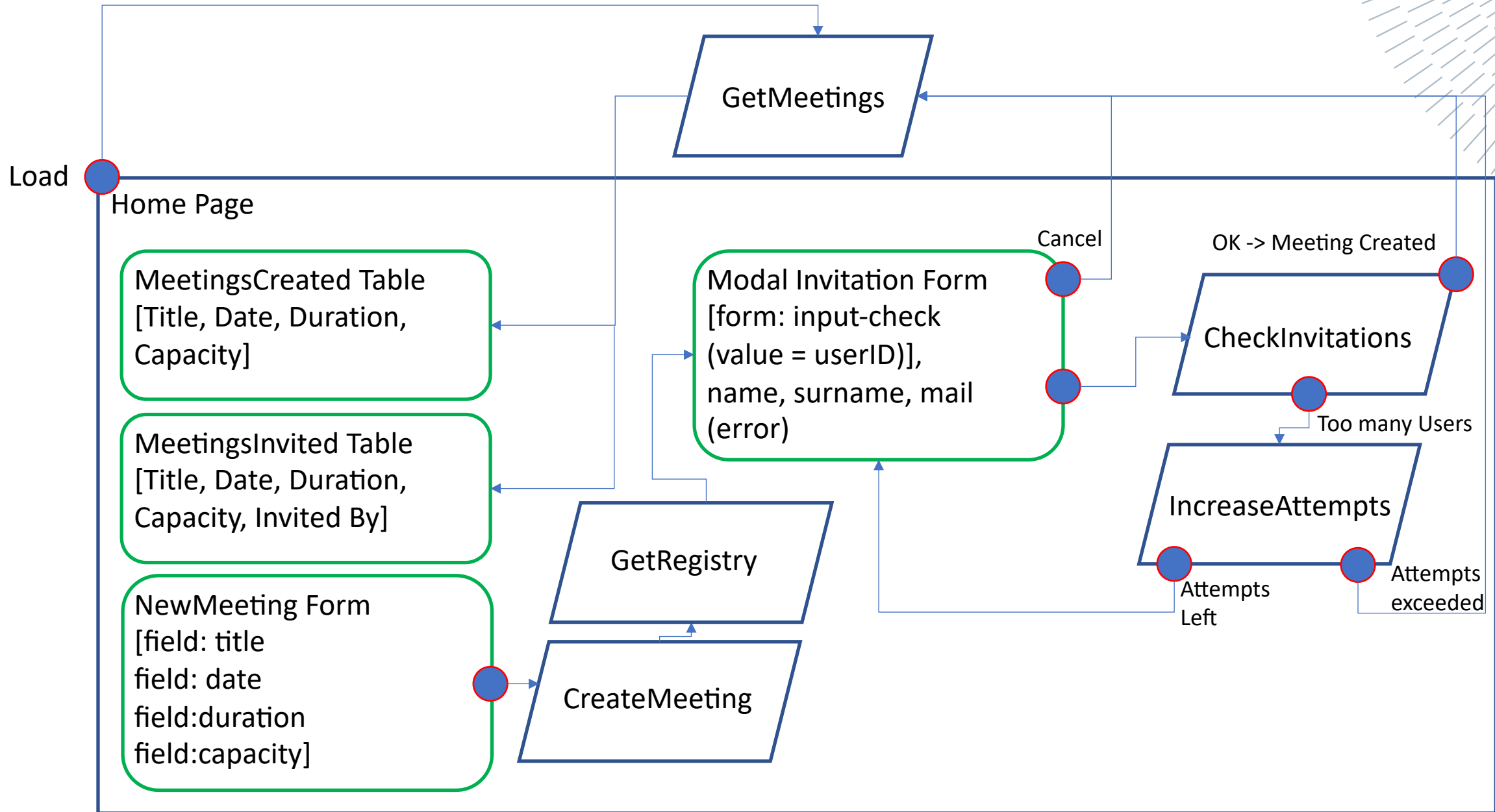
- Model Objects (Beans)
 - User
 - Meeting
 - Invitation
- DAOs
 - UserDao
 - checkCredentials (usr, pwd)
 - registerUser (mail, usr, pwd, name, surname)
 - getOtherUsers (int iduser) //SELECT ... WHERE ID != iduser
 - checkUserExists (usr)
 - checkUserIDExists (ID)
 - MeetingDAO
 - createMeeting (ID_Creator, title, startDate, duration, capacity)
 - findMeetingsByOwner(ID_Owner)
 - getMeetingID (ID_Creator, title, startDate, duration, capacity)
 - getMeetingOwnerUsername (IDMeeting)
 - InvitationDAO
 - inviteUser (IDMeeting, IDUserToInvite)
 - findMeetingsByInvitation (IDUser)
- Controllers (Servlets)
 - CheckInvitations
 - CheckLogin
 - CreateMeeting
 - ~~• GoToErrorCreationMeeting~~
 - GoToHomePage -> **GetMeetings**
 - GoToRegistry -> **GetRegistry**
 - **IncreaseAttempts**
 - Logout
 - SignUp
 - (filter) AuthenticatedCheck (Filter check will be omitted from Events UML for brevity)
 - (utils) ConnectionHandler

Components (RIA)

- Views
 - index.html (login & SignUp)
 - Home.html (WebApp)
 - Tables:
 - My Meetings
 - Invited To Meetings
 - Form: Create meeting
 - Modal: Modal Window

Application Design



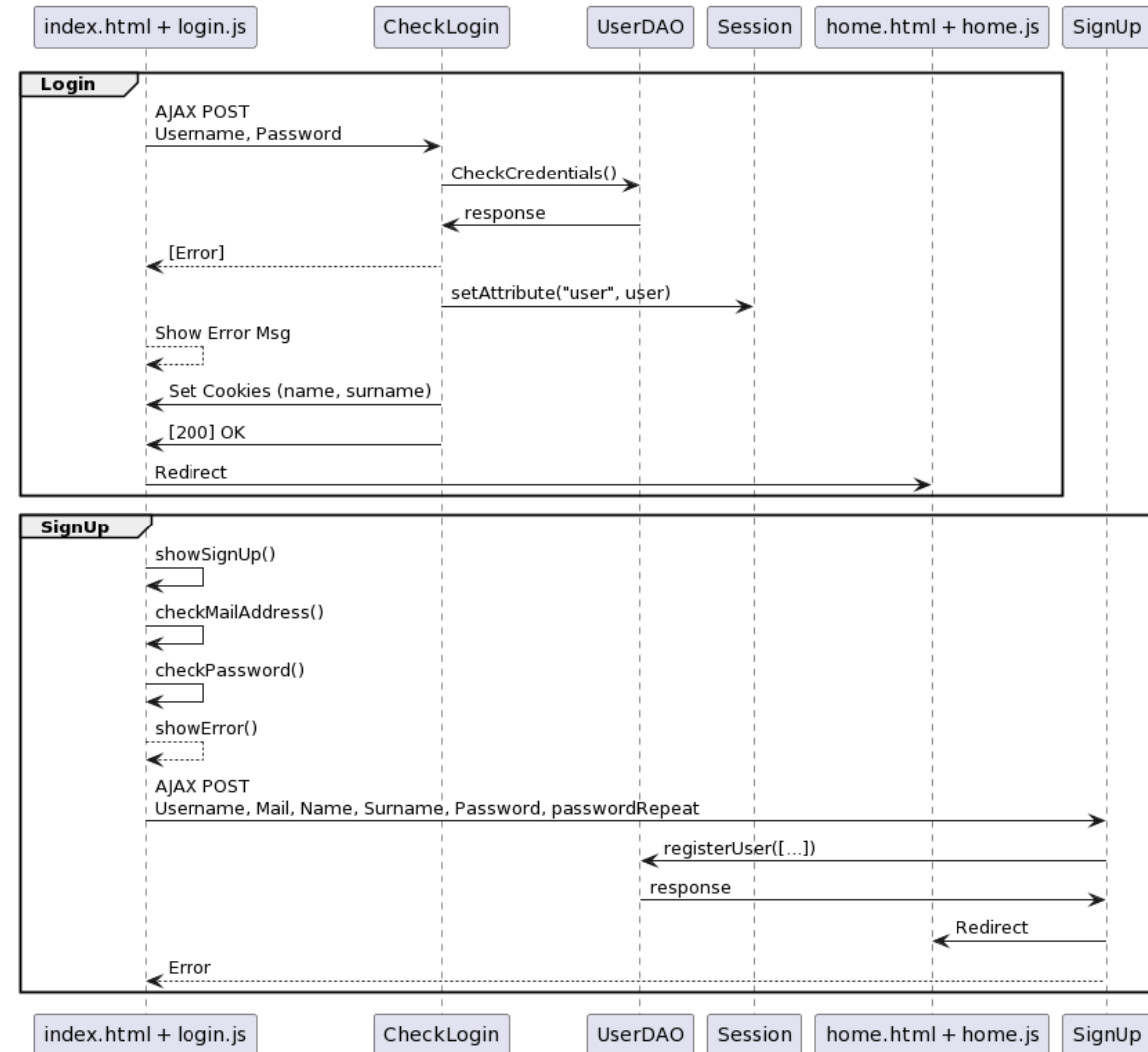


Events & Actions

Client side		Server side	
Evento	Azione	Evento	Azione
index → login form → submit	Controllo dati	POST username password	Controllo credenziali
Index -> showSignUp	Mostra form di iscrizione		
Index -> SignUp	Check correttezza e Invio	POST username mail name surname password passwordRepeat	Controllo validità e iscrizione
Home page → load	Aggiorna view con riunioni.	GET (Meetings)	Estrazioni riunioni da DB
Nuovo Meeting -> submit	Controllo dati, apertura finestra modale	POST (Meeting Data)	Controllo dati, salvataggio in sessione meetingToCreate
Apertura Finestra Modale -> load	Caricamento utenti iscritti alla piattaforma	GET (Users)	Estrazione utenti da DB
Form inviti → submit	Controllo numero partecipanti, gestione numero tentativi	POST (Invitati, Tentativi)	Controllo numero e validità ID partecipanti, controllo tentativi
Close button → (re)load Home		GET (Meetings)	Estrazioni riunioni da DB
Logout		GET	Terminazione della sessione

Client side		Server side	
Evento	Controller	Evento	Servlet
index → login form → submit	login.js: makeCall()	POST username password	CheckCredentials
Index -> showSignUp	showSignUp()		
Index -> SignUp	login.js: makeCall(), pswErrorSignUp(), mailAddress()	POST username mail name surname password passwordRepeat	SignUp
Home page → load	home.js: makeCall(), MeetingsCreated(), MeetingsInvited()	GET (Meetings)	GetMeetings
Nuovo Meeting -> submit	meetingCreation.js: makeCall(), checkMeetingInfo()	POST (Meeting Data)	CreateMeeting
Apertura Finestra Modale -> load	meetingCreation.js: showModal(), makeCall()	GET (Users)	GetRegistry
Form inviti → submit	meetingCreation.js: makeCall(), getInvitationAttempts(), ...	POST (Invitati, Tentativi)	CheckInvitations
Close button → (re)load Home	home.js: makeCall(), MeetingsCreated(), MeetingsInvited()	GET (Meetings)	GetMeetings
Logout		GET	Logout

Event: Login / SignUp



Events: Home

