Progetto di Tecnologie Informatiche per il Web

Progetto per la catalogazione di immagini

William Zeni

Technical Specifications

HTML - Specs

Un'applicazione permette all'utente (ad esempio il curatore di un catalogo online di immagini) di etichettare le immagini allo scopo di consentire la ricerca in base alla categoria. Dopo il login, l'utente accede a una pagina HOME in cui compare un albero gerarchico di categorie. Le categorie non dipendono dall'utente e sono in comune tra tutti gli utenti. Un esempio di un ramo dell'albero è il seguente:

9 Mitologia classica e storia antica >>sposta
91 Divinità della mitologia classica >>sposta
911 Divinità del cielo >>sposta
9111 Giove >>sposta
91111 Attributi di Giove >>sposta
9112 Giunone >>sposta
912 Divinità degli inferi >>sposta
9121 Plutone >>sposta
9122 Ecate >>sposta

L'utente può inserire una nuova categoria nell'albero. Per fare ciò usa una form nella pagina HOME in cui specifica il nome della nuova categoria e sceglie la categoria padre. L'invio della nuova categoria comporta l'aggiornamento dell'albero: la nuova categoria è appesa alla categoria padre come ultimo sottoelemento. Alla nuova categoria viene assegnato un codice numerico che ne riflette la posizione (ad esempio, la nuova categoria Mercurio, figlia della categoria "911 Divinità del cielo" assume il codice 9113). Per semplicità si ipotizzi che per ogni categoria il numero massimo di sottocategorie sia 9, numerate da 1 a 9. Dopo la creazione di una categoria, la pagina HOME mostra l'albero aggiornato. L'utente può spostare di posizione una categoria: per fare ciò clicca sul link "sposta" associato alla categoria da spostare. A seguito di tale azione l'applicazione mostra, sempre nella HOME page, l'albero con evidenziato il sotto albero attestato sulla categoria da spostare: tutte le altre categorie hanno un link "sposta qui". Ad esempio, a seguito del click sul link "sposta" associato alla categoria "9111 Giove" l'applicazione visualizza l'albero come segue:

9 Mitologia classica e storia antica >>sposta qui
91 Divinità della mitologia classica >>sposta qui
911 Divinità del cielo >>sposta qui
9111 Giove
91111 Attributi di Giove
9112 Giunone >>sposta qui
912 Divinità degli inferi >>sposta qui
9121 Plutone >>sposta qui
9122 Ecate >>sposta qui

La selezione di un link "sposta qui" comporta l'inserimento della categoria da spostare come ultimo figlio della categoria destinazione. Ad esempio, la selezione del link "sposta qui" della categoria "912 Divinità degli inferi" comporta la seguente modifica dell'albero:

9 Mitologia classica e storia antica >>sposta
91 Divinità della mitologia classica >>sposta
911 Divinità del cielo >>sposta
9111 Giunone >>sposta
912 Divinità degli inferi >>sposta
9121 Plutone >>sposta
9122 Ecate >>sposta
9123 Giove >>sposta
91231 Attributi di Giove >>sposta

Le modifiche effettuate da un utente e salvate nella base di dati diventano visibili agli altri utenti.

Technical Specifications

RIA - Specs

Si realizzi un'applicazione client server web che modifica le specifiche precedenti come segue:

- Dopo il login dell'utente, 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 funzione di spostamento di una categoria è realizzata mediante drag & drop.
- A seguito del drop della categoria da spostare compare una finestra di dialogo con cui l'utente può confermare o cancellare lo spostamento. La conferma produce l'aggiornamento a lato client dell'albero.
- L'utente realizza spostamenti anche multipli a lato client. A seguito del primo spostamento compare un bottone SALVA la cui pressione provoca l'invio al server dell'elenco degli spostamenti realizzati (NON dell'intero albero). L'invio degli spostamenti produce l'aggiornamento dell'albero nella base dei dati e la comparsa di un messaggio di conferma dell'avvenuto salvataggio.

Database- Local Schema

User Table

```
CREATE TABLE User (
        username varchar(25) NOT NULL,
        passwd char(64) NOT NULL,
        name varchar(20) NOT NULL,
        surname varchar(20) NOT NULL,
        PRIMARY KEY (username)
)
```

Topic Table

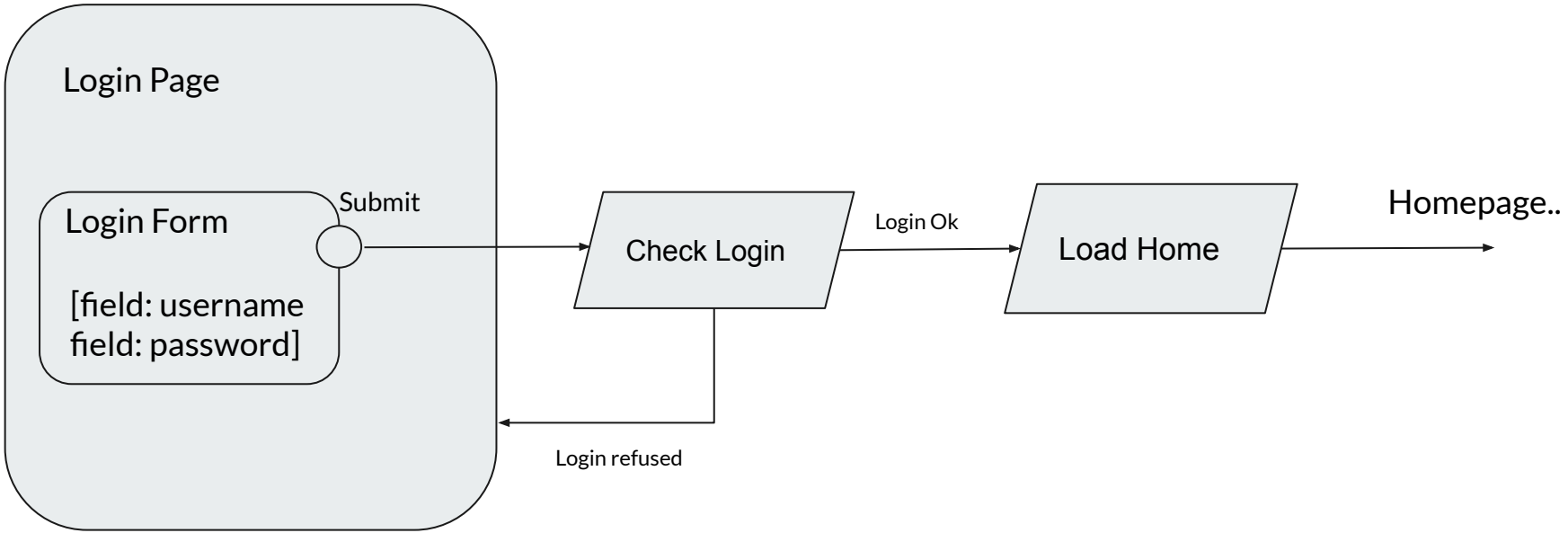
```
CREATE TABLE Topic (
        id int NOT NULL,
        name varchar(30) NOT NULL,
        idFather int NOT NULL,
        PRIMARY KEY (id),
        CONSTRAINT 'check_IdFather' CHECK
        (((not((cast(id as char) like '%0%'))) and
        ((idFather * 10) < id) and (((idFather * 10) + 10) > id)))
)
```

HTML Project Version

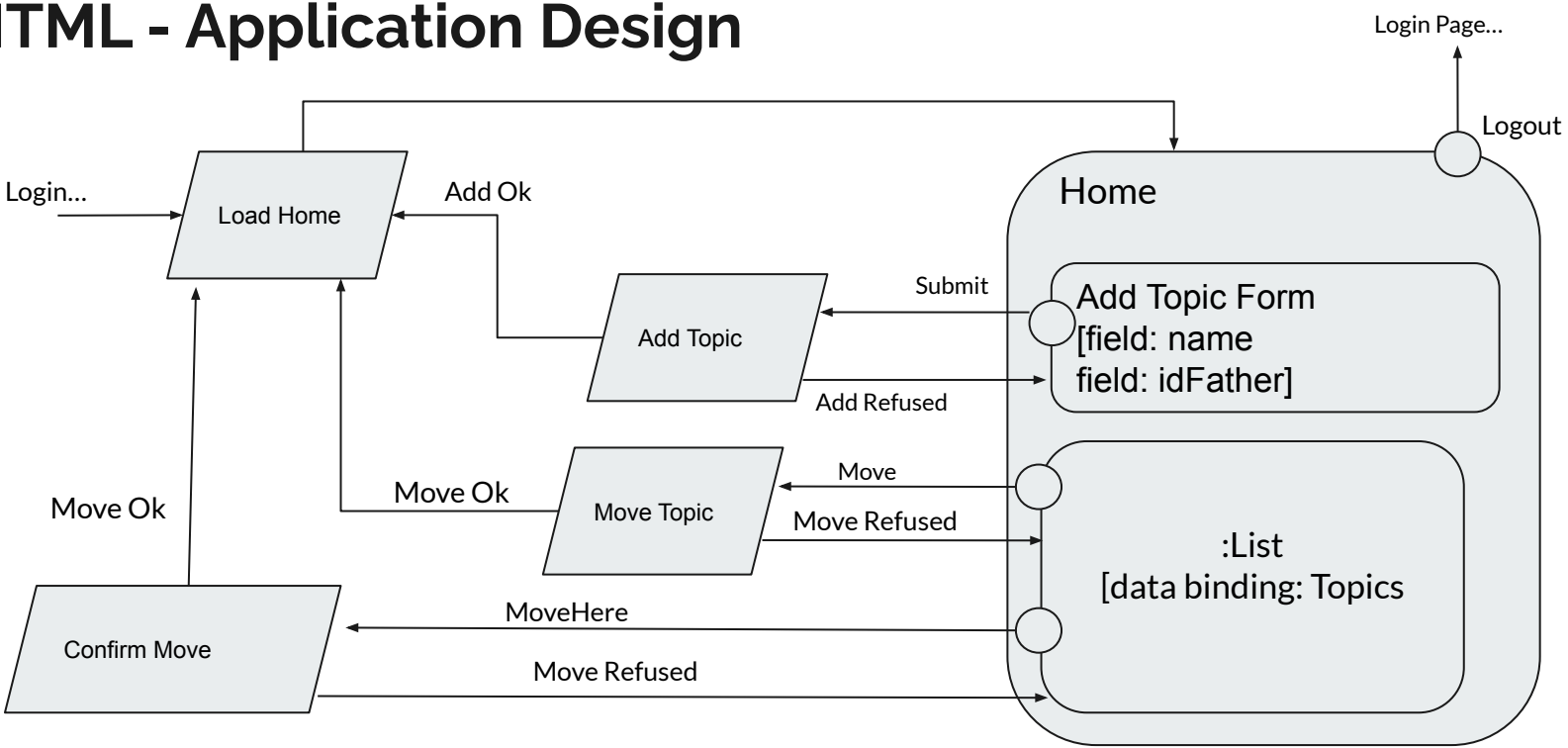
Tools

- IDEA IntelliJ IDE
- MySQL Workbench - SQL Database Manager System
- Apache Tomcat - Open Source Web Server
- Thymeleaf - Server side Template Engine

HTML - Application Design



HTML - Application Design



HTML - Components

Model Objecs (Beans)

- Topic
- User
- AddTopicForm : formBean
- LoginForm: formBean

Filters

- UserChecker
- MoveFilter
- MoveConfirmFilter

Controllers (Servlets)

- LoadHome
- AddTopic
- LoginHandler
- Logout
- MoveTopic
- MoveTopicConfirm

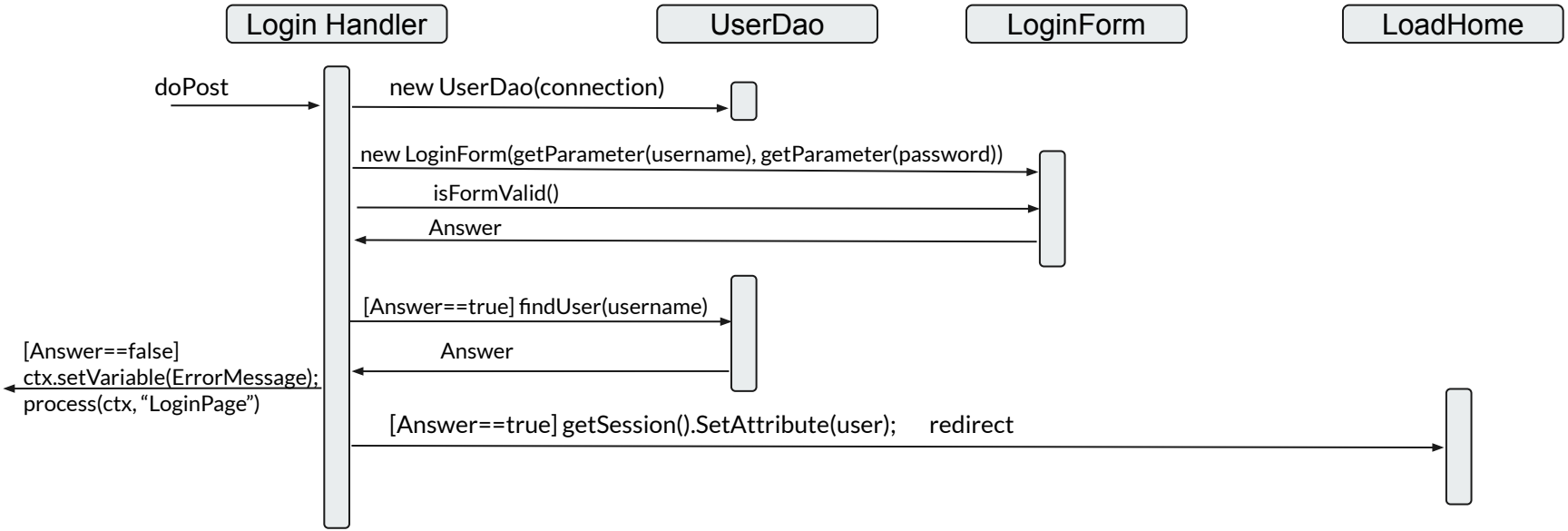
Views (Templates)

- HomePage
- LoginPage
- topicFrame (fragment)
- moveFrame (fragment)

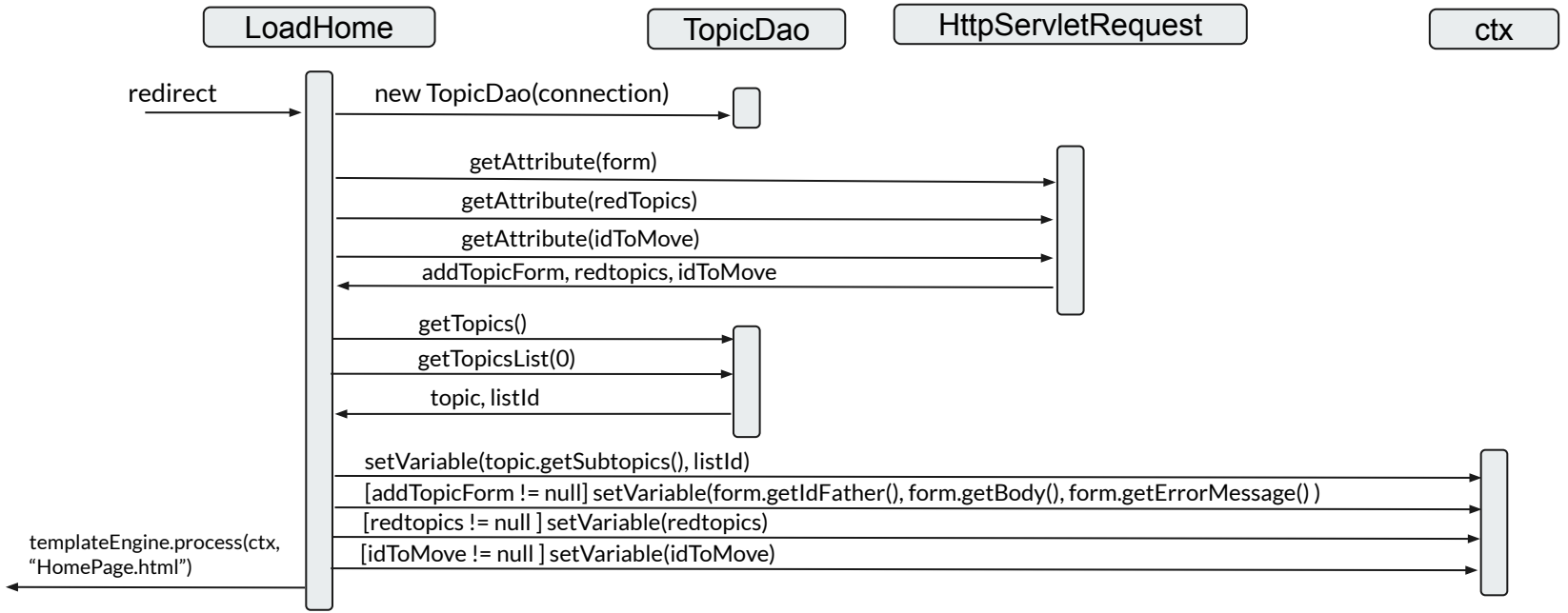
Data Access Objects (Classes)

- TopicDao
  - getTopics(): Topic
  - getTopicsList(): Integer List
  - addTopicDB(idFather, name)
  - isMySon(idFather, idSon): Boolean
  - idsExist(Ids) : Boolean
  - moveTopic(idToMove, idWhereToMove)
- UserDao
  - findUser(username) : Boolean
  - match(username, password): User

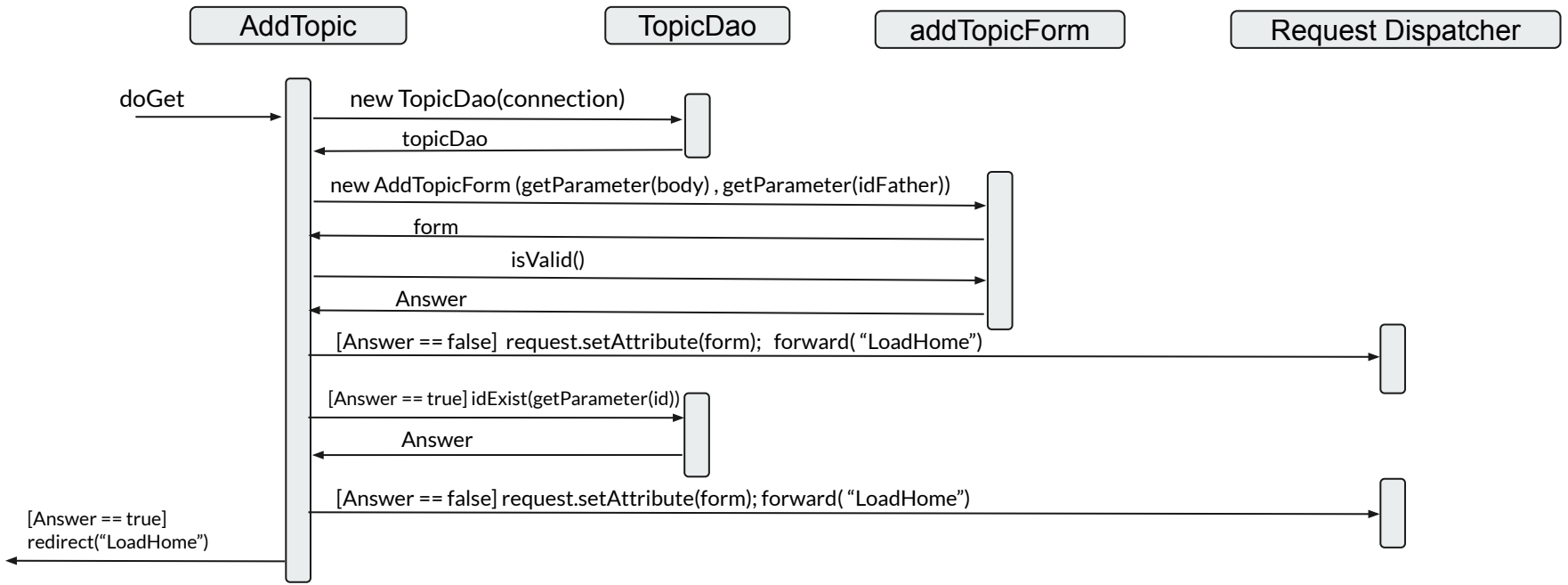
HTML - Event: Check Login



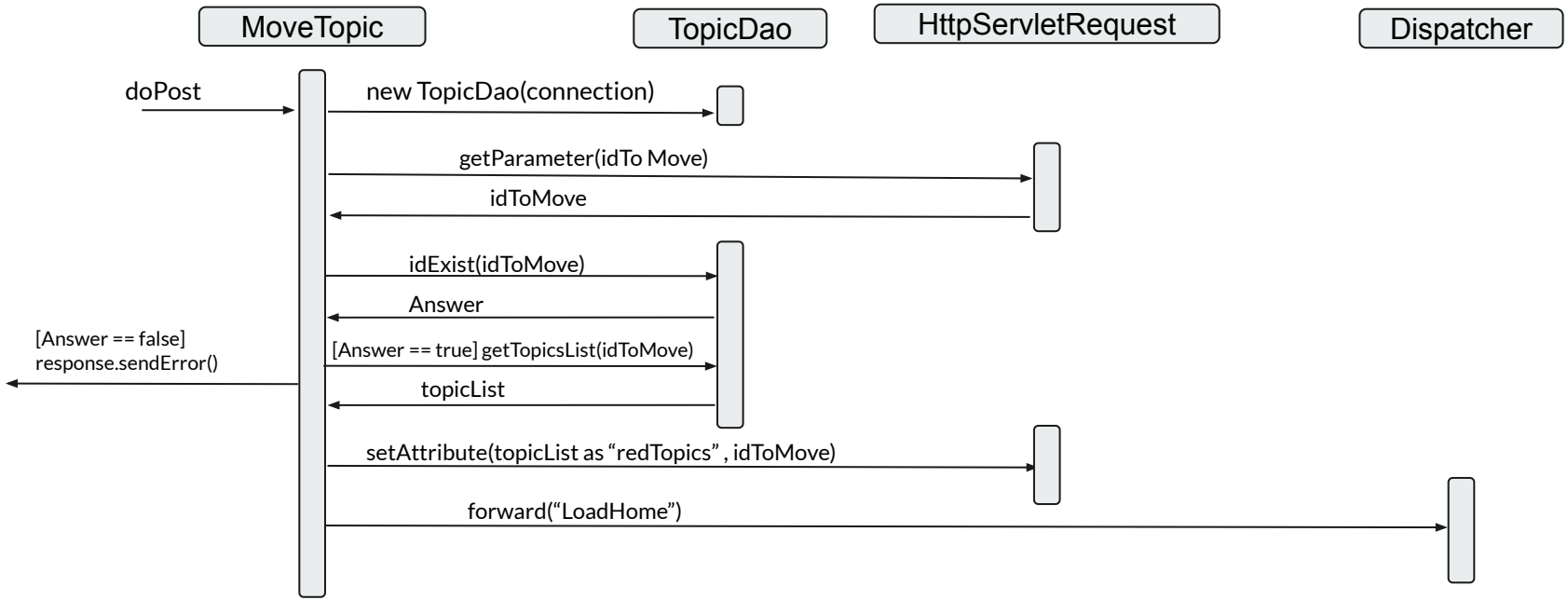
HTML - Event: Load Home



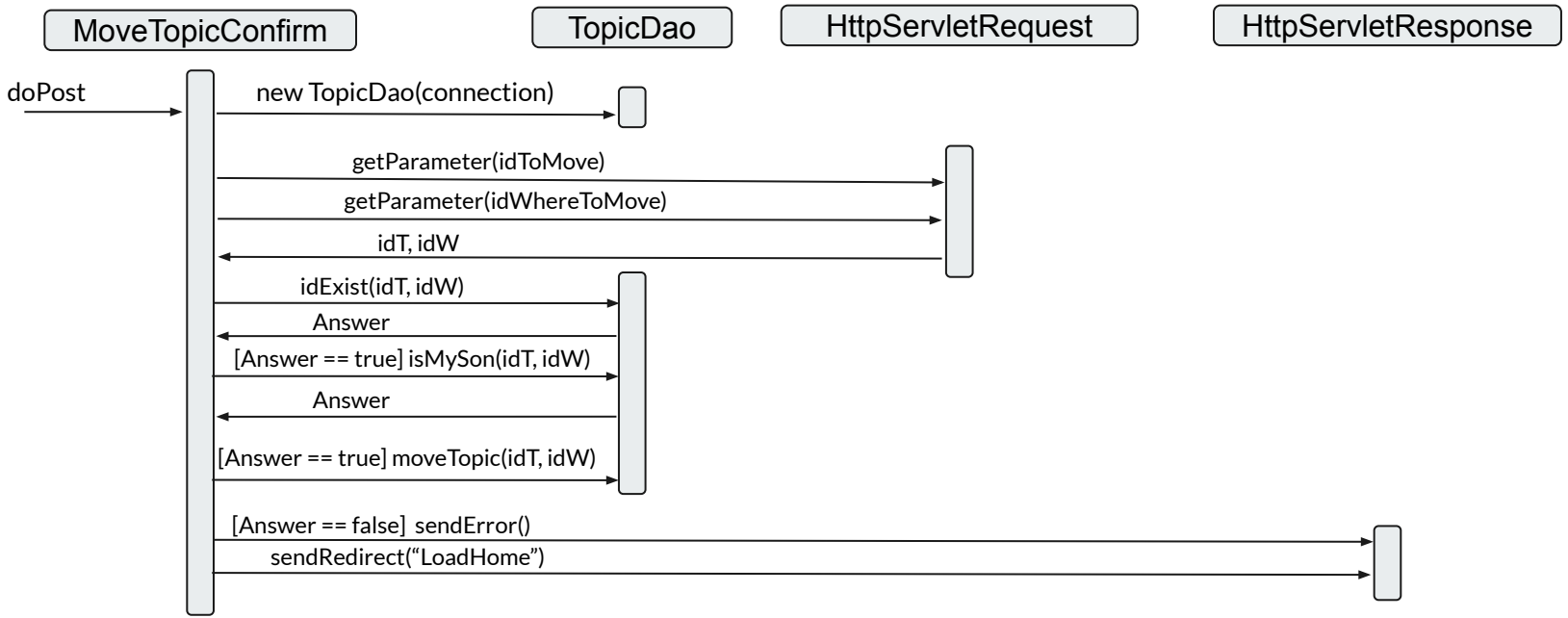
HTML - Event: Add Topic



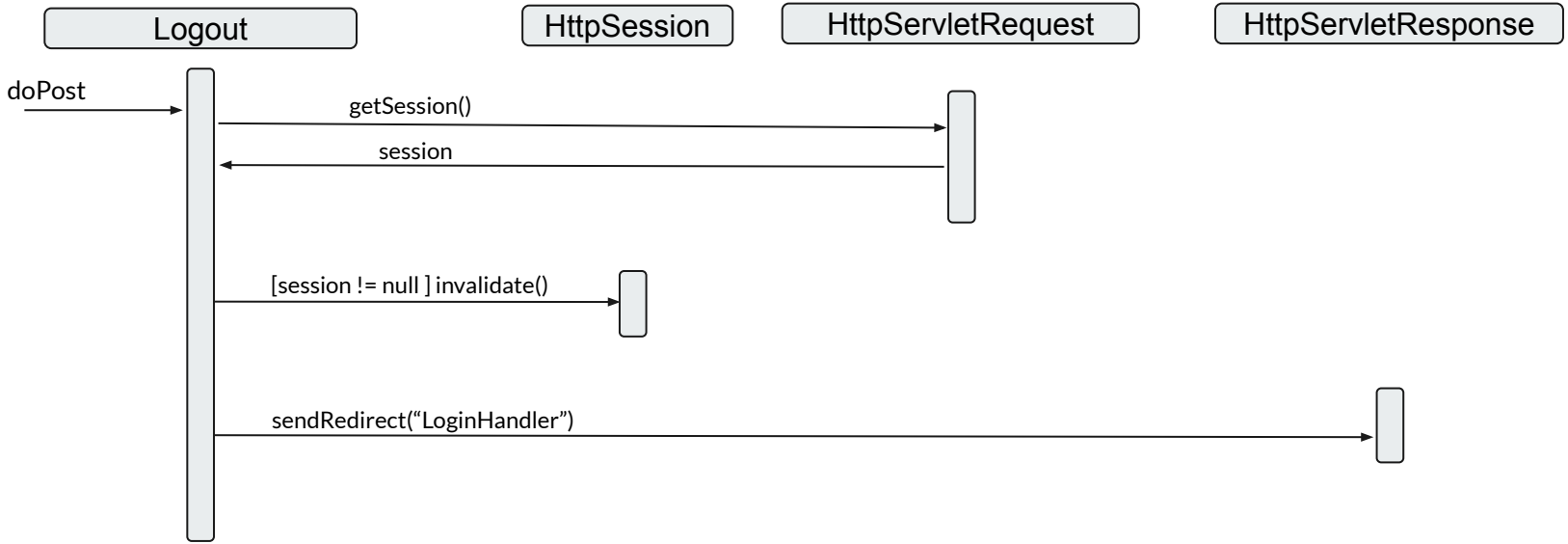
HTML - Event: Move Topic



HTML - Event: Confirm Move



HTML - Event: Logout



HTML - Filters Mapping

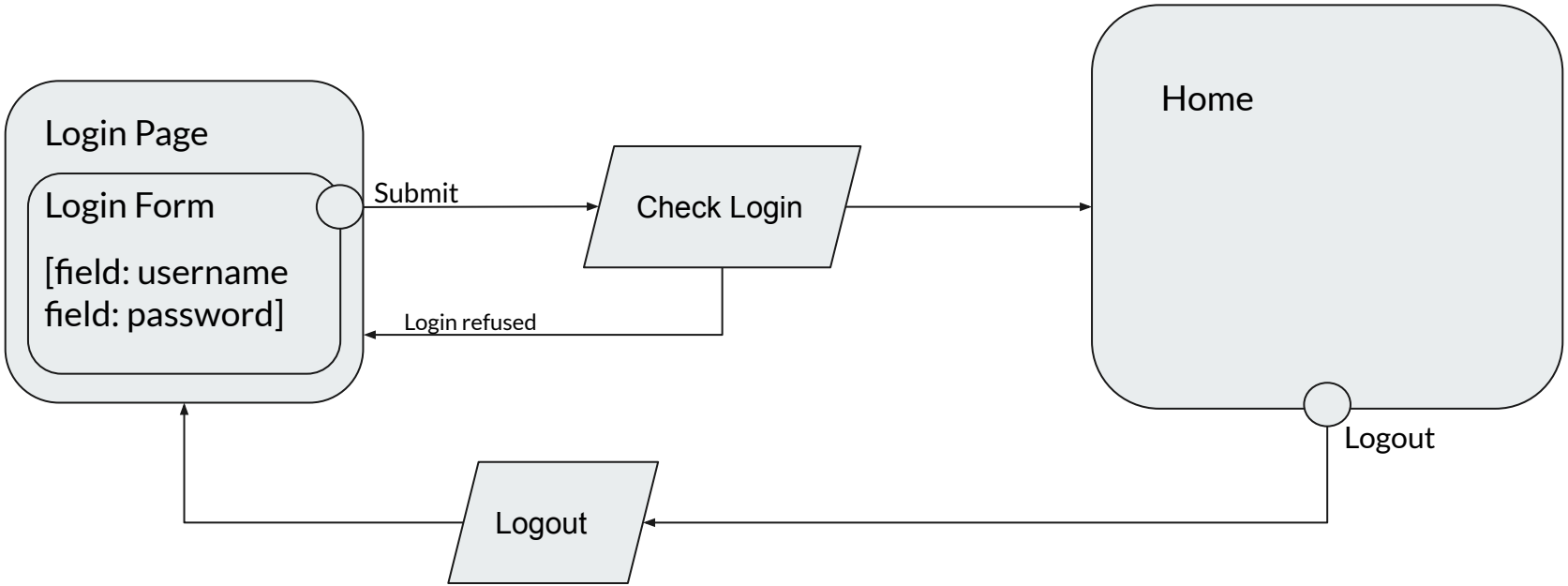
Filter Name	Servlets
UserChecker	LoadHome, AddTopic, MoveTopic, MoveTopicConfirm
MoveFilter	MoveTopic, MoveTopicConfirm
MoveConfirmFilter	MoveTopicConfirm

RIA Project Version

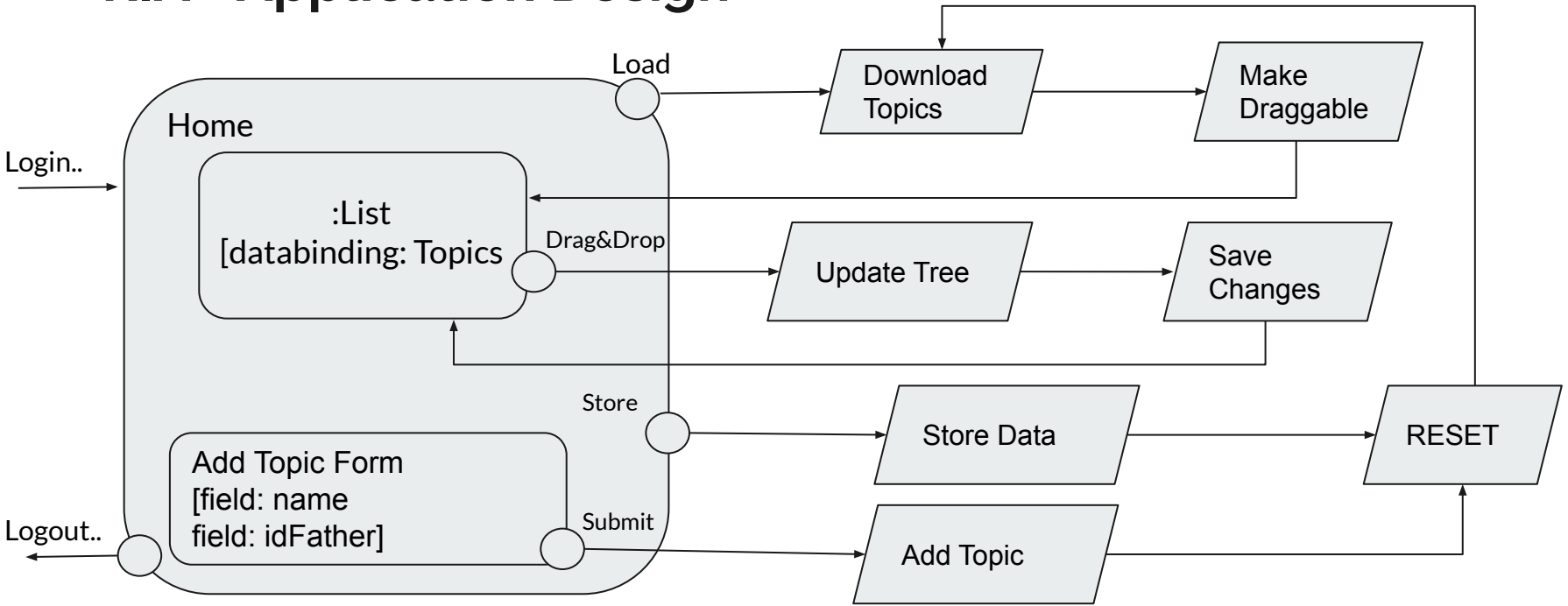
Tools

- IDEA IntelliJ IDE
- MySQL Workbench - SQL Database Manager System
- Apache Tomcat - Open Source Web Server

RIA - Application Design



RIA - Application Design



RIA - Components

Model Objecs (Beans)

- Move
- TopicJS
- UserJS

Filters

- UserCheckerJS

Controllers (Servlets)

- DownloadTopicsJS
- GetOptionsTopicsJS
- AddTopicJS
- LogoutJS
- StoreDataJS
- LoginHandlerJS

Views (Templates)

- HomePageJS
- LoginJS

Data Access Objects (Classes)

- TopicDaoJS
  - getTopics(): Topic
  - getTopicsIdst(): Integer List
  - addTopicDB(idFather, name)
  - isMySon(idFather, idSon): Boolean
  - idsExist(Ids) : Boolean
  - moveTopic(idToMove, idWhereToMove)
- UserDaoJS
  - findUser(username) : Boolean
  - match(username, password): User

RIA - Ajax Calls

Send Form Data / Simple Ajax Call

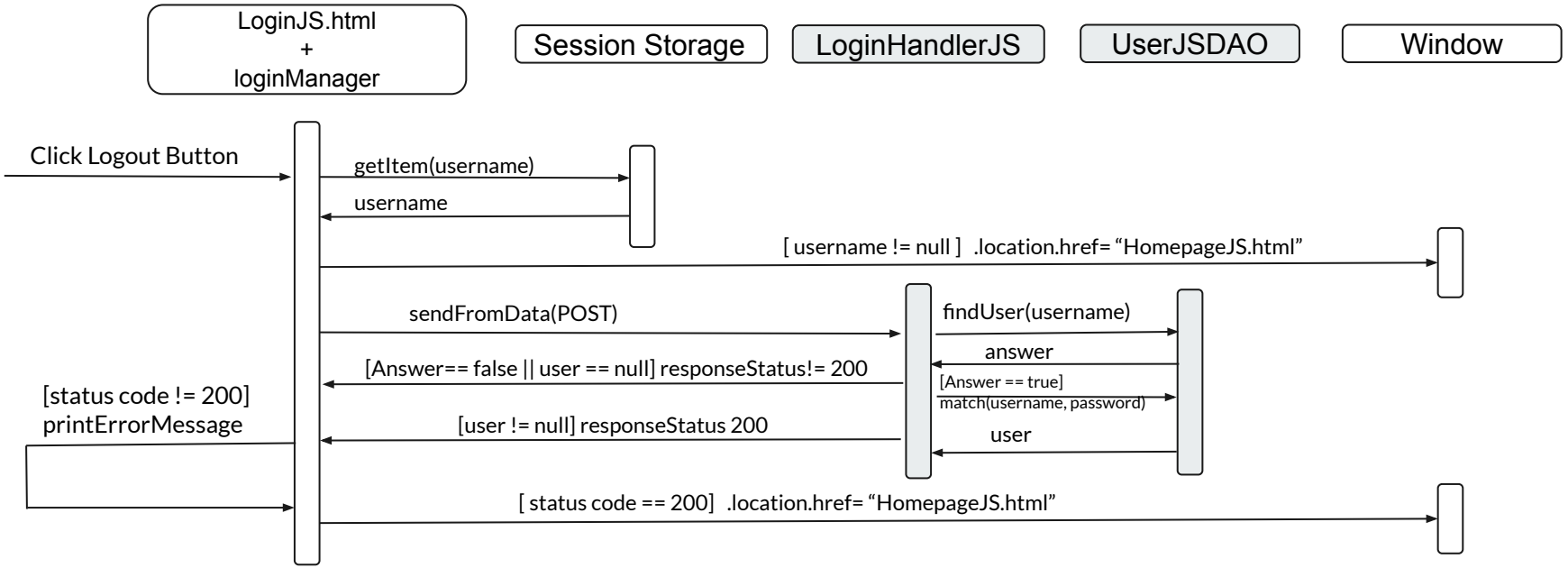
```
function sendFormData (method, url, formElement, cback, reset = true) {
  var req = new XMLHttpRequest(); // visible by closure
  req.onreadystatechange = function() {
    cback(req)
  }; // closure
  req.open(method, url);
  if (formElement == null) {
    req.send();
  } else {
    const formData = new FormData(formElement);
    req.send(formData);
  }
  if (formElement !== null && reset === true) {
    formElement.reset();
  }
}
```

Send Json Object

```
function sendJsonObject(method, url, obj, cback){
  var req = new XMLHttpRequest();   // new HttpRequest instance
  req.onreadystatechange=function (){
    cback(req);
  };
  req.open("POST", url);
  if(obj==null){
    req.send();
  }
  else{
    const json= JSON.stringify(obj);
    req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
    req.send(json)
  }
}
```

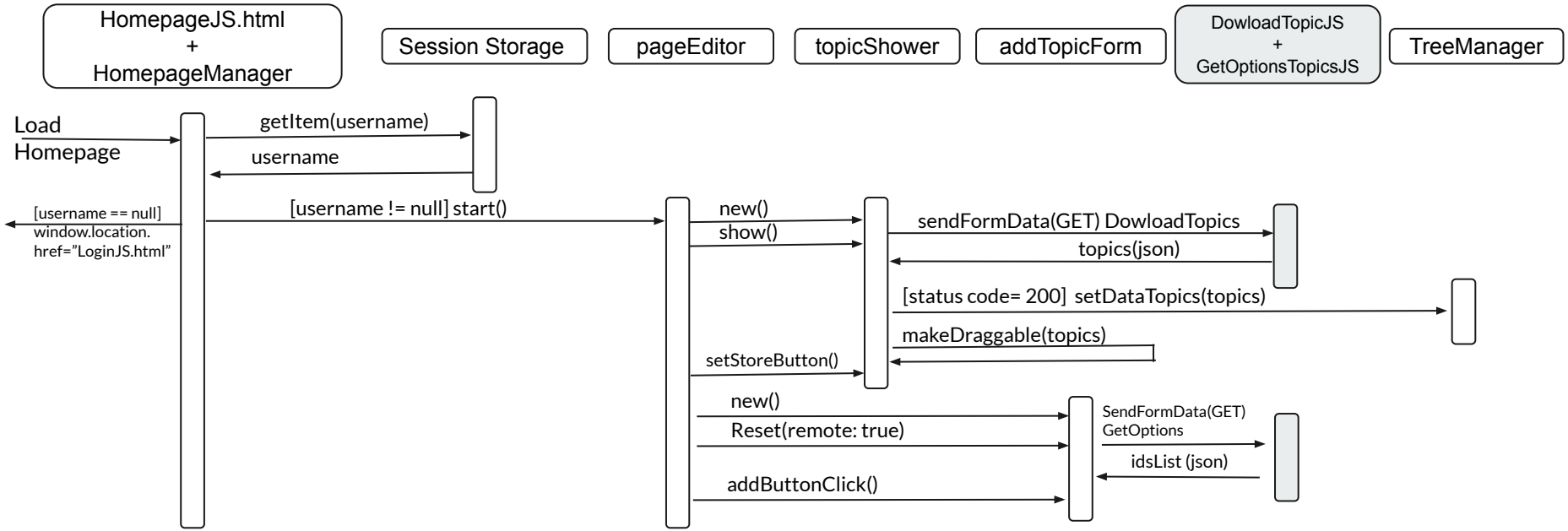
Server side Client side

RIA - Event: Login



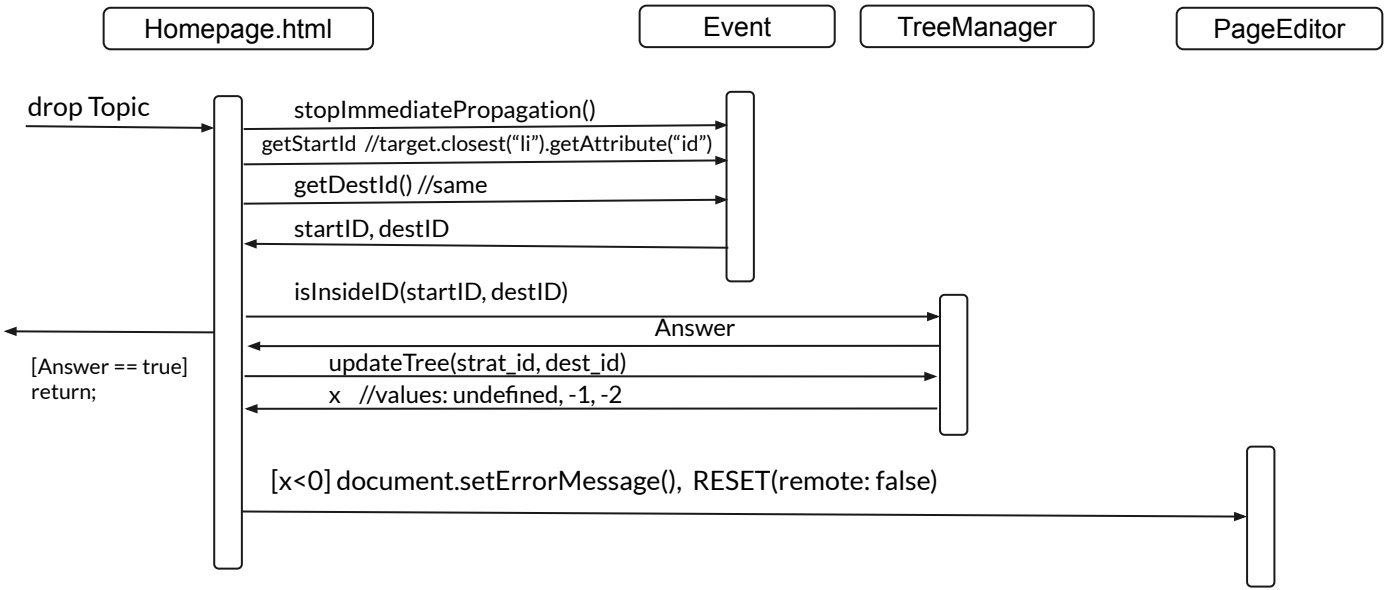
Server side Client side

RIA - Event: Load Home



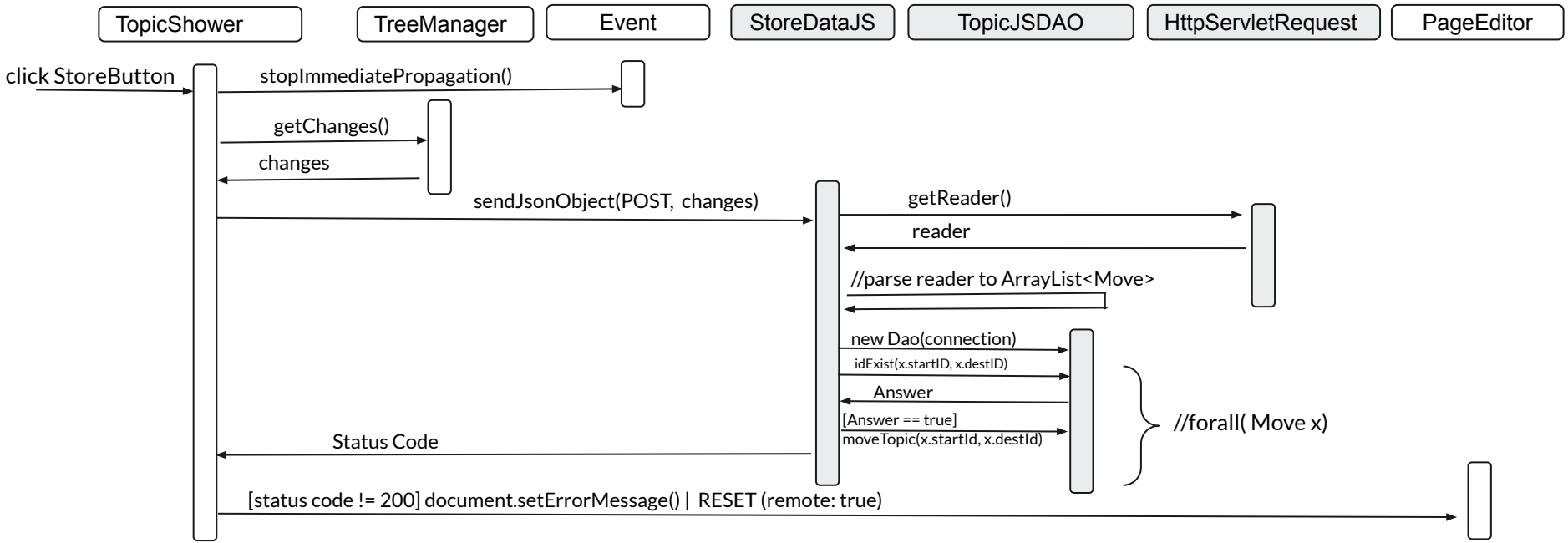
Server side Client side

RIA - Event: Drag & Drop



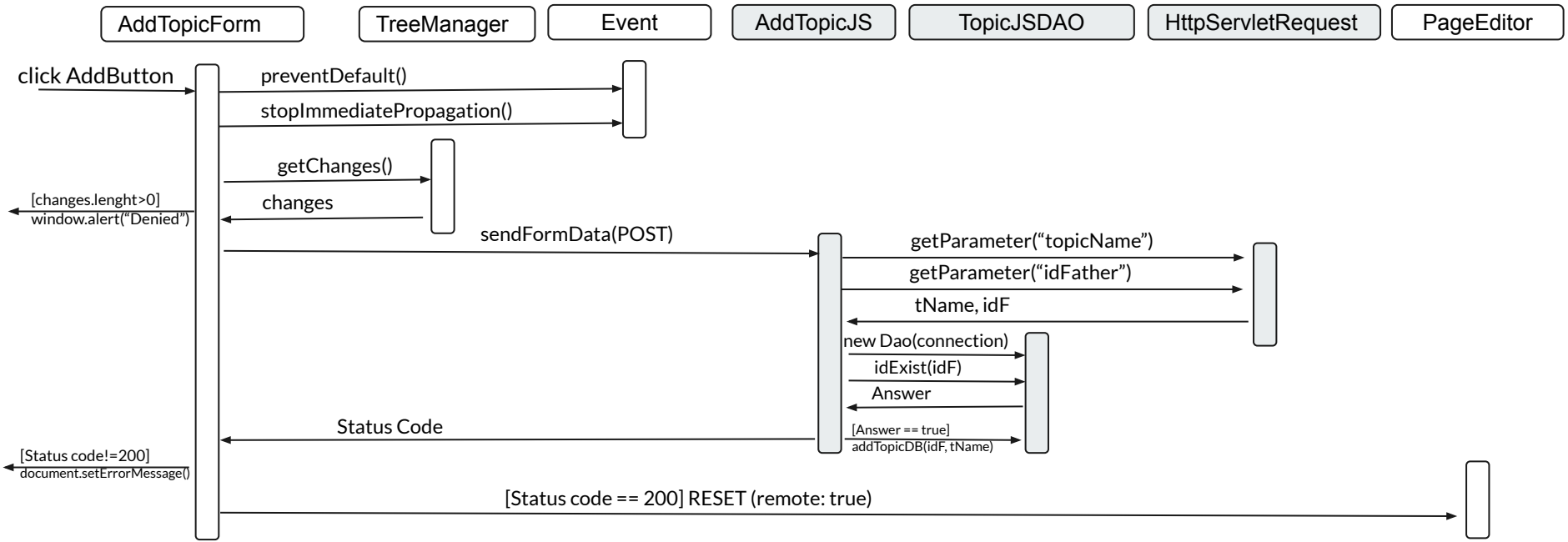
Server side Client side

RIA - Event: Store Data



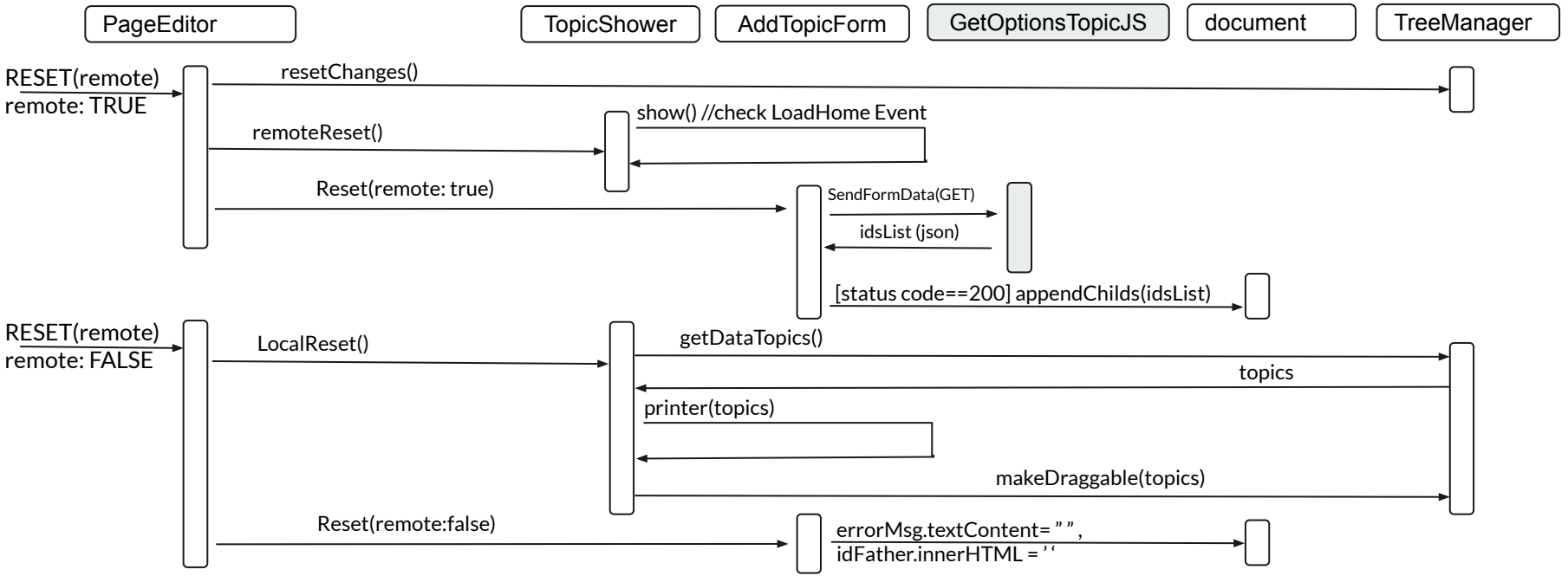
Server side Client side

RIA - Event: Add Topic



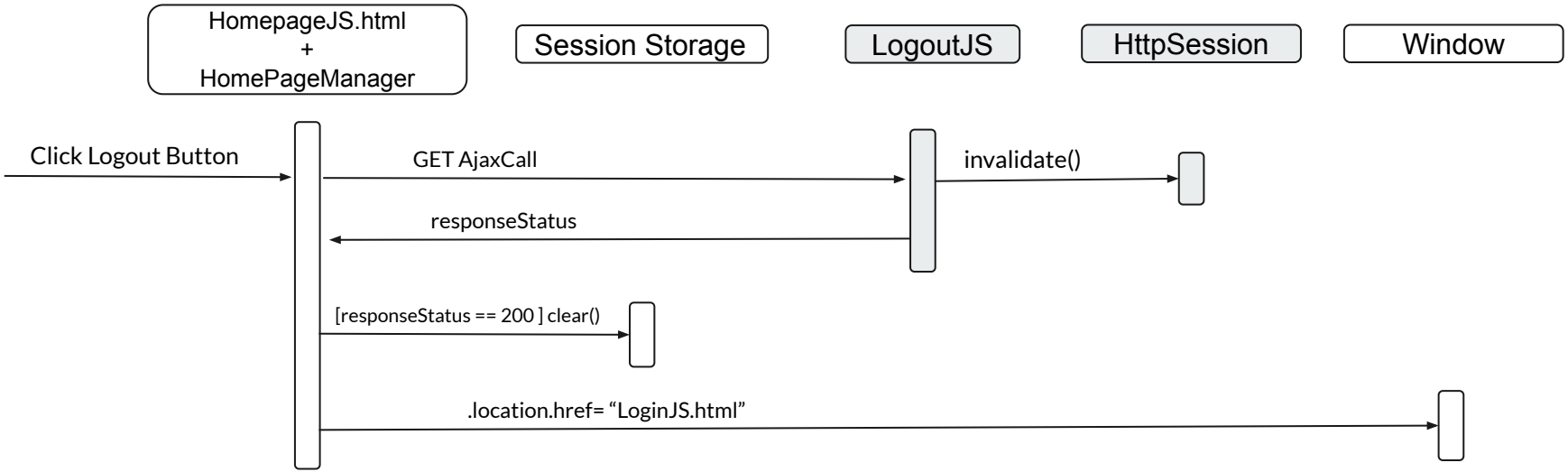
Server side Client side

RIA - Event: Reset



Server side Client side

RIA - Event: Logout



RIA - Filters Mapping

Filter Name	Servlets
UserChecker	DownloadTopicsJS, GetOptionsTopicJS, AddTopicJS, StoreDataJS, LogoutJS