

Catalogazione di immagini

Tecnologie informatiche per il web

Esercizio 3 – Versione RIA

Eliahu Itamar Cohen – 10704321

Lorenzo Fonnesu - 10740840

Analisi dei dati (comune alla versione HTML)

Un'applicazione permette all'utente (ad esempio il responsabile dei servizi ambientali di una regione) di gestire una collezione di immagini satellitari e una tassonomia di classificazione utile per etichettare immagini allo scopo di consentire la ricerca per 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. 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 "Amianto in tubi", figlia della categoria "9111 Amianto" assume il codice 91113). Dopo la creazione di una categoria, la pagina HOME mostra l'albero aggiornato. Per velocizzare la costruzione della tassonomia l'utente può copiare un intero sottoalbero in una data posizione: per fare ciò clicca sul link "copia" associato alla categoria radice del sottoalbero da copiare. A seguito di tale azione l'applicazione mostra, sempre nella HOME page, l'albero con evidenziato il sottoalbero da copiare: tutte le altre categorie hanno un link "copia qui".

La selezione di un link "copia qui" comporta l'inserimento di una copia del sottoalbero come ultimo figlio della categoria destinazione. Le modifiche effettuate da un utente e salvate nella base di dati diventano visibili agli altri utenti.

Per semplicità si ipotizzi che per ogni categoria il numero massimo di sottocategorie sia 9, numerate da 1 a 9. In questo caso l'operazione di copia deve controllare che lo spostamento non determini un numero di sottocategorie superiore a 9. Si preveda anche un link "copia qui" non associato a un nodo della tassonomia che permette di copiare un sottoalbero al primo livello della tassonomia (se non esistono già 9 nodi al primo livello della tassonomia) .

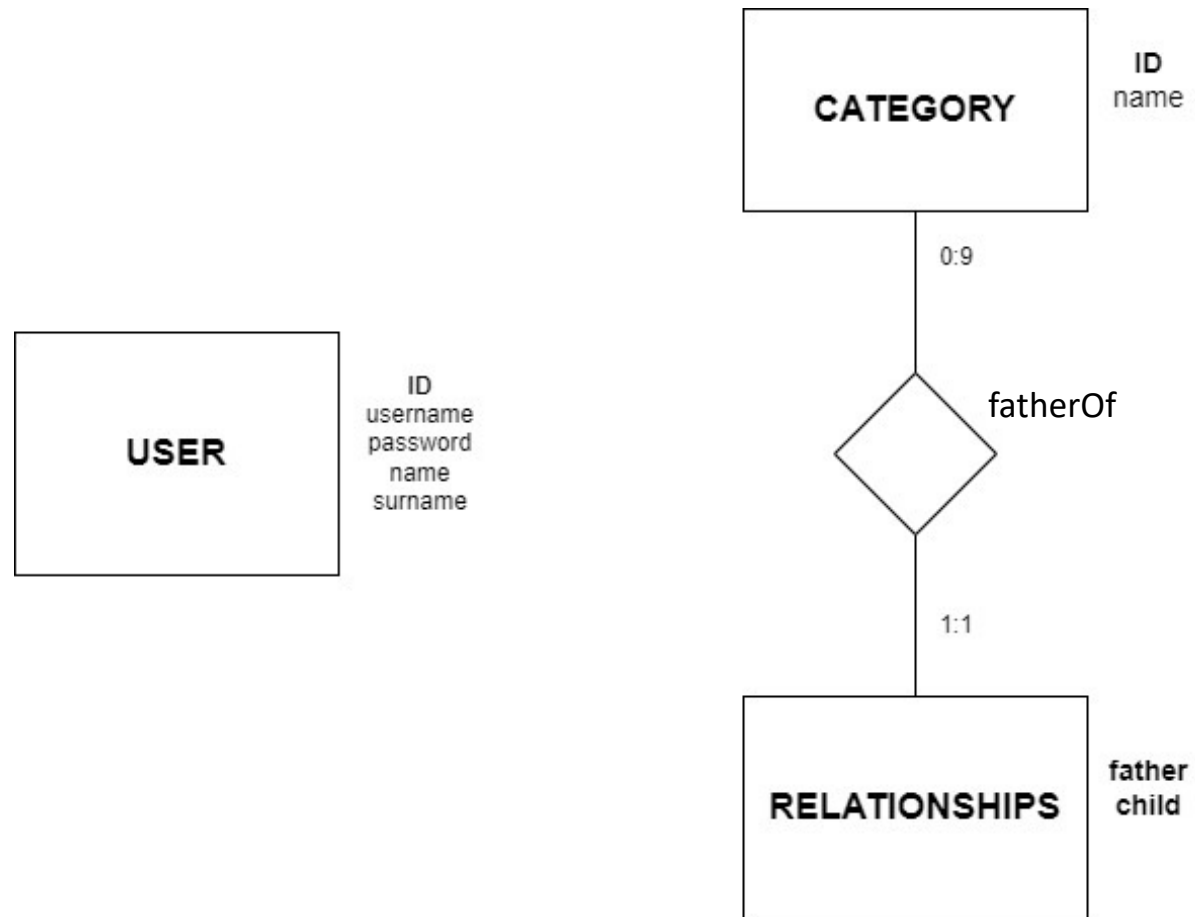
Entità, attributi, relazioni

Specifiche RIA

Si realizzi un'applicazione client server web che estende e/o 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 copia di un sottoalbero è realizzata mediante drag & drop. A seguito del drop della radice del sottoalbero da copiare compare una finestra di dialogo con cui l'utente può confermare o cancellare la copia. La conferma produce l'aggiornamento solo a lato client dell'albero. La cancellazione riconduce allo stato precedente al drag & drop. A seguito della conferma compare un bottone SALVA che consente il salvataggio a lato server della tassonomia modificata.
- L'utente può cliccare sul nome di una categoria. A seguito di tale evento compare al posto del nome un campo di input contenente la stringa del nome modificabile. L'evento di perdita del focus del campo di input produce il salvataggio nel database del nome modificato della categoria.

Database design



Local database schema

```
CREATE TABLE `category` (  
    `id` varchar(100) NOT  
    NULL,  
    `name` varchar(45) NOT  
    NULL,  
    PRIMARY KEY (`id`),  
    KEY `existence` (`id`)  
)
```

```
CREATE TABLE  
`relationships` (  
    `father` varchar(100)  
    NOT NULL,  
    `child` varchar(100) NOT  
    NULL,  
    KEY `father_id_idx`  
    (`father`),  
    KEY `child_id_idx`  
    (`child`),  
    CONSTRAINT `child_id`  
    FOREIGN KEY (`child`)  
    REFERENCES `category`  
    (`id`) ON DELETE CASCADE  
    ON UPDATE CASCADE,  
    CONSTRAINT `father_id`  
    FOREIGN KEY (`father`)  
    REFERENCES `category`  
    (`id`) ON DELETE CASCADE  
    ON UPDATE CASCADE  
)
```

```
CREATE TABLE `user` (  
    `id` int NOT NULL  
    AUTO_INCREMENT,  
    `username`  
    varchar(45) NOT NULL,  
    `password`  
    varchar(45) NOT NULL,  
    `name` varchar(45)  
    NOT NULL,  
    `surname` varchar(45)  
    NOT NULL,  
    PRIMARY KEY (`id`)  
)
```

Analisi dei requisiti

Un'applicazione permette all'utente (ad esempio il responsabile dei servizi ambientali di una regione) di gestire una collezione di immagini satellitari e una tassonomia di classificazione utile per etichettare immagini allo scopo di consentire la ricerca per 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. 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 "Amianto in tubi", figlia della categoria "9111 Amianto" assume il codice 91113). Dopo la creazione di una categoria, la pagina HOME mostra l'albero aggiornato. Per velocizzare la costruzione della tassonomia l'utente può copiare un intero sottoalbero in una data posizione: per fare ciò clicca sul link "copia" associato alla categoria radice del sottoalbero da copiare. A seguito di tale azione l'applicazione mostra, sempre nella HOME page, l'albero con evidenziato il sottoalbero da copiare: tutte le altre categorie hanno un link "copia qui".

La selezione di un link "copia qui" comporta l'inserimento di una copia del sottoalbero come ultimo figlio della categoria destinazione. Le modifiche effettuate da un utente e salvate nella base di dati diventano visibili agli altri utenti.

Per semplicità si ipotizzi che per ogni categoria il numero massimo di sottocategorie sia 9, numerate da 1 a 9. In questo caso l'operazione di copia deve controllare che lo spostamento non determini un numero di sottocategorie superiore a 9. Si preveda anche un link "copia qui" non associato a un nodo della tassonomia che permette di copiare un sottoalbero al primo livello della tassonomia (se non esistono già 9 nodi al primo livello della tassonomia)

Pages (views), view components, events, actions

Specifiche RIA

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

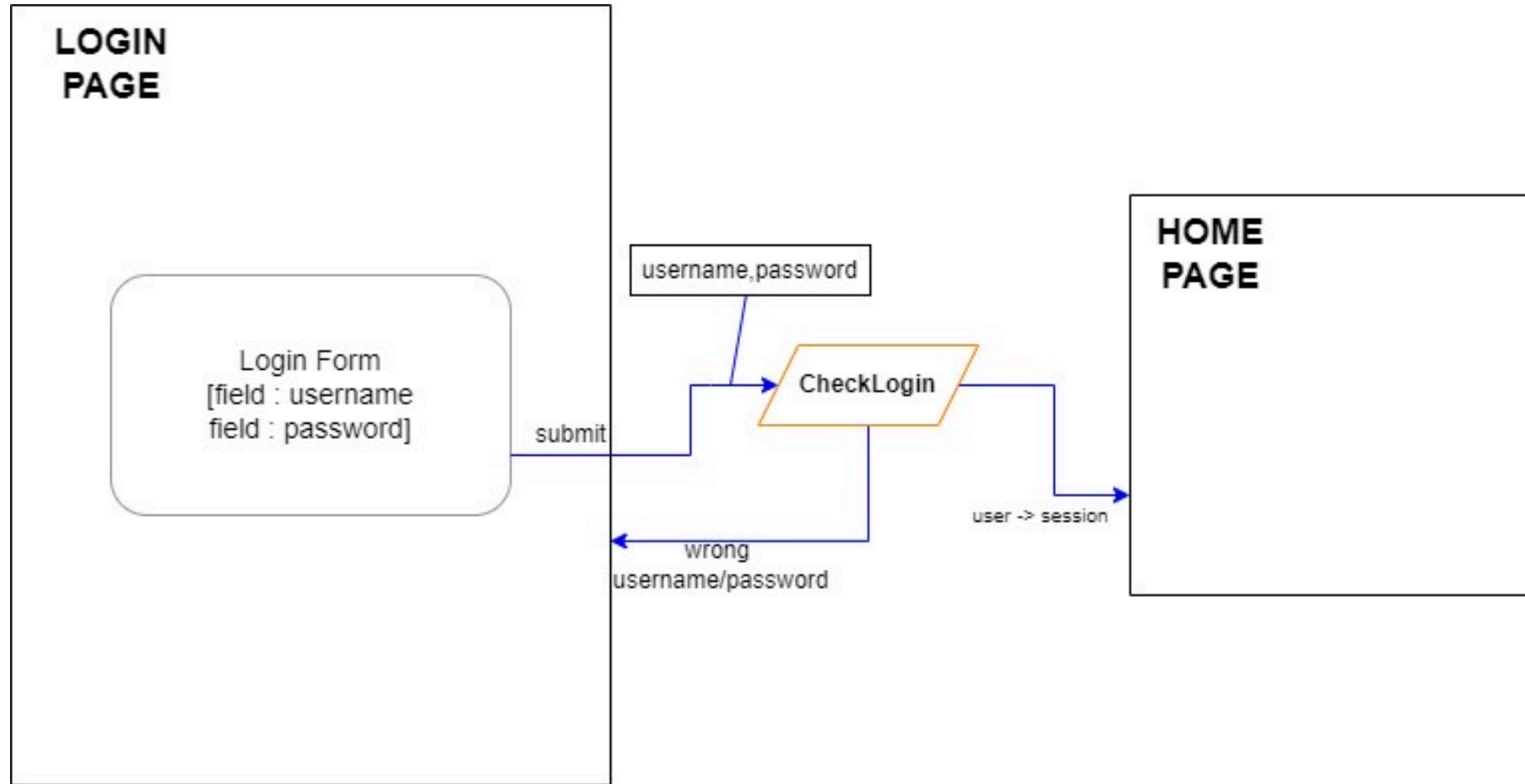
- Dopo il login dell'utente, l'intera applicazione è realizzata con un'unica pagina.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- La funzione di copia di un sottoalbero è realizzata mediante drag & drop. A seguito del drop della radice del sottoalbero da copiare compare una finestra di dialogo con cui l'utente può confermare o cancellare la copia. La conferma produce l'aggiornamento solo a lato client dell'albero. La cancellazione riconduce allo stato precedente al drag & drop. A seguito della conferma compare un bottone SALVA che consente il salvataggio a lato server della tassonomia modificata.
- L'utente può cliccare sul nome di una categoria. A seguito di tale evento compare al posto del nome un campo di input contenente la stringa del nome modificabile. L'evento di perdita del focus del campo di input produce il salvataggio nel database del nome modificato della categoria.

Completamento delle specifiche

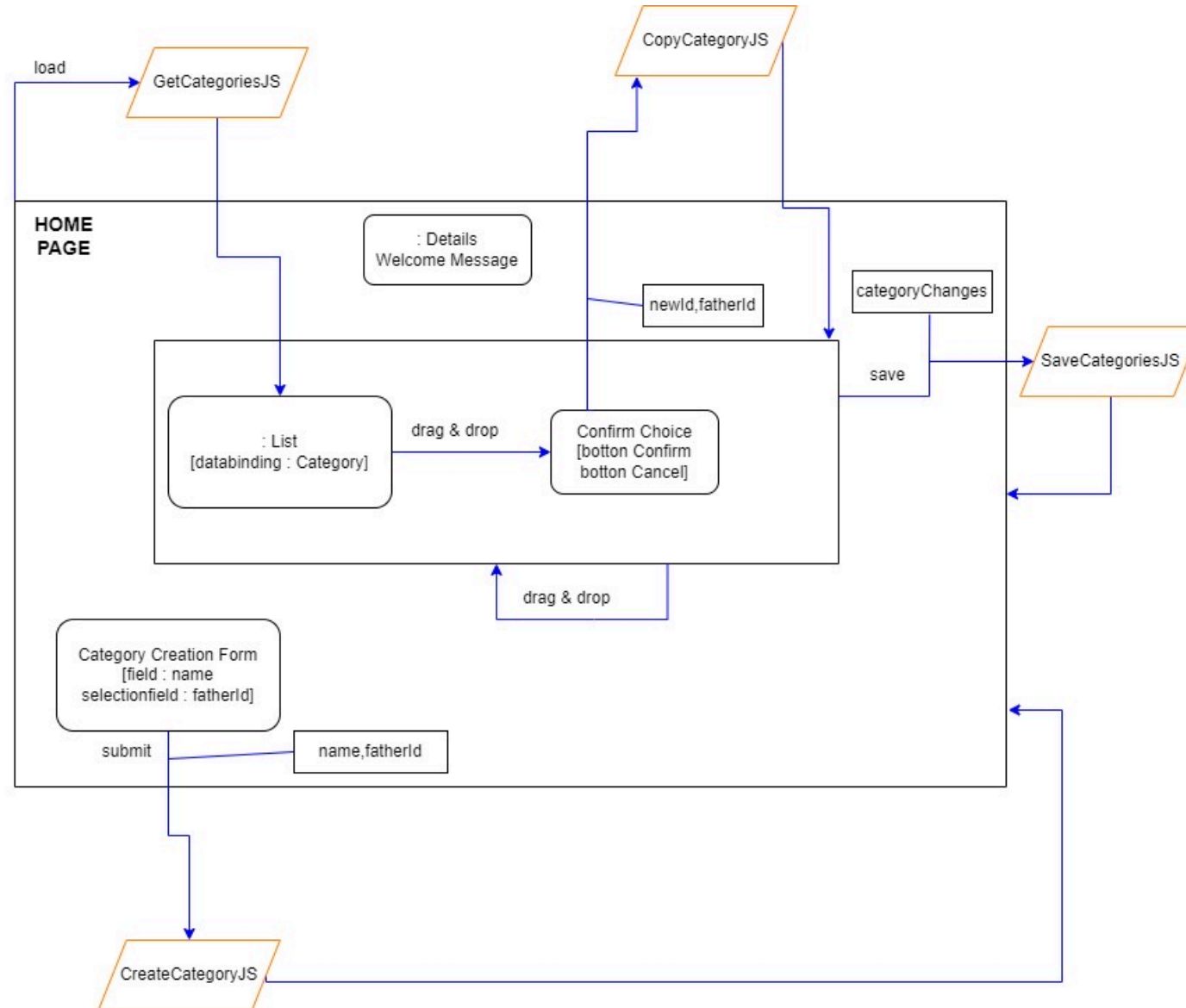
- La **pagina iniziale** contiene la **form di login**
- Nella form del login sono obbligatori sia l'username che la password. Nella form di creazione di una nuova categoria sono obbligatori sia il nome che la categoria padre.
- Le categorie con id da '1' a '9' non hanno padre.
- Una categoria non può essere copiata all'interno di se stessa o di una sua sottocategoria.
- Se la categoria in cui viene copiato l'albero ha già raggiunto il massimo numero di sottocategorie, **la copia viene annullata quando viene effettuato il salvataggio o il refresh della pagina.**
- Non è possibile eseguire il drop di un albero senza scegliere una categoria padre.
- Quando l'utente aggiorna la pagina, le modifiche non salvate vengono annullate.
- Non è possibile copiare un albero dentro il suo padre diretto.
- Nell'analisi dei requisiti non è stata considerata la sezione relativa ai link, poiché viene sostituita dal drag&drop.

Pages (views), view components, events, actions

Application design



Application design



Client-side: view components and events

■ Index:

- Login form: controllo delle credenziali

■ Home:

- AlertContainer: modale contenente un messaggio di errore
- ConfirmCopy:
 - Show(): mostra la finestra di conferma in seguito all'evento di drop
 - Close(): chiude la finestra di conferma
 - Confirm(): chiude la finestra di conferma e aggiorna l'albero in locale
- SaveCopy:
 - Show(): mostra il bottone che permette il salvataggio delle modifiche nel server
 - Close(): nasconde il bottone di salvataggio
- PersonalMessage:
 - Show(): mostra il messaggio di benvenuto

• CategoriesList:

- Show(): richiede i dati dal server e chiama print
- Print(): stampa la lista di categorie ricevuta dal server e assegna a ciascuna le funzioni relative alla gestione del drag&drop e della modifica del nome
- RegisterEvents(): gestisce la comparsa del bottone 'Salva' e gli assegna l'event Listener corrispondente

• CreationForm:

- Enable(): abilita il form di creazione delle categorie
- Disable(): disabilita il form
- RegisterEvents(): assegna al bottone 'Crea' l'event Listener corrispondente
- Show(): richiede i dati dal server e chiama update
- Update(): aggiunge al padre selezionato la nuova categoria

➤ In ConfirmCopy, SaveCopy, CategoriesList e CreationForm è presente anche la funzione 'reset()', la quale si occupa di riportare i vari elementi alla condizione iniziale.

Server side: DAO & model objects

■ Model objects (Beans):

- User
- Category
- CategoryChanges

■ Data Access Objects (Classes):

- UserDao:
 - CheckCategory(username, password)
- CategoryDAO:
 - checkCategory(id)
 - addSubparts(category, ID)
 - createCategory(name, id)
 - getNewID(fatherID)
 - paste(fatherId, fatherNewId)
 - findAllCategories()
 - updateName(categoryId, newName)

■ Controllers (Servlets):

- CheckLoginJS
- CopyCategoryJS
- CreateCategoryJS
- GetCategoriesJS
- ModifyNameJS
- SaveCategoriesJS

■ Views (templates):

- Index
- Home

Server side: additional packages

■ Utils

- ConnectionHandler: classe che gestisce la creazione e la chiusura della connessione con il database

■ Filters

- LoginCheckers: ogni volta che viene chiamato un metodo di una servlet viene controllata la validità della sessione e, in caso di invalidità, l'utente viene reindirizzato alla pagina di login

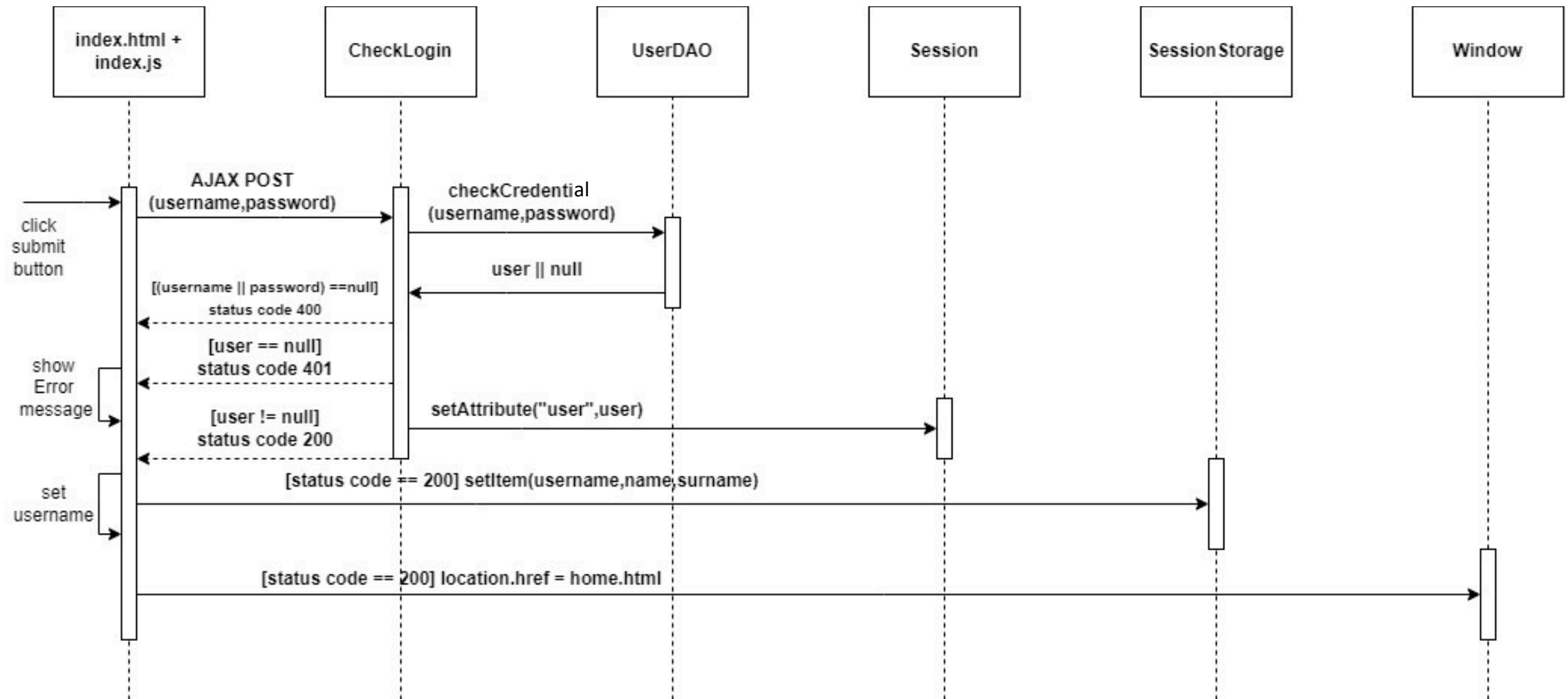
Eventi e azioni

CLIENT SIDE		SERVER SIDE	
EVENTO	AZIONE	EVENTO	AZIONE
Index -> login form -> submit	Controllo dati	POST username, password	Controllo credenziali
Home -> load	Aggiorna la view con lista delle categorie	GET	Estrazione delle categorie dal database
Home -> lista categorie -> Drag&Drop -> Conferma	Controllo dati + Incolla l'albero selezionato nella categoria di destinazione e mostra il tasto Salva	GET fatherId	Estrazione delle categorie copiate
Home -> lista categorie -> Drag&Drop -> Cancella	Annulla la modifica e mostra la lista invariata	-	-
Home -> Salva	-	POST categoryUpdateArray	Aggiorna la lista delle categorie
Home -> form di creazione	Controllo dati	POST name, fatherID	Aggiunge la nuova categoria come figlio del padre selezionato
Home -> lista categorie -> click	-	POST id, newName	Aggiornamento del nome della categoria

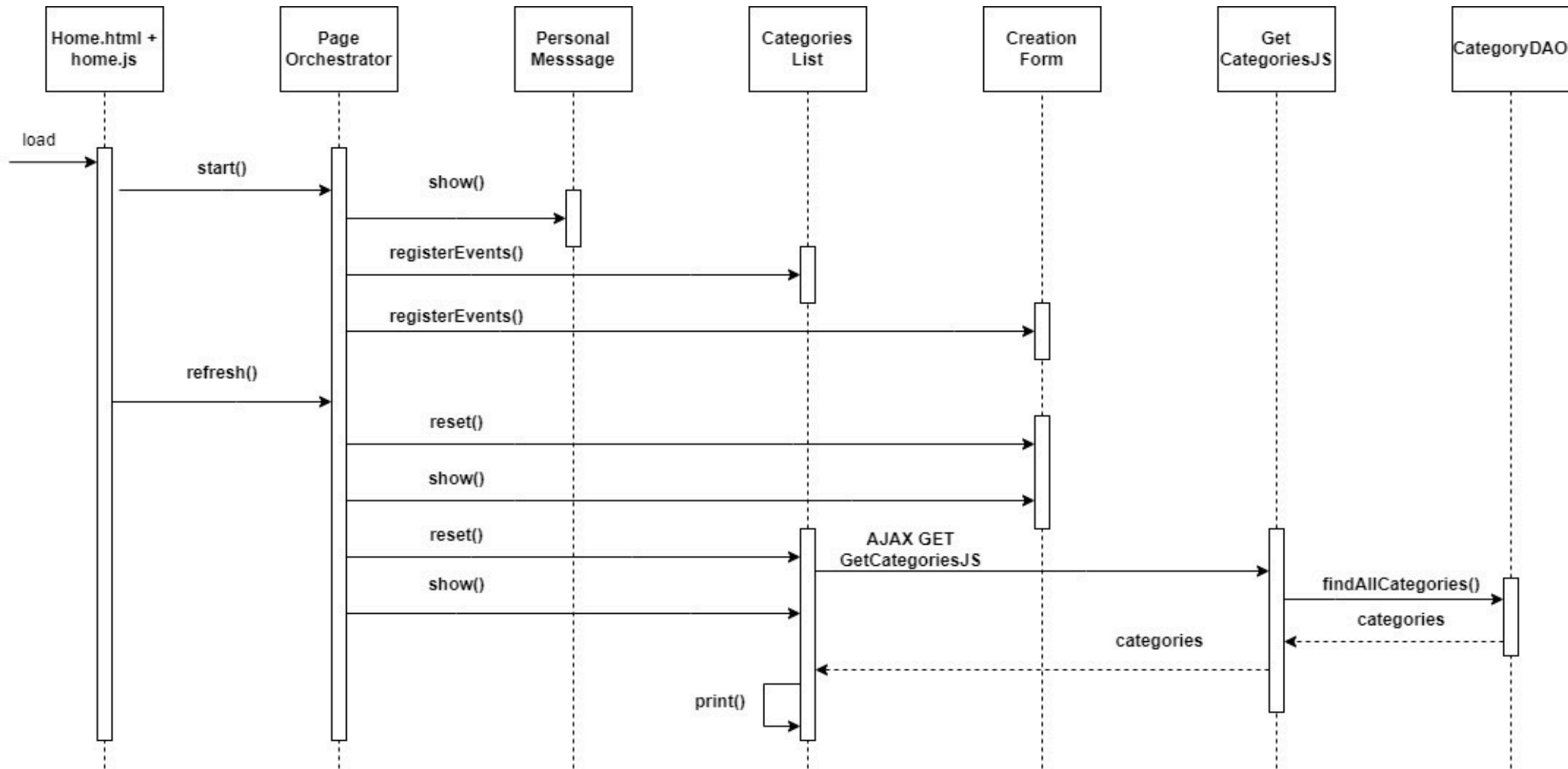
Controllori e handler

CLIENT SIDE		SERVER SIDE	
EVENTO	AZIONE	EVENTO	AZIONE
Index -> login form -> submit	Function makeCall	POST username, password	CheckLoginJS(Servlet)
Home -> load	Function PageOrchestrator + Function makeCall	GET	GetCategoriesJS(Servlet)
Home -> lista categorie -> Drag&Drop -> Conferma	Function makeCall	GET fatherId	CopyCategoriesJS(Servlet)
Home -> lista categorie -> Drag&Drop -> Cancella	-	-	-
Home -> Salva	Function makeCall	POST categoryUpdateArray	SaveCategoryJS(Servlet)
Home -> form di creazione	Function makeCall	POST name, fatherID	CreateCatgeoryJS(Servlet)
Home -> lista categorie -> click	Function makeCall	POST id, newName	ModifyNameJS(Servlet)

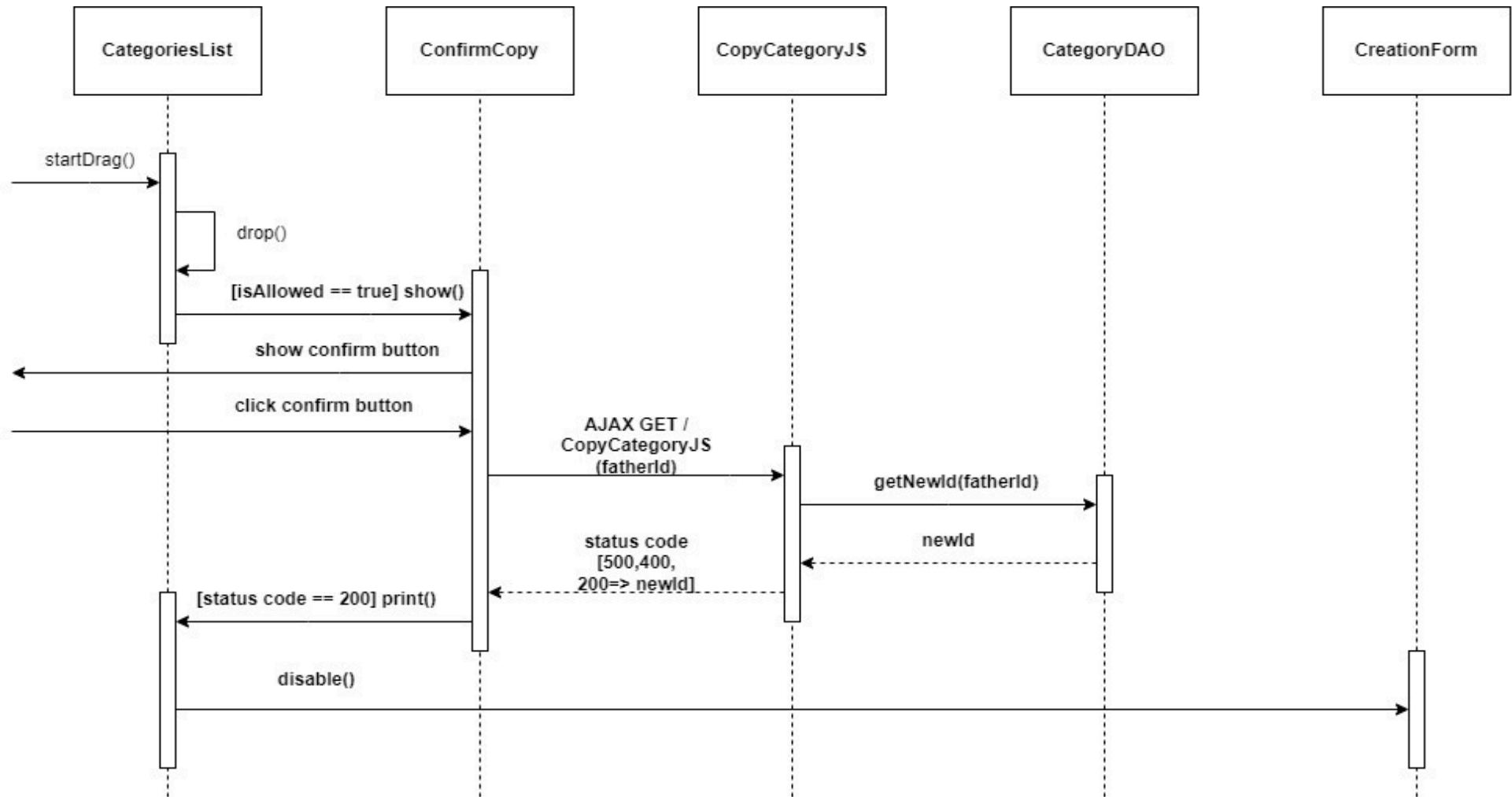
Event: login



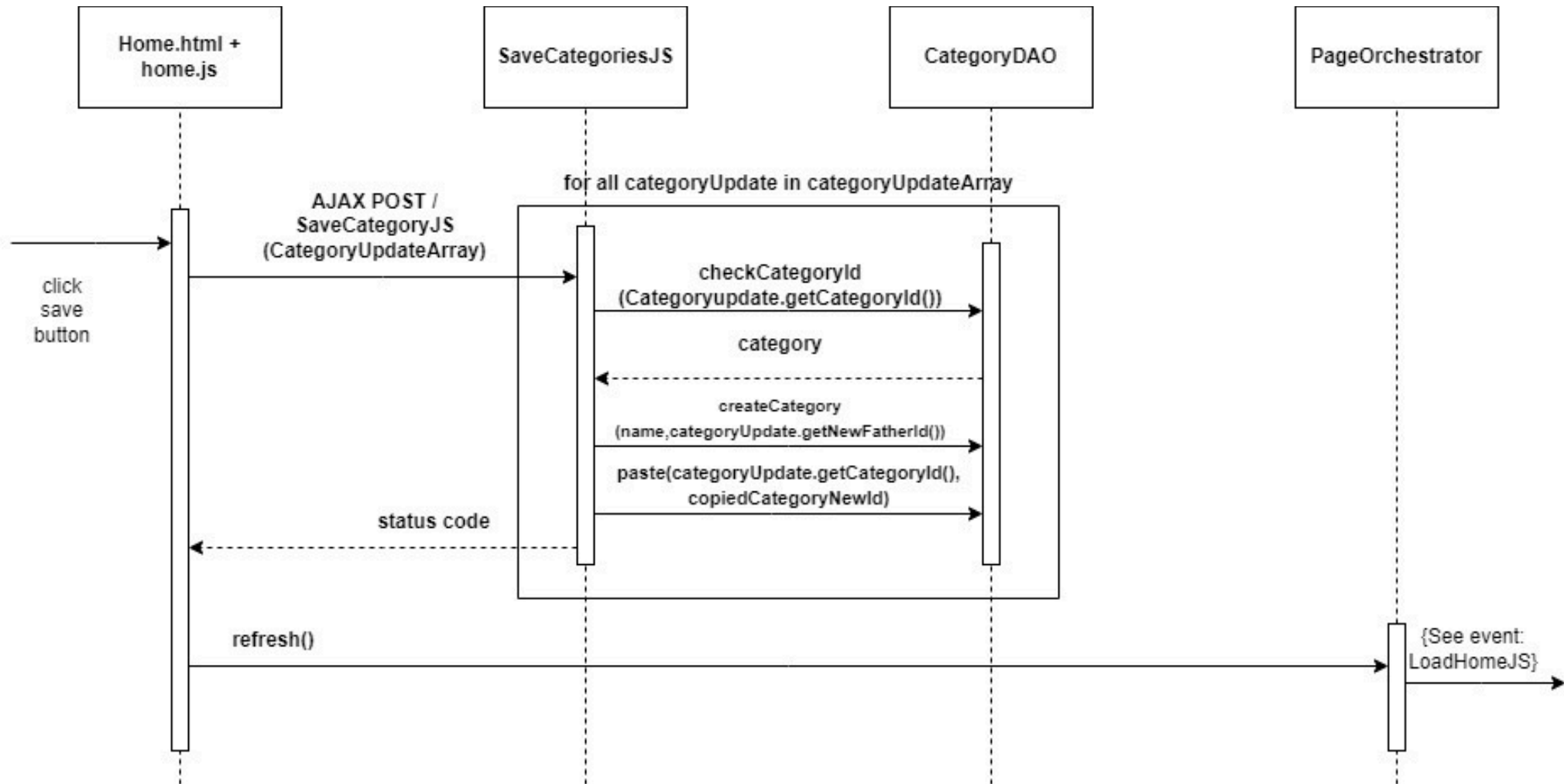
Event: home page loading



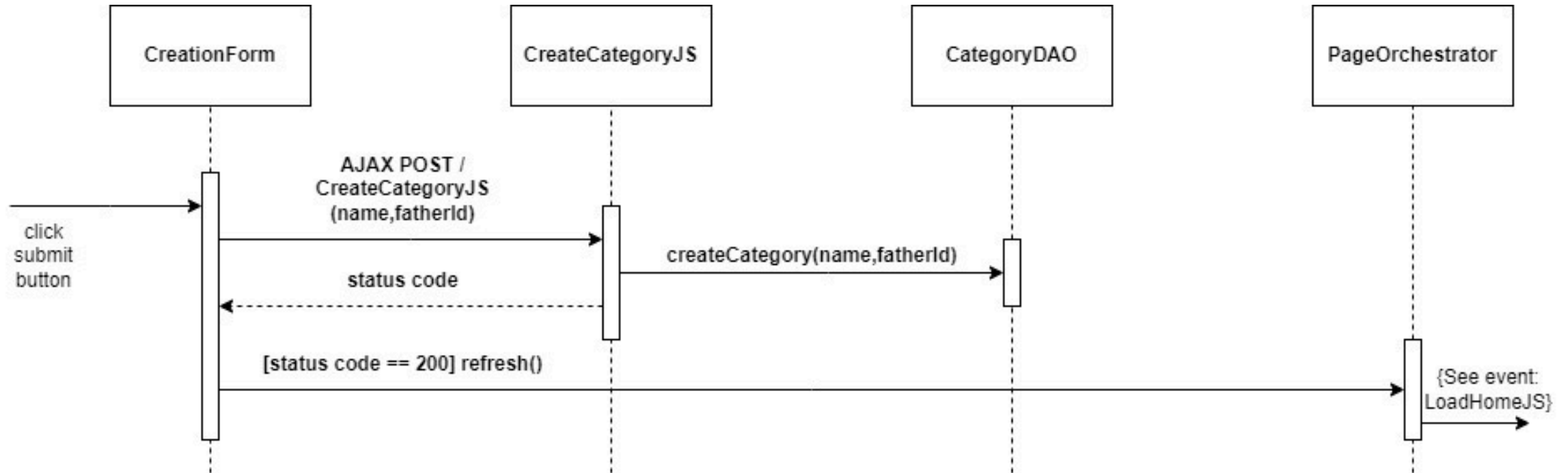
Event: copy category



Event: save category



Event: create category



Event: modify name

