Esercizio 4: trasferimento denaro

Versione pure HTML

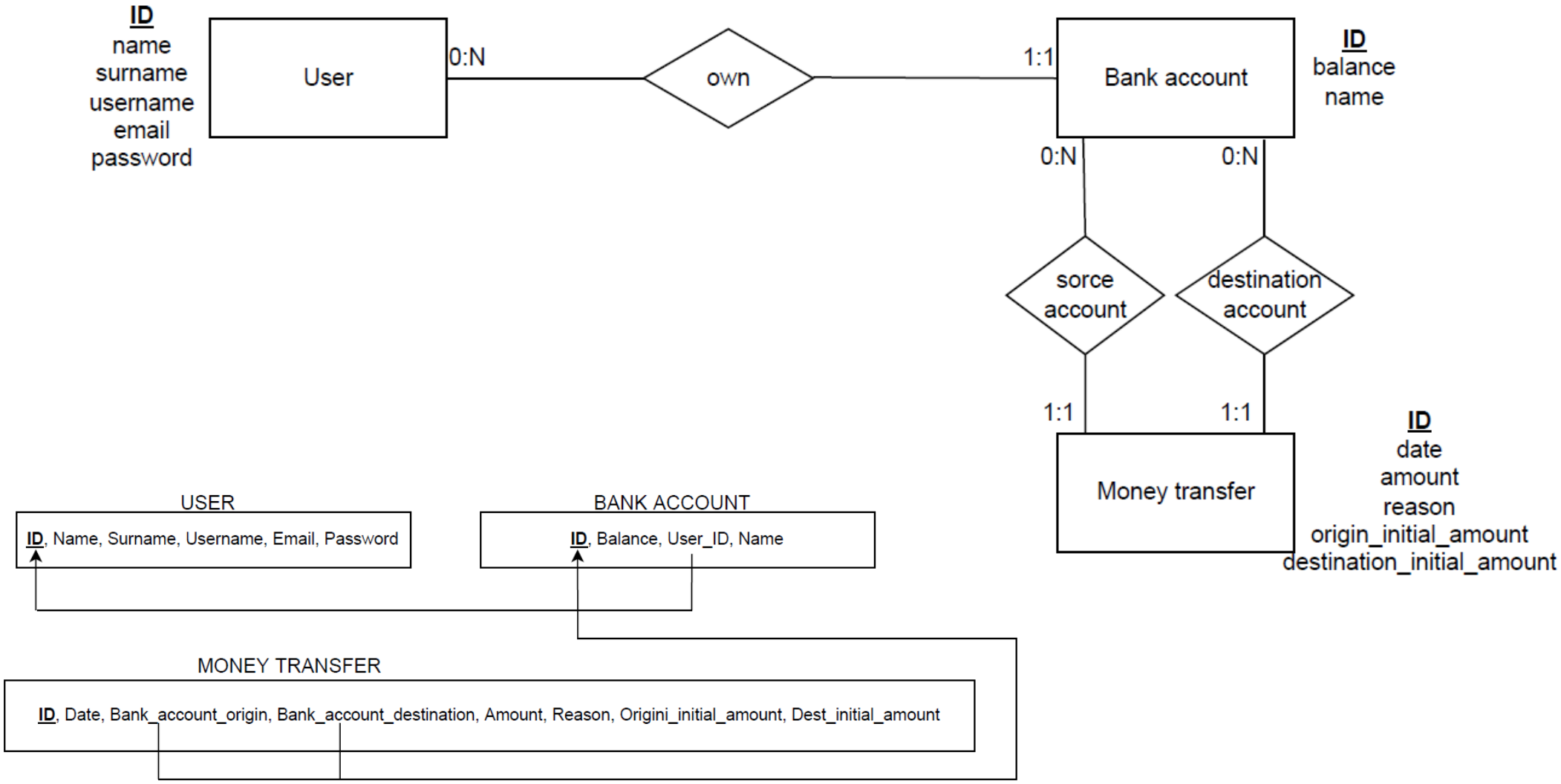
Un'applicazione web consente la gestione di trasferimenti di denaro online da un conto a un altro. L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password". La registrazione controlla l'unicità dello username. Un utente ha un nome, un cognome, uno username e uno o più conti correnti. Un conto ha un codice, un saldo, e i trasferimenti fatti (in uscita) e ricevuti (in ingresso) dal conto. Un trasferimento ha una data, un importo, un conto di origine e un conto di destinazione. Quando l'utente accede all'applicazione appare una pagina LOGIN per la verifica delle credenziali. In seguito all'autenticazione dell'utente appare l'HOME page che mostra l'elenco dei suoi conti. Quando l'utente seleziona un conto, appare una pagina STATO DEL CONTO che mostra i dettagli del conto e la lista dei movimenti in entrata e in uscita, ordinati per data discendente. La pagina contiene anche una form per ordinare un trasferimento. La form contiene i campi: codice utente destinatario, codice conto destinatario, causale e importo. All'invio della form con il bottone INVIA, l'applicazione controlla che il conto di destinazione appartenga all'utente specificato e che il conto origine abbia un saldo superiore o uguale all'importo del trasferimento. In caso di mancanza di anche solo una condizione, l'applicazione mostra una pagina con un avviso di fallimento che spiega il motivo del mancato trasferimento. Nel caso in cui entrambe le condizioni siano soddisfatte, l'applicazione deduce l'importo dal conto di origine, aggiunge l'importo al conto di destinazione e mostra una pagina CONFERMA TRASFERIMENTO che presenta i dati dell'importo trasferito e i dati del conto di origine e di destinazione con i rispettivi saldi precedenti al trasferimento e aggiornati dopo il trasferimento. L'applicazione deve garantire l'atomicità del trasferimento: ogni volta che il conto di destinazione viene addebitato, il conto di origine deve essere accreditato. Ogni pagina contiene un collegamento per tornare alla pagina precedente. L'applicazione consente il logout dell'utente.

Analisi dati per database

Un’applicazione web consente la gestione di trasferimenti di denaro online da un conto a un altro. 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. Un utente ha un nome, un cognome, uno username e uno o più conti correnti. Un conto ha un codice, un saldo, e i trasferimenti fatti (in uscita) e ricevuti (in ingresso) dal conto. Un trasferimento ha una data, un importo, un conto di origine e un conto di destinazione. Quando l’utente accede all’applicazione appare una pagina LOGIN per la verifica delle credenziali. In seguito all’autenticazione dell’utente appare l’HOME page che mostra l’elenco dei suoi conti. Quando l’utente seleziona un conto, appare una pagina STATO DEL CONTO che mostra i dettagli del conto e la lista dei movimenti in entrata e in uscita, ordinati per data discendente. La pagina contiene anche una form per ordinare un trasferimento. La form contiene i campi: codice utente destinatario, codice conto destinatario, causale e importo. All’invio della form con il bottone INVIA, l’applicazione controlla che il conto di destinazione appartenga all’utente specificato e che il conto origine abbia un saldo superiore o uguale all’importo del trasferimento. In caso di mancanza di anche solo una condizione, l’applicazione mostra una pagina con un avviso di fallimento che spiega il motivo del mancato trasferimento. Nel caso in cui entrambe le condizioni siano soddisfatte, l’applicazione deduce l’importo dal conto di origine, aggiunge l’importo al conto di destinazione e mostra una pagina CONFERMA TRASFERIMENTO che presenta i dati dell’importo trasferito e i dati del conto di origine e di destinazione con i rispettivi saldi precedenti al trasferimento e aggiornati dopo il trasferimento. L’applicazione deve garantire l’atomicità del trasferimento: ogni volta che il conto di destinazione viene addebitato, il conto di origine deve essere accreditato. Ogni pagina contiene un collegamento per tornare alla pagina precedente. L’applicazione consente il logout dell’utente.

- Entities, attributes, relationship

Database design



Database schema (1/2)

```
CREATE TABLE 'User' (
        'id' int AUTO_INCREMENT primary key,
        'name' varchar(45) not null,
        'surname' varchar(45) not null,
        'username' varchar(45) unique not null,
        'e-mail' varchar(60) unique not null,
        'password' varchar(45) not null
)
```

```
CREATE TABLE 'BankAccount' (
        'id' int(11)  AUTO_INCREMENT primary key,
        'balance'  decimal(10, 2) not null default '0.00',
        'user_id' int not null,
        'name' varchar(45) not null,
        costraint 'userAccount' foreign key ('user_id')
                                references 'User'('id') on update cascade
                                                         on delete cascade
)
```

Database schema (2/2)

```
CREATE TABLE 'MoneyTransfer' (
        'id' int(11) AUTO_INCREMENT primary key,
        'date' timestamp not null default CURRENT_TIMESTAMP,
        'bankAccountOrigin' int not null,
        'bankAccountDestination' int not null,
        'amount' decimal(10, 2) not null,
        'reason' varchar(60) not null,
        'origin_initial_amount' decimal(10, 2) not null,
        'destination_initial_amount' decimal(10, 2) not null,
        costraint 'sourceAccount' foreign key ('bankAccountOrigin')
                                        references 'BankAccount'('id') on update cascade
                                                                        on delete no action,
        costraint 'destinationAccount' foreign key ('bankAccountDestination')
                                        references 'BankAccount'('id') on update cascade
                                                                        on delete no action
)
```

Analisi requisiti applicazione

Un'applicazione web consente la gestione di trasferimenti di denaro online da un conto a un altro. 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. Un utente ha un nome, un cognome, uno username e uno o più conti correnti. Un conto ha un codice, un saldo, e i trasferimenti fatti (in uscita) e ricevuti (in ingresso) dal conto. Un trasferimento ha una data, un importo, un conto di origine e un conto di destinazione. Quando l'utente accede all'applicazione appare una pagina LOGIN per la verifica delle credenziali. In seguito all'autenticazione dell'utente appare l'HOME page che mostra l'elenco dei suoi conti. Quando l'utente seleziona un conto, appare una pagina STATO DEL CONTO che mostra i dettagli del conto e la lista dei movimenti in entrata e in uscita, ordinati per data discendente. La pagina contiene anche una form per ordinare un trasferimento. La form contiene i campi: codice utente destinatario, codice conto destinatario, causale e importo. All'invio della form con il bottone INVIA, l'applicazione controlla che il conto di destinazione appartenga all'utente specificato e che il conto origine abbia un saldo superiore o uguale all'importo del trasferimento. In caso di mancanza di anche solo una condizione, l'applicazione mostra una pagina con un avviso di fallimento che spiega il motivo del mancato trasferimento. Nel caso in cui entrambe le condizioni siano soddisfatte, l'applicazione deduce l'importo dal conto di origine, aggiunge l'importo al conto di destinazione e mostra una pagina CONFERMA TRASFERIMENTO che presenta i dati dell'importo trasferito e i dati del conto di origine e di destinazione con i rispettivi saldi precedenti al trasferimento e aggiornati dopo il trasferimento. L'applicazione deve garantire l'atomicità del trasferimento: ogni volta che il conto di destinazione viene addebitato, il conto di origine deve essere accreditato. Ogni pagina contiene un collegamento per tornare alla pagina precedente. L'applicazione consente il logout dell'utente.

- Pages, view components, events, actions.

Completamento delle specifiche

- Il controllo delle credenziali di login viene fatto su mail e password.
- Un utente ha id, e-mail ed username univoci.
- Quando l'utente fa il logout, deve tornare nella pagina di login.
- Dalla pagina di LOGIN è possibile anche effettuare una registrazione e creare un utente tramite una form che contiene i campi: e-mail, nome, cognome, username, password, ripeti password. Se l'username o l'e-mail sono già utilizzate l'utente visualizza l'errore altrimenti viene registrato.
- Se un utente cerca di accedere ad informazioni per cui non dispone l'autorizzazione viene indirizzato nella home page, se l'utente è loggato, altrimenti alla pagina di login.
- Dalla pagina di errore l'utente può tornare alla home page, se loggato, altrimenti alla pagina di login.
- Ad ogni conto è associato un nome che è univoco per ogni utente.
- Un utente loggato può creare dall'home page un nuovo conto tramite una form indicando il nome del nuovo conto da creare, nel caso l'utente abbia già un conto con questo nome, viene indirizzato a una pagina di errore, altrimenti il conto viene creato e si viene reindirizzati alla pagina home aggiornata.
- Un conto non può avere saldo negativo.
- L'importo del trasferimento deve essere strettamente positivo.
- Un trasferimento non deve avvenire se il conto sorgente e destinazione sono uguali tra loro.

Components

Model object (*Beans*):

- User
- BankAccount
- MoneyTransfer

Data access Objects (*DAO*):

- UserDAO
  - checkCredentials (email, pwd): User
  - getUserByEmail (email): User
  - getUserByUsername (username): User
  - registerUser (name, surname, email, username, pwd)

- BankAccountDAO
  - getBankAccountByID (id): BankAccount
  - findBankAccountsByUser (id_user): List<BankAccount>
  - findBankAccountByName (id_user, name_account): BankAccount
  - createBankAccount (id_user, name_account)

- MoneyTransferDAO
  - findMoneyTransferByAccountID (id_account): List<MoneyTransfer>
  - makeTransfer (id_bankAccount_src, id_user_dest, id_bankAccount_dest, reason, amount)
  - getLastTransfer(): MoneyTransfer

Filters:

- LoggedChecked
- NotLoggedChecked

Controllers (*servlets*):

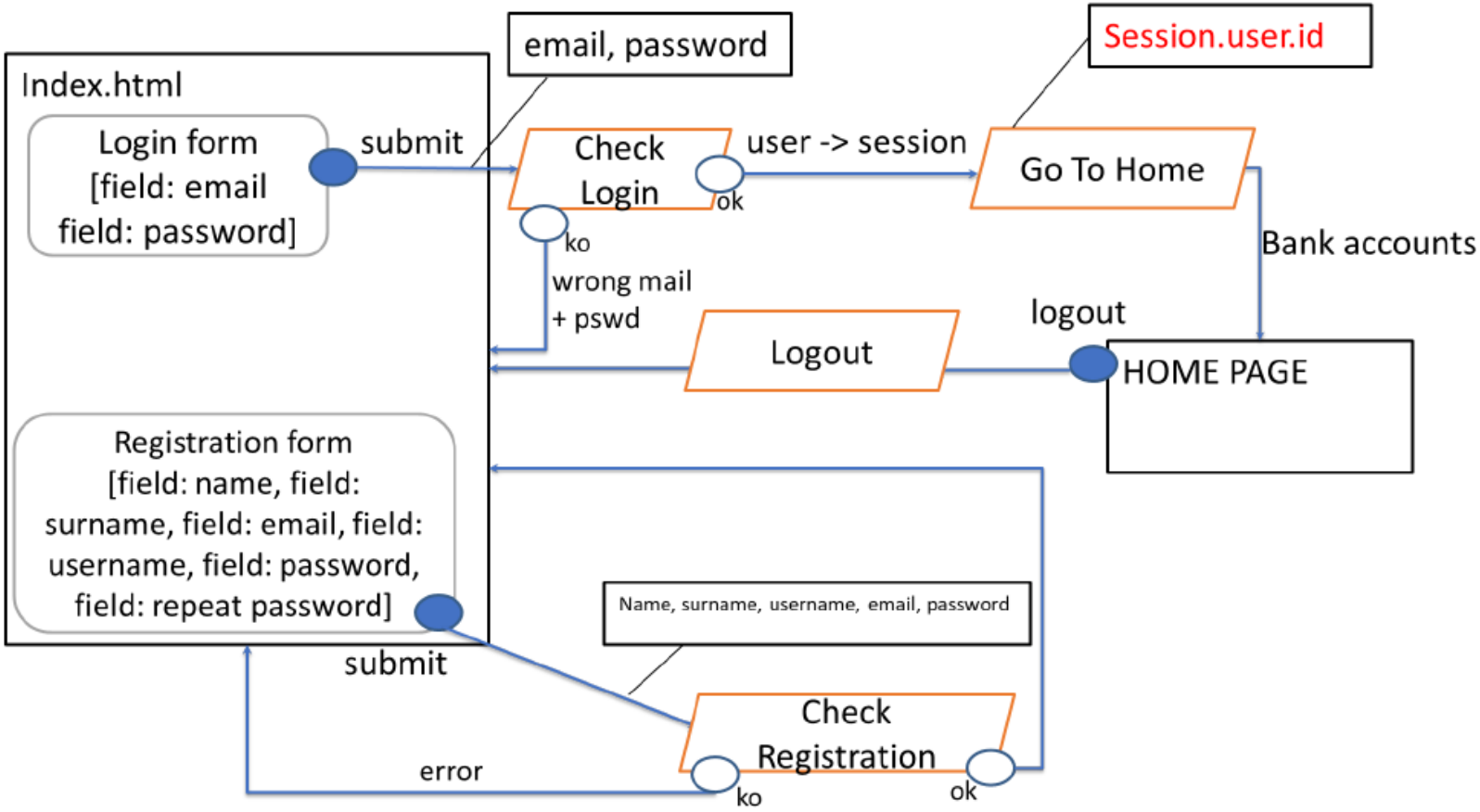
- CheckLogin [not logged]
- CheckRegistration [not logged]
- GoToHomePage [logged]
- GetBankAccount [logged]
- CreateBankAccount [logged]
- MakeTransfer [logged]
- Logout [logged]
- GoToTransferConfirmationPage [logged]

Views (*templates*):

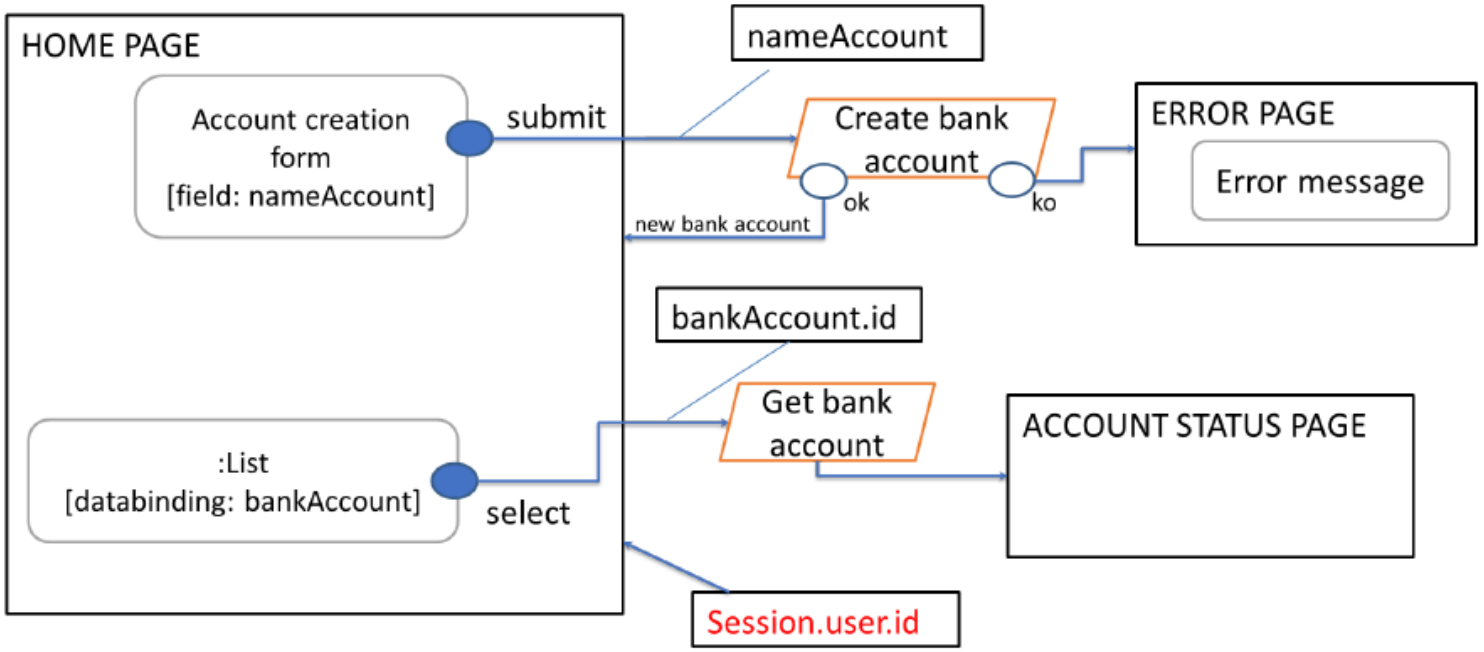
- index
- HomePage
- AccountStatusPage
- TransferConfirmationPage
- TransferFailurePage
- ErrorPage

Design applicativo (IFML)

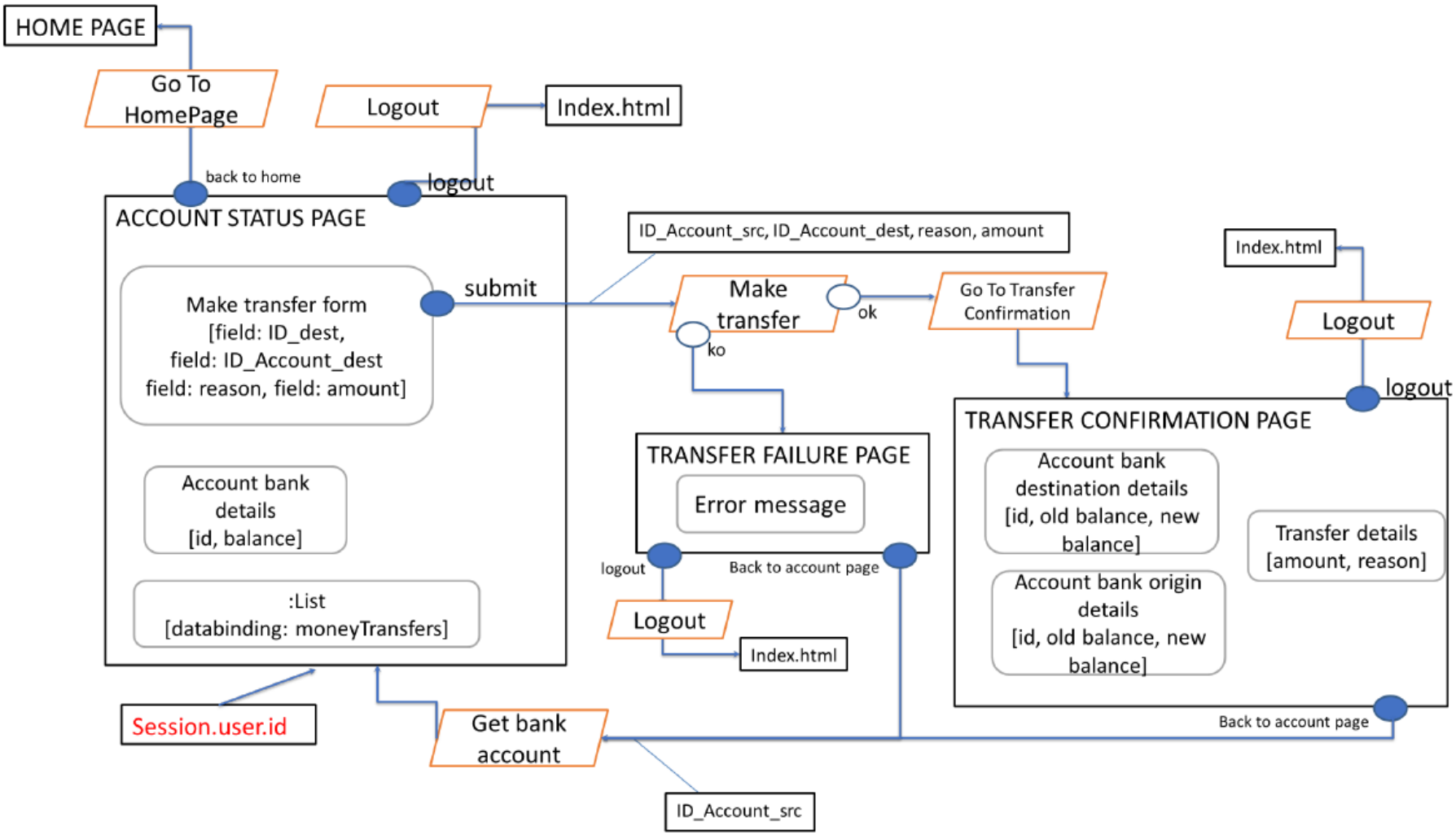
Login – Registration



Home


HOME PAGE
Account creation form [field: nameAccount]
submit
nameAccount
Create bank account
ok
ko
new bank account
ERROR PAGE
Error message
bankAccount.id
:List [databinding: bankAccount]
select
Get bank account
ACCOUNT STATUS PAGE
Session.user.id

Account status

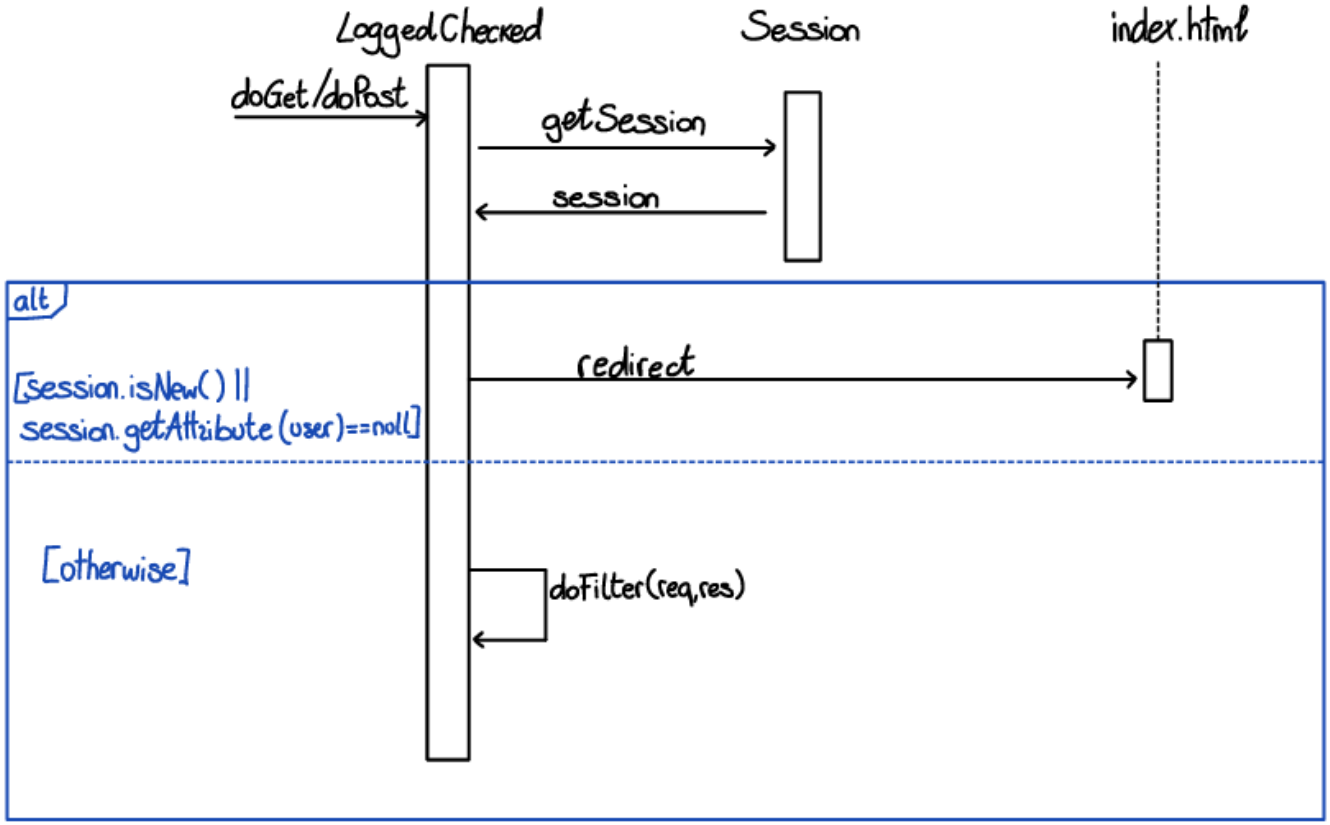


Sequence diagrams

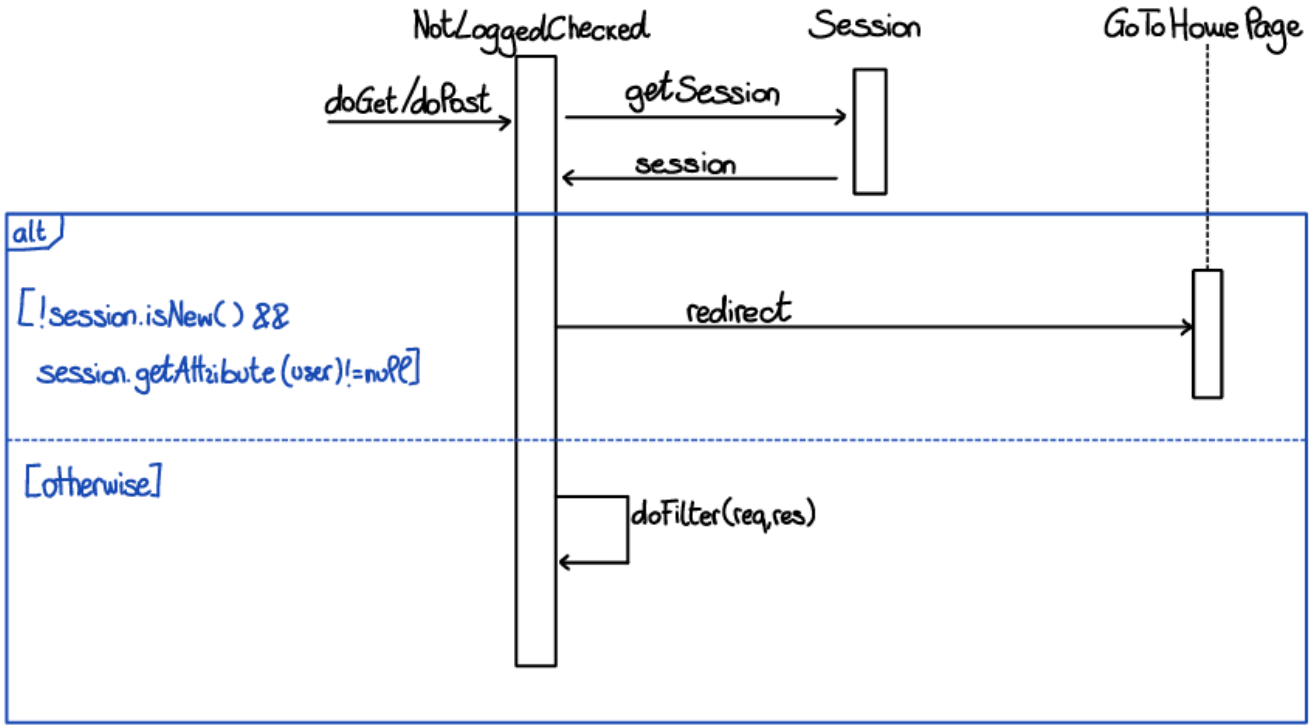
Some controls over parameters and potential internal errors and database access have been omitted for abbreviation.

The controls conducted by filters have been presented in the first diagrams and omitted in the following ones.

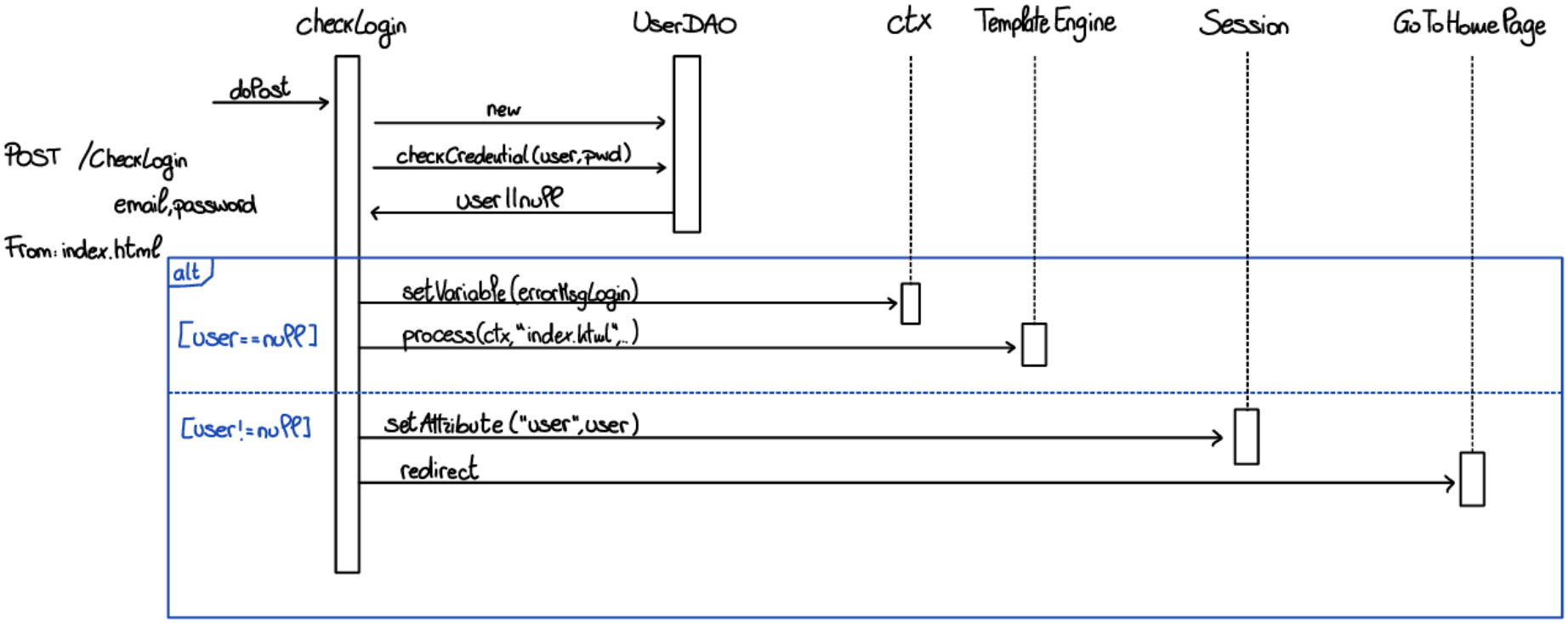
Logged Checked



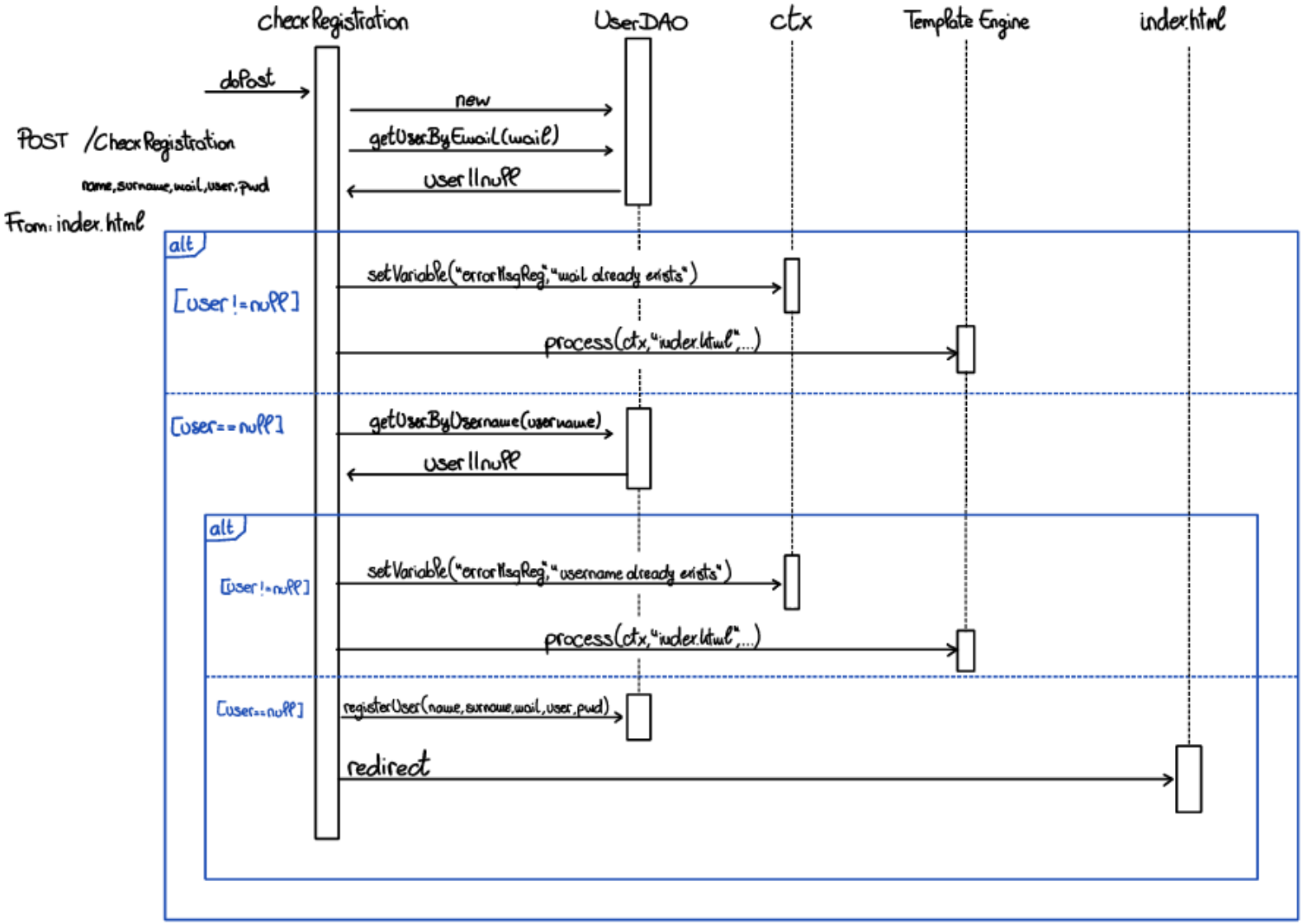
Not Logged Checked



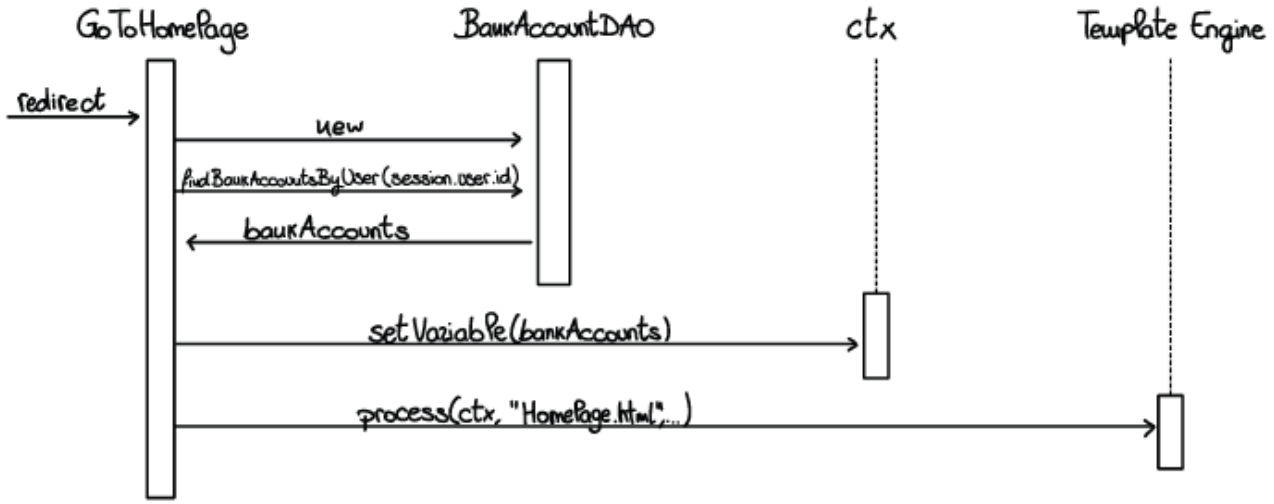
Check Login


checkLogin
UserDAO
ctx
Template Engine
Session
GoToHomePage
doPost
new
checkCredential(user,pwd)
user!=null
POST /CheckLogin
email,password
From: index.html
alt
[user==null]
setVariable(errorMsgLogin)
process(ctx,"index.html",..)
[user!=null]
setAttribute("user",user)
redirect

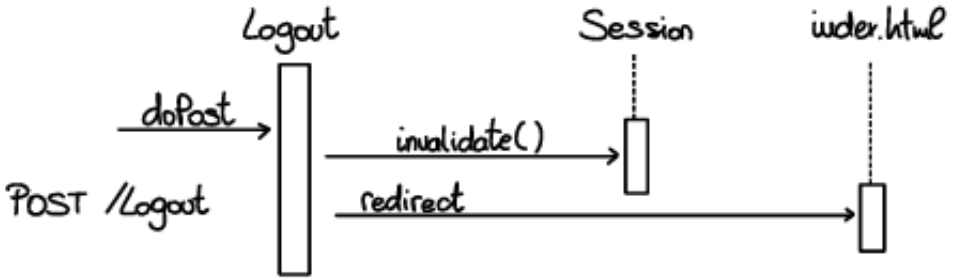
Check Registration



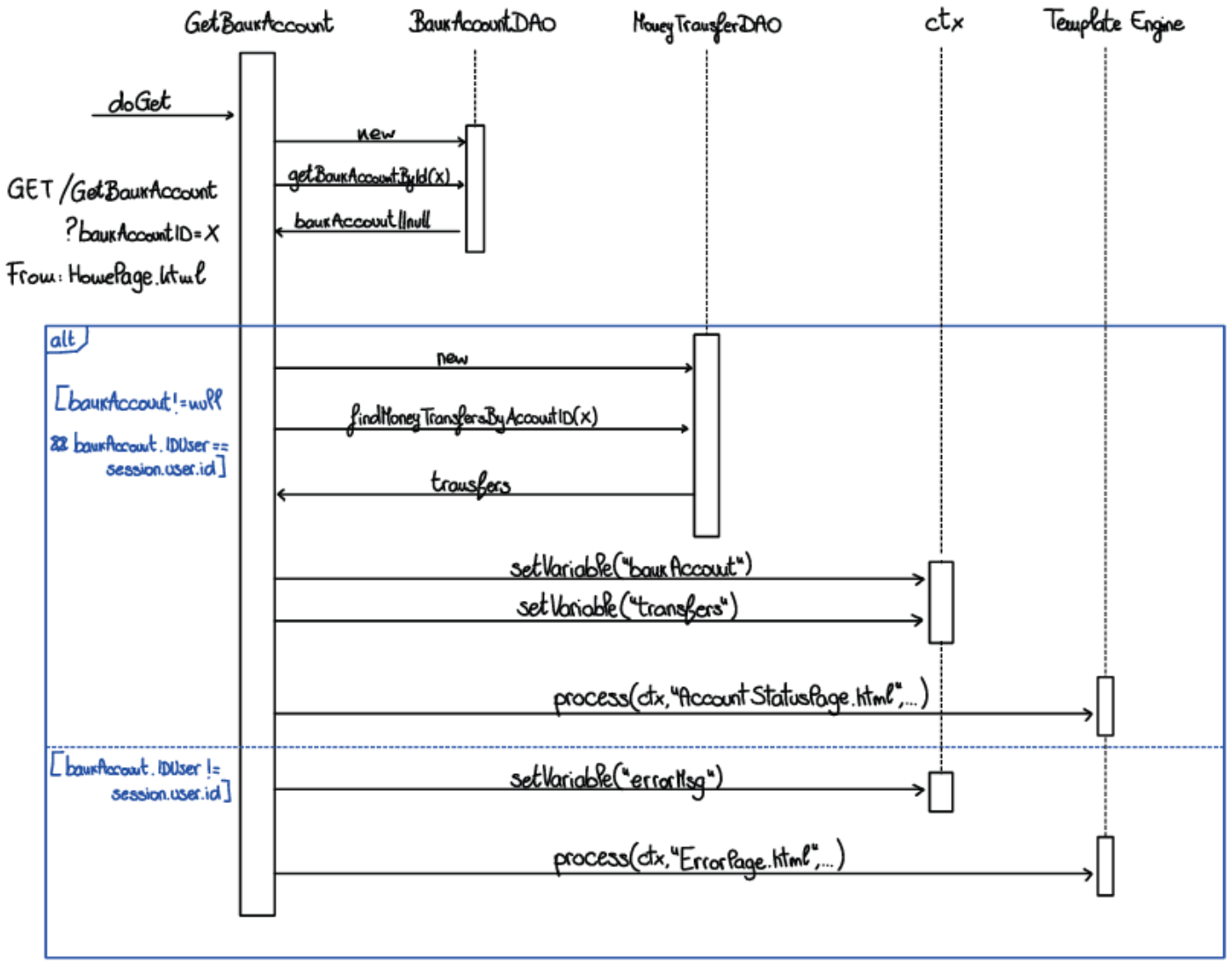
Go To Home Page



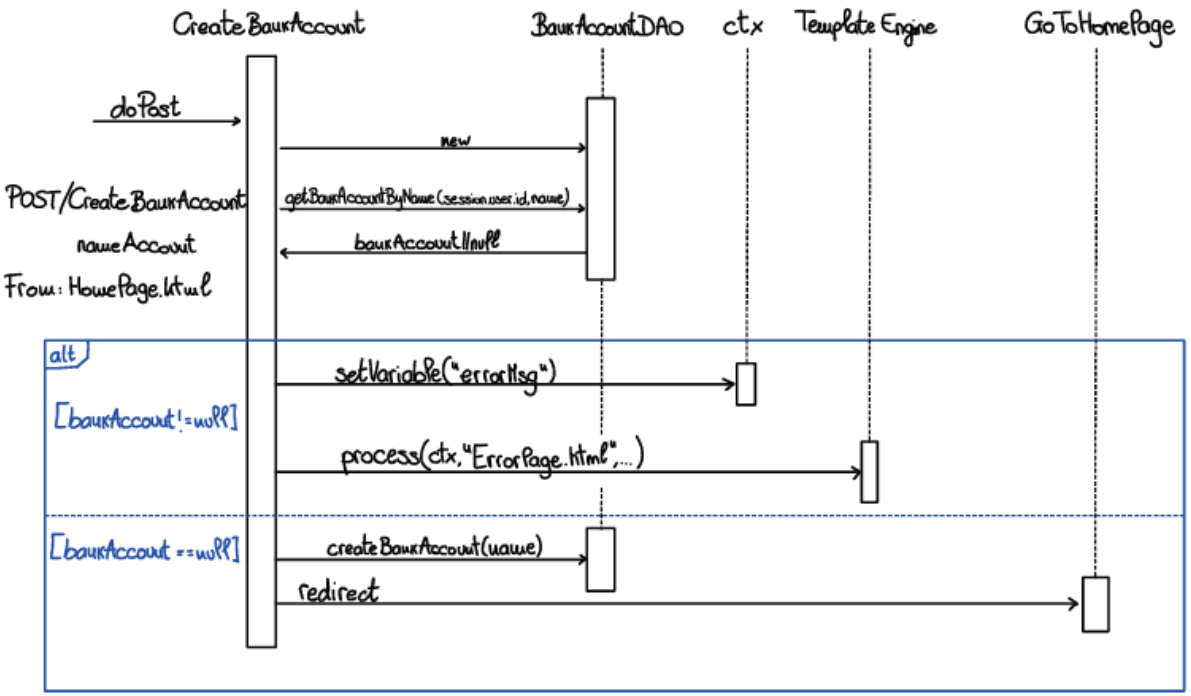
Logout



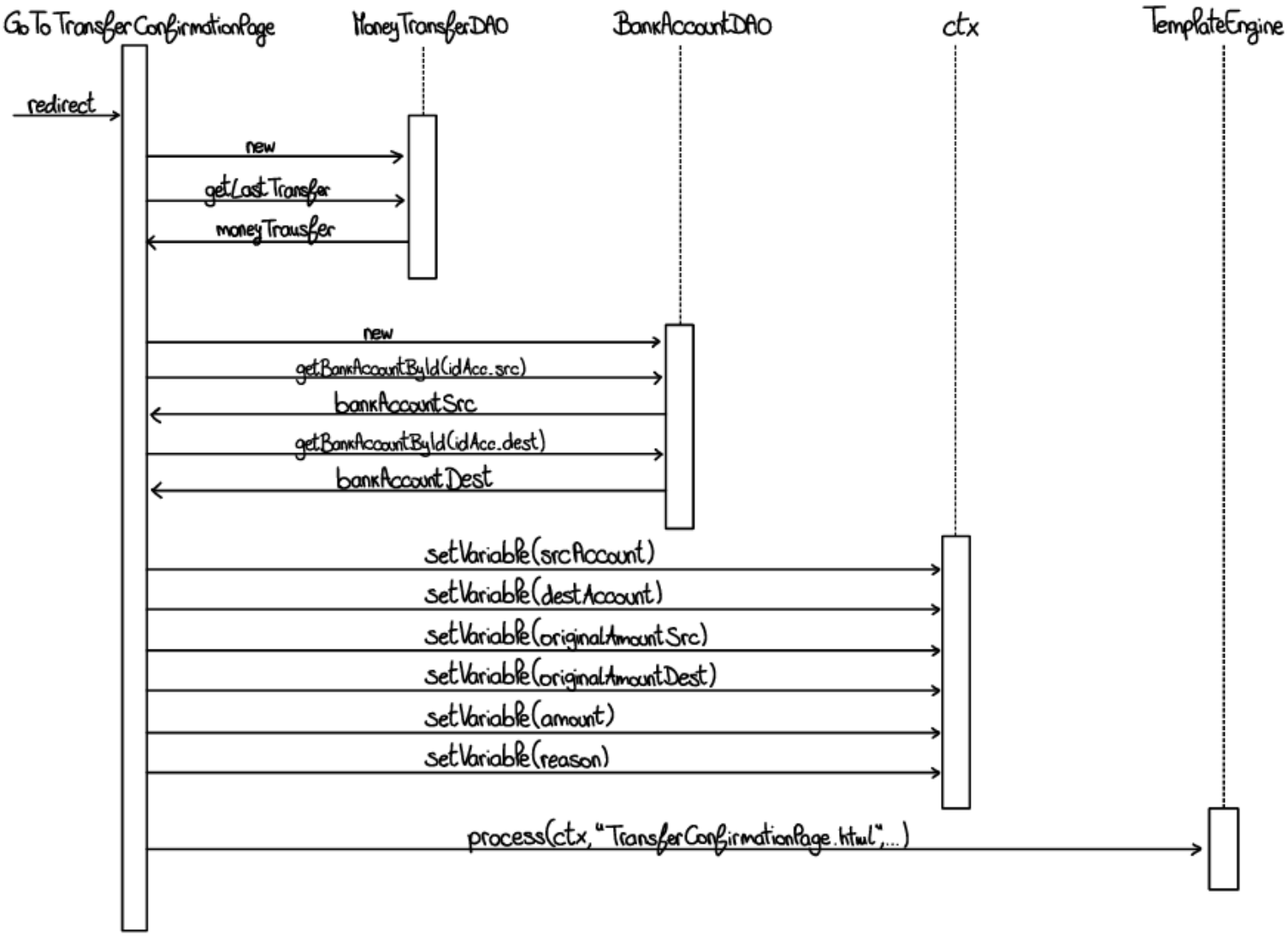
Get Bank Account



Create Bank Account



Go To Transfer Confirmation Page



Make Transfer