

Tecnologie Informatiche per il Web

Progetto 4 – Trasferimento Denaro

Anno accademico 2021-22



POLITECNICO
MILANO 1863

Niccolò Betto
Paolo Cerutti

Trasferimento Denaro

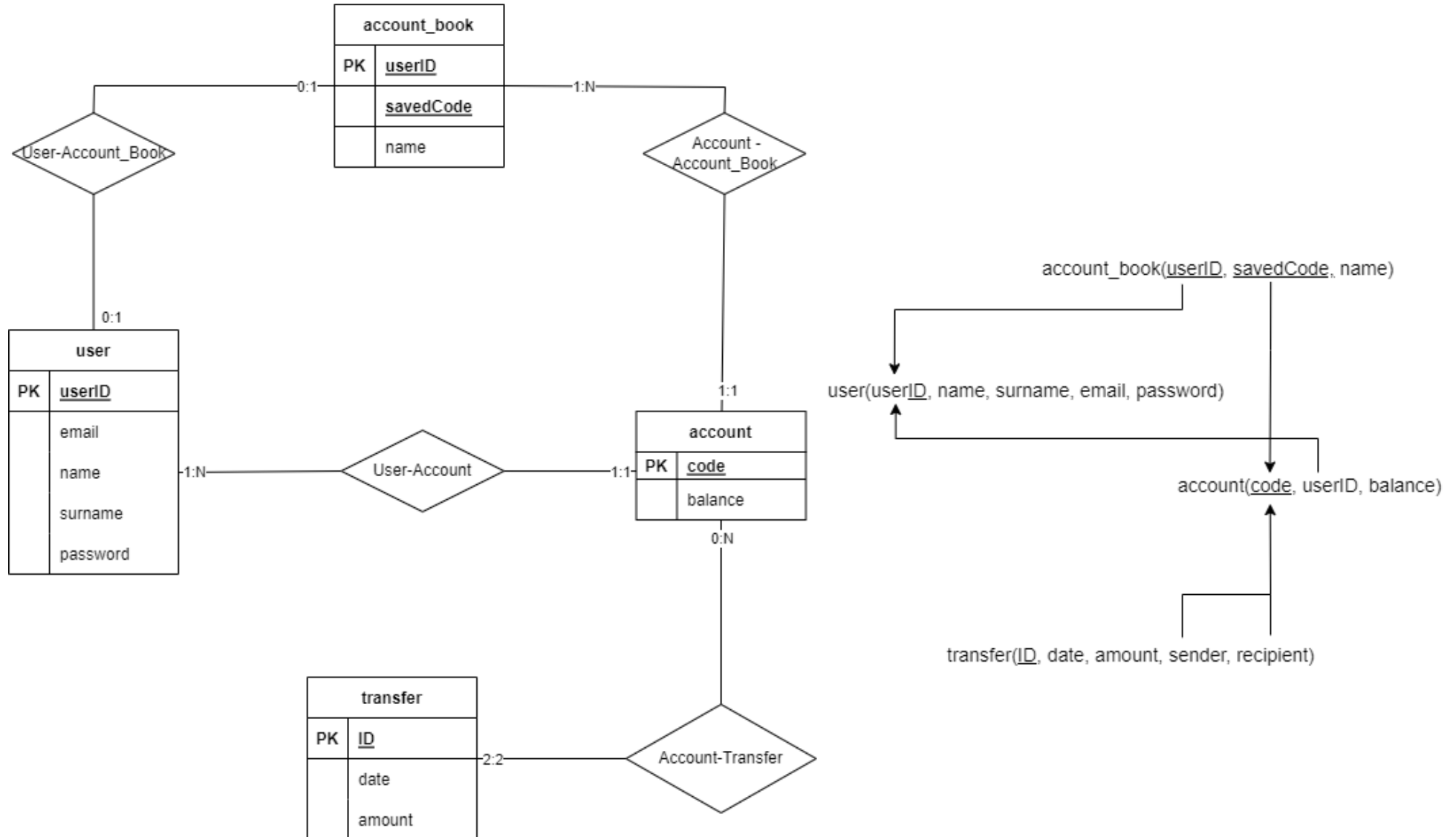
Un'applicazione web consente la gestione di trasferimenti di denaro online da un conto a un altro. L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password"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

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, relationships

Database Design



Schema Database

```
DROP TABLE IF EXISTS user;
CREATE TABLE user(
    `userID`      int PRIMARY KEY NOT NULL
    AUTO_INCREMENT,
    `email`       varchar(45)      NOT NULL,
    `name`        varchar(45)      NOT NULL,
    `surname`     varchar(45)      NOT NULL,
    `password`    varchar(45)      NOT NULL
) DEFAULT CHARSET = utf8mb4
  AUTO_INCREMENT = 1001
  COLLATE = utf8mb4_0900_ai_ci;
```

Schema Database

```
DROP TABLE IF EXISTS account;
CREATE TABLE account
(
    `code`      int                NOT NULL PRIMARY KEY
    AUTO_INCREMENT,
    `userID`    int                NOT NULL,
    `balance`  decimal(10, 2) NOT NULL DEFAULT 0,
    FOREIGN KEY (userID) REFERENCES `user` (userID) ON
    UPDATE CASCADE ON DELETE NO ACTION
) DEFAULT CHARSET = utf8mb4
  AUTO_INCREMENT = 2001
  COLLATE = utf8mb4_0900_ai_ci;
```

Schema Database

```
DROP TABLE IF EXISTS transfer;
CREATE TABLE transfer
(
    `ID`          int PRIMARY KEY NOT NULL AUTO_INCREMENT,
    `date`        date            NOT NULL,
    `amount`      decimal(10, 2)  NOT NULL,
    `sender`      int             NOT NULL,
    `recipient`   int             NOT NULL,
    `reason`      varchar(150)    NOT NULL,
    FOREIGN KEY (sender) REFERENCES `account` (code) ON UPDATE CASCADE ON
DELETE NO ACTION,
    FOREIGN KEY (recipient) REFERENCES `account` (code) ON UPDATE CASCADE ON
DELETE NO ACTION
) DEFAULT CHARSET = utf8mb4
  AUTO_INCREMENT = 100000
  COLLATE = utf8mb4_0900_ai_ci;
```

Schema Database

```
DROP TABLE IF EXISTS account_book;
CREATE TABLE account_book
(
    `userID`      int          NOT NULL,
    `savedCode`   int          NOT NULL,
    `name`        varchar(45) NOT NULL,
    PRIMARY KEY (userID, savedCode),
    FOREIGN KEY (userID) REFERENCES `user` (userID) ON UPDATE CASCADE ON
DELETE CASCADE,
    FOREIGN KEY (savedCode) REFERENCES `account` (code) ON UPDATE CASCADE ON
DELETE CASCADE
) DEFAULT CHARSET = utf8mb4
  COLLATE = utf8mb4_0900_ai_ci;
```


Analisi requisiti

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'**HOMEPAGE** 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**

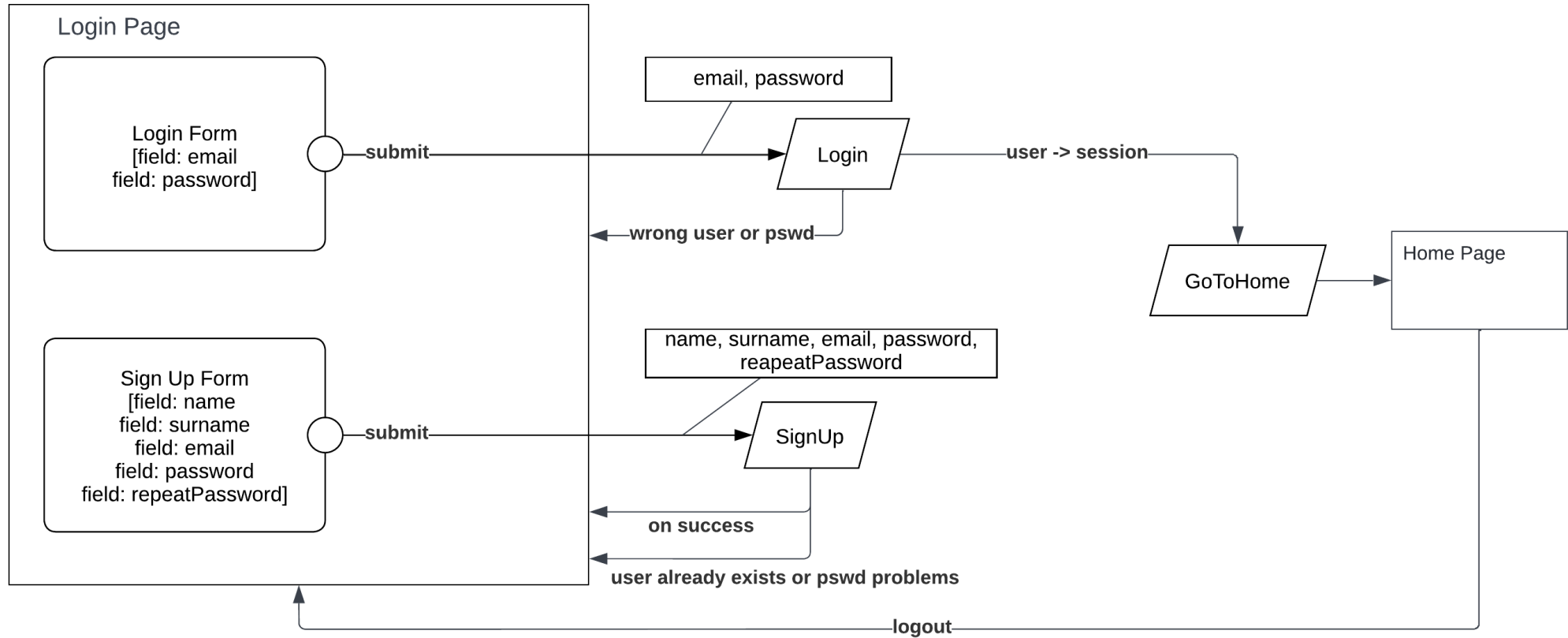
Analisi requisiti - RIA

- La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password", anche a lato client.
- 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.
- I controlli di validità dei dati di input (ad esempio importo non nullo e maggiore di zero) devono essere realizzati anche a lato client. L'avviso di fallimento è realizzato mediante un messaggio nella pagina che ospita l'applicazione.
- L'applicazione chiede all'utente se vuole inserire nella propria rubrica i dati del destinatario di un trasferimento andato a buon fine non ancora presente.
- Se l'utente conferma, i dati sono memorizzati nella base di dati e usati per semplificare l'inserimento.
- Quando l'utente crea un trasferimento, l'applicazione propone mediante una funzione di auto-completamento i destinatari in rubrica il cui codice corrisponde alle lettere inserite nel campo codice utente destinatario

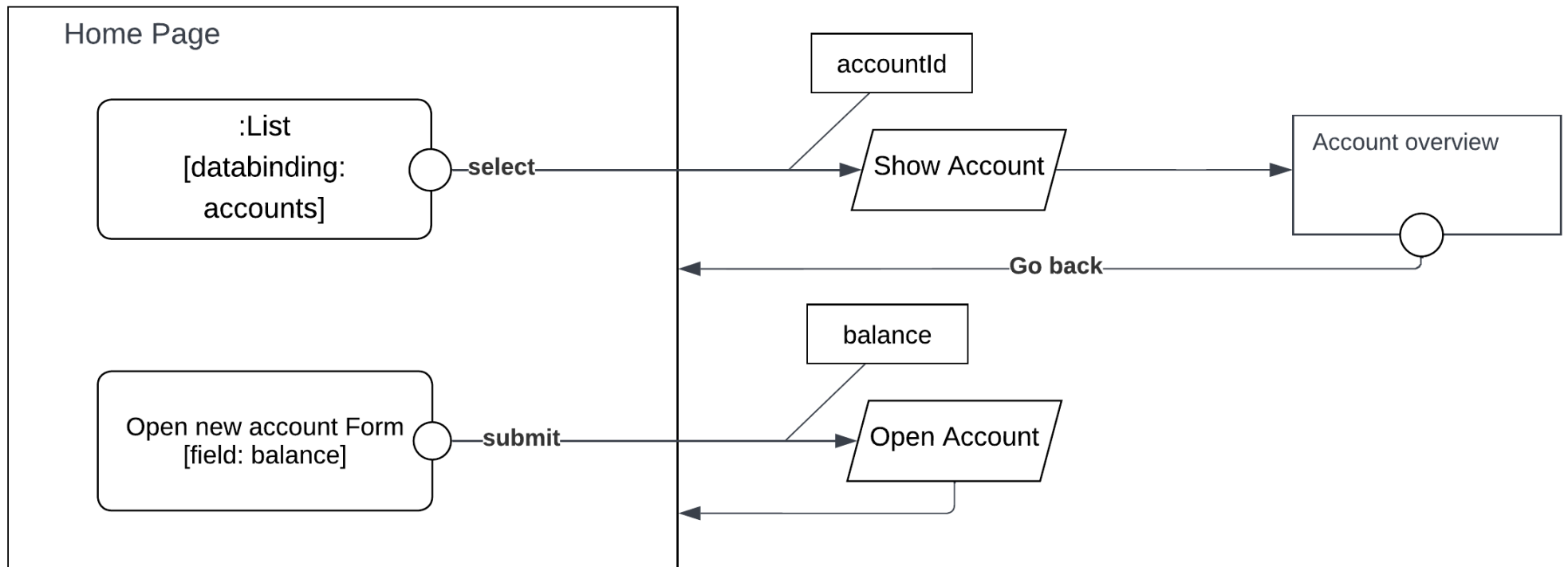
Completamento delle specifiche

- L'applicazione è interamente in lingua inglese, pertanto alcune parole chiave potranno differire dalla specifica.
- La pagina di default contiene la form di login e la form di registrazione
- Tutti i dati di registrazione sono obbligatori.
- I conti non possono avere saldi negativi.
- Un nuovo utente viene creato senza un conto associato, ma ha la possibilità di crearlo successivamente.
- A seguito della creazione di un trasferimento se ne visualizzano i dettagli.
- L'auto-completamento fornisce suggerimenti di ID e numero account dell'utente destinatario

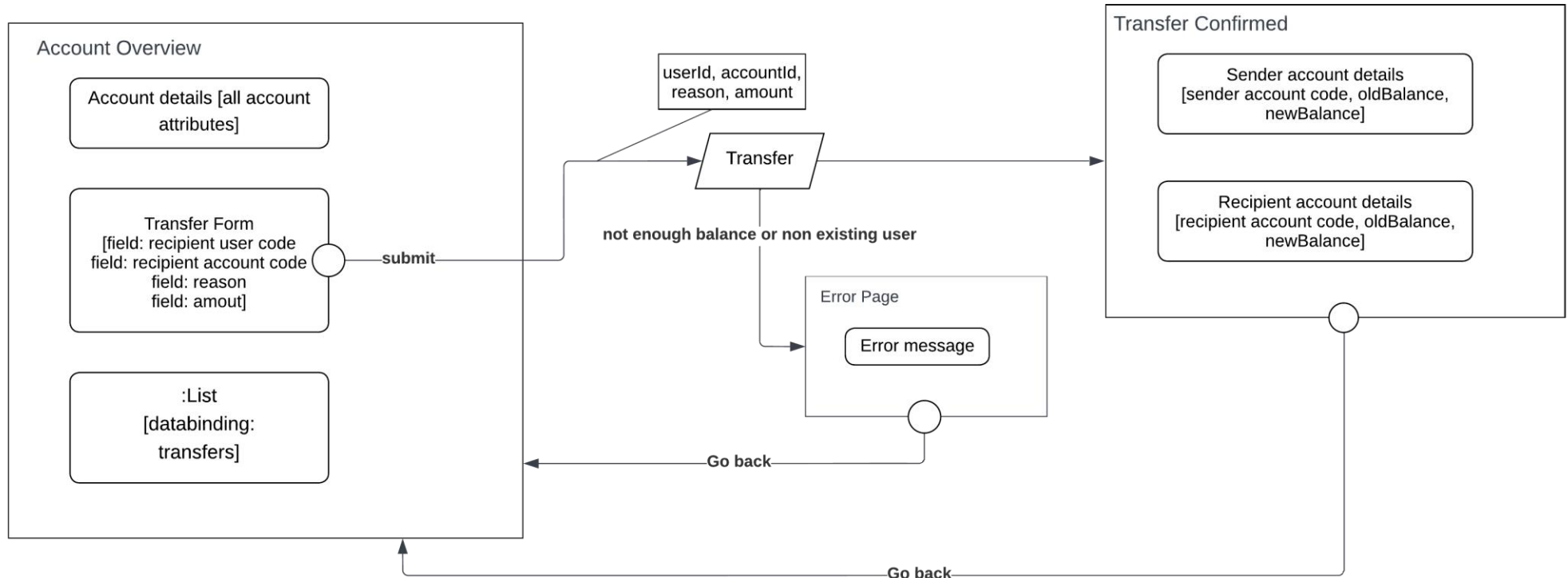
Pure HTML Flow Diagram - Login



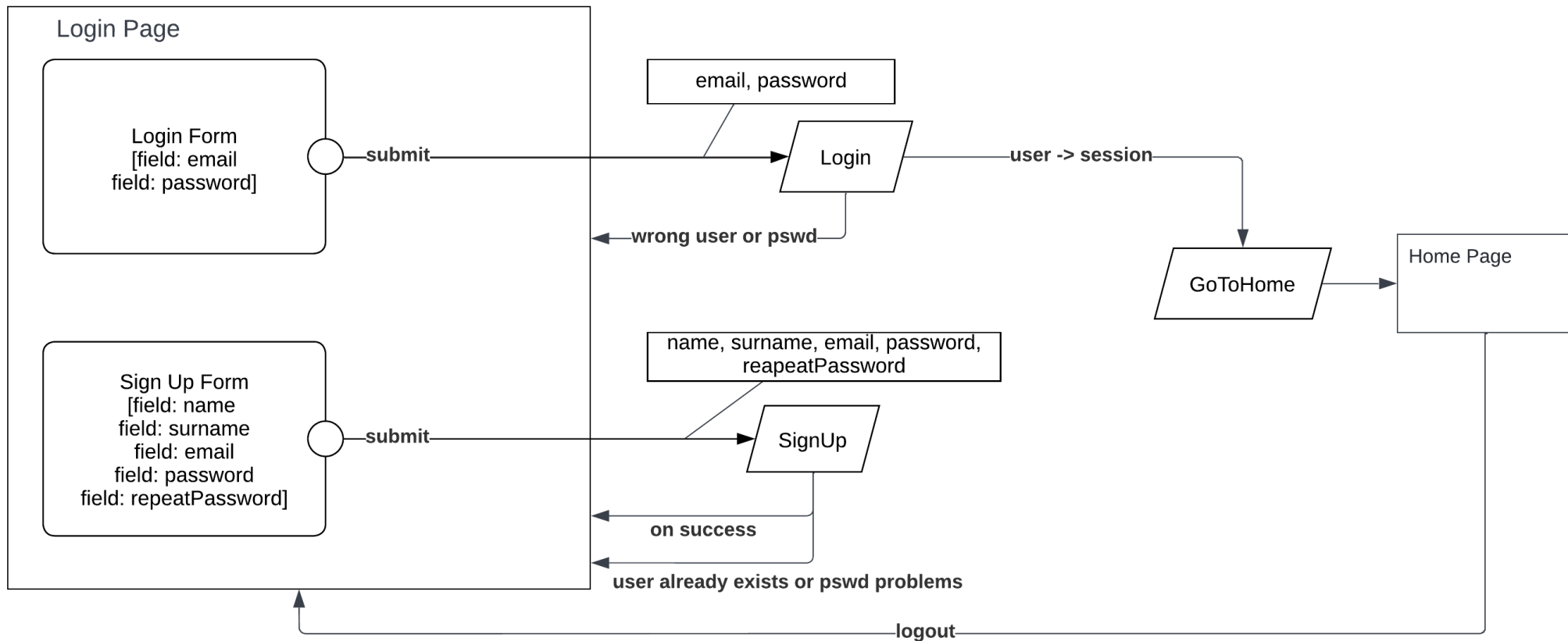
Pure HTML Flow Diagram - Homepage



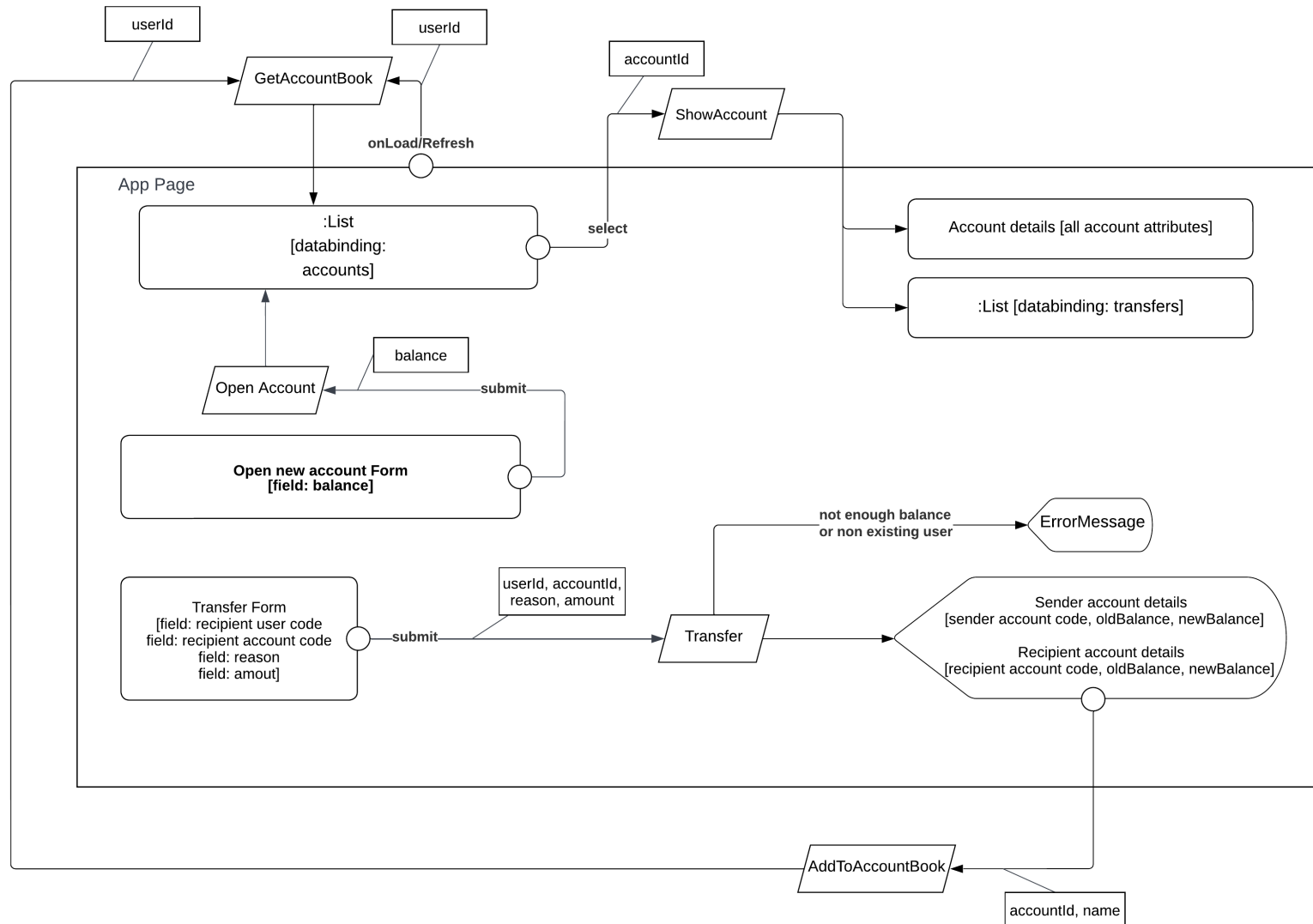
Pure HTML Flow Diagram - Account



RIA Flow Diagram - Login



RIA Flow Diagram - App



Server Side DAO & Objects - RIA

- Controllers:
 - AddToBook
 - GetAccountBook
 - GetAccountInfo
 - GetAccounts
 - GetTransfers
 - DoTransfer
 - Home
 - Login
 - Logout
 - OpenAccount
 - SignUp
- Beans:
 - Account
 - AccountBookEntry
 - Transfer
 - User
- DAO:
 - AccountBookDAO
 - AccountDAO
 - TransferDAO
 - UserDAO
- Filters:
 - LoggedInFilter
 - NotLoggedInFilter

Server Side DAO & Objects - PURE

- Controllers:
 - TransferResult
 - DoTransfer
 - Home
 - Login
 - Logout
 - OpenAccount
 - ShowAccount
 - SignUp
- Beans:
 - Account
 - Transfer
 - User
- DAO:
 - AccountDAO
 - TransferDAO
 - UserDAO
- Filters:
 - LoggedInFilter
 - NotLoggedInFilter
 - NoActiveAccountFilter