Digitálne meny a blockchain

**Hyperledger Fabric smart systém**

Roman Bitarovský, Emma Macháčová

**Meno cvičiaceho** : Viktor Valaštín

**Čas cvičení** : Streda, 18.00

**Dátum vytvorenia** : 01.05. 2022

Obsah

[1. Cieľ projektu 1](#_Toc102313804)

[2. Voľba implementačného prostredia 2](#_Toc102313805)

[3. Návrh architektúry a jej prepojenia 3](#_Toc102313806)

[4. Implementované časti kódu 4](#_Toc102313807)

[5. Opis prostredia na testovanie a opis vybraných testov 5](#_Toc102313808)

[6. Užívateľská príručka 6](#_Toc102313809)

[7. Odpovede na otázky 7](#_Toc102313810)

[8. Záver 8](#_Toc102313811)

1. Cieľ projektu

Cieľom projektu je implementovať a konfigurovať **biznis sieť nazvanú FlyNet**, a teda vytvoriť biznis sieť poskytujúcu viac efektívnu B2B (biznis-to-biznis) spoluprácu s využitím **Hyperledger Fabric**.

Spoločnosti vo FlyNet majú rôzne role. EconFly a BusiFly vytvárajú lety a objednávajú sedadlá a GladlyAbroad dopytuje lety a rezervuje sedadlá pre zákazníkov. Keď je prijatá rezervácia od cestovnej kancelárie, korešpondujúca aerolínia objedná požadované sedadlá aktualizovaním informácií v blockchaine.

Aplikácie môžu interagovať so sieťou a aktualizovať stav pomocou chaincode funkcií. Možnými transakciami sú:

* **createFlight(flyFrom, flyTo, dateTime, seats)**
  + Môže byť volaná jedine aerolíniou na vytvorenie nového letu s danými from, to, dateTime a počtom seats dostupných v lete. Potom let uloží do ledgeru a vráti správu o úspechu volania iba ak je volajúci tejto funkcie niekto z aerolínií. Inak by mala vrátit error. Parameter flightNr nevstupuje ako argument funkcie, musí byť vygenerovaný automaticky.
* **getAllFlights()**
  + Vráti všetky dostupné lety uložené v ledgeri.
* **getFlight(id)**
  + Vráti informácie o lete s daným id.
* **reserveSeats(flightNr, number)**
  + Môže byť volaná jedine cestovnou kanceláriou na rezervovanie number sedadiel na lete s id flightNr.
* **bookSeats(reservationNr)**
  + Volaná jedine aerolíniami na zmenu stavu rezervácie s reservationNr na „completed“ a aktualizovanie záznamu na ledgeri korešpondujúceho letu. Sedadlá sú objednané iba keď je požadované množstvo sedadiel stále dostupné na danom lete a keď flightNr rezervácie patrí aerolíniám, ktoré spracúvajú danú rezerváciu sedadiel.
* **checkIn(reservationNr, passportIDs[])**
  + Volaná buď cestovnou kanceláriou alebo konečným zákazníkom za účelom vybrať si konrétne sedadlá v lietadle a zaevidovanie cestovných dokladov na účely evidencie pasžierov v letoch. Cestovné doklady sú pár customerNames z rezervácie a čísla dokladu. Môžu byť dodané ako pole alebo ako zoznam, príp. aj ako string, kde sú jednotlivé záznamy oddelené čiarkou. Príklad záznamu: „Andrej Andrejev/OP123456, Bohuš Bohuš/PA123456“ Táto žiadosť je spracovávaná chaincode a zmení stav rezervácie na ledgeri na „Checked-In.“ Kontroluje, či sú žiadané sedadlá voľné a či sedia customerNames a reservationNr s rezerváciou. Vráti správu o úspechu volania a pošle email s „letenkami“ na emailovú adresu customerEmail z rezervácie.

2. Voľba implementačného prostredia

Projekt je implementovaný v prostredí HyperLedger Fabric (verzia 2.2.), so synchronizáciou na localhoste, ktorý je spúšťaný pomocou softvéru Docker (verzia 20.10.14.) a nástrojom Docker Compose (verzia 1.29.2.) v operačnom systéme MacOS.

Kód bol písaný v editore Visual Studio Code, v jazyku JavaScript. Projekt je spúšťaný a beží pomocou CLI.

Boli využité tiež node 16.5.0. a npm 8.5.5.

3. Návrh architektúry a jej prepojenia

Sieť sa skladá z:

* *doplniť*
* *Org1*
* *Org2*
* *CA*
* *myChannel1*

*Dôvod*

*Ideálna architektúra*

4. Implementované časti kódu

Projekt bol vytvorený upravením fabric-samples a asset-transfer-basic, a obsahuje tieto významnejšie časti kódu:

4.1. Asset Transfer

*doplniť*

4.2. Application

*doplniť*

5. Opis prostredia na testovanie a opis vybraných testov

5.1. Opis prostredia

Testovanie bolo vykonané v prostredí *doplniť*

5.2. Opis vybraných testov

V rámci testovania boli vykonané nasledovné testy:

*doplniť*

6. Užívateľská príručka

Po spustení programu sa zobrazí CLI. Výber možností je na účely obmedzenia chýb umožnený metódou prompt(). Pohyb medzi jednotlivými možnosťami sa vykonáva pomocou arrow-keys na klávesnici. CLI vyzerá nasledovne:

*doplniť*

Ako prvé je potrebné vybrať si užívateľskú rolu (EconFly, BusiFly alebo cestovná kancelária). Ďalej sú užívateľovi prístupné funkcie podľa jeho roly:

6.1. CLI pre cestovnú kanceláriu

Cestovná kancelária môže vykonávať funkciu reserveSeats, ktorá slúži na rezervovanie určitého počtu sedadiel v konkrétnom lete zadanom pomocou flightNr.

Argumenty pre funkciu sa zadávajú v poradí: počet sedadiel, číslo letu.

6.2. CLI pre EconFly a BusiFly

Letecké spoločnosti môžu vykonávať tieto funkcie:

* createAsset
* getAllAssets
* readAsset

6.3. Spúšťanie aplikácie

Aplikácia sa spúšťa nasledovnými príkazmi:

7. Odpovede na otázky

Napíšte krátke vysvetlenie a vaše názory:

1. **Podľa vášho názoru, je takéto blockchain-based riešenie najlepšia možnosť na vyriešenie daných výziev?** 
   * Odpoveď:
2. **Aké sú výhody a nevýhody používania technológie distribuovanej účtovnej knihy (distributed ledger technology) v porovnaní s centralizovaným systémom na tomto konkrétnom prípade použitia?**

* Odpoveď:

8. Záver

V závere sme dospeli k tomu, že

*doplniť*