NÁVRH RIEŠENIA – SEMESTRÁLNA PRÁCA

REZERVAČNÝ SYSTÉM PRE LETECKÉ SPOLOČNOSTI

Radovan Murin
murinrad@fel.cvut.cz

OBSAH

ÚČEL DOKUMENTU

Účelom tohto dokumentu je predstaviť riešenie semestrálnej práce na predmet A4M36AOS.

ZHRNUTIE

Dokument predstavuje riešenie semestrálnej práce na predmet Architektury orientované na služby.

POŽIADAVKY NA SYSTÉM

FUNKČNÉ POŽIADAVKY

1. Systém musí umožňovať nakupujúcemu vyhľadať si letenku.
2. Systém musí umožňovať nakupujúcemu rezervovať si letenku.
3. Systém musí umožňovať nakupujúcemu zaplatiť si letenku.
4. Systém musí umožňovať nakupujúcemu vytlačiť si letenku.
5. Systém musí umožňovať nakupujúcemu zmeniť/vymeniť si letenku.
    a. Výmena/zmena podlieha obmedzeniam.

NEFUNKČNÉ POŽIADAVKY

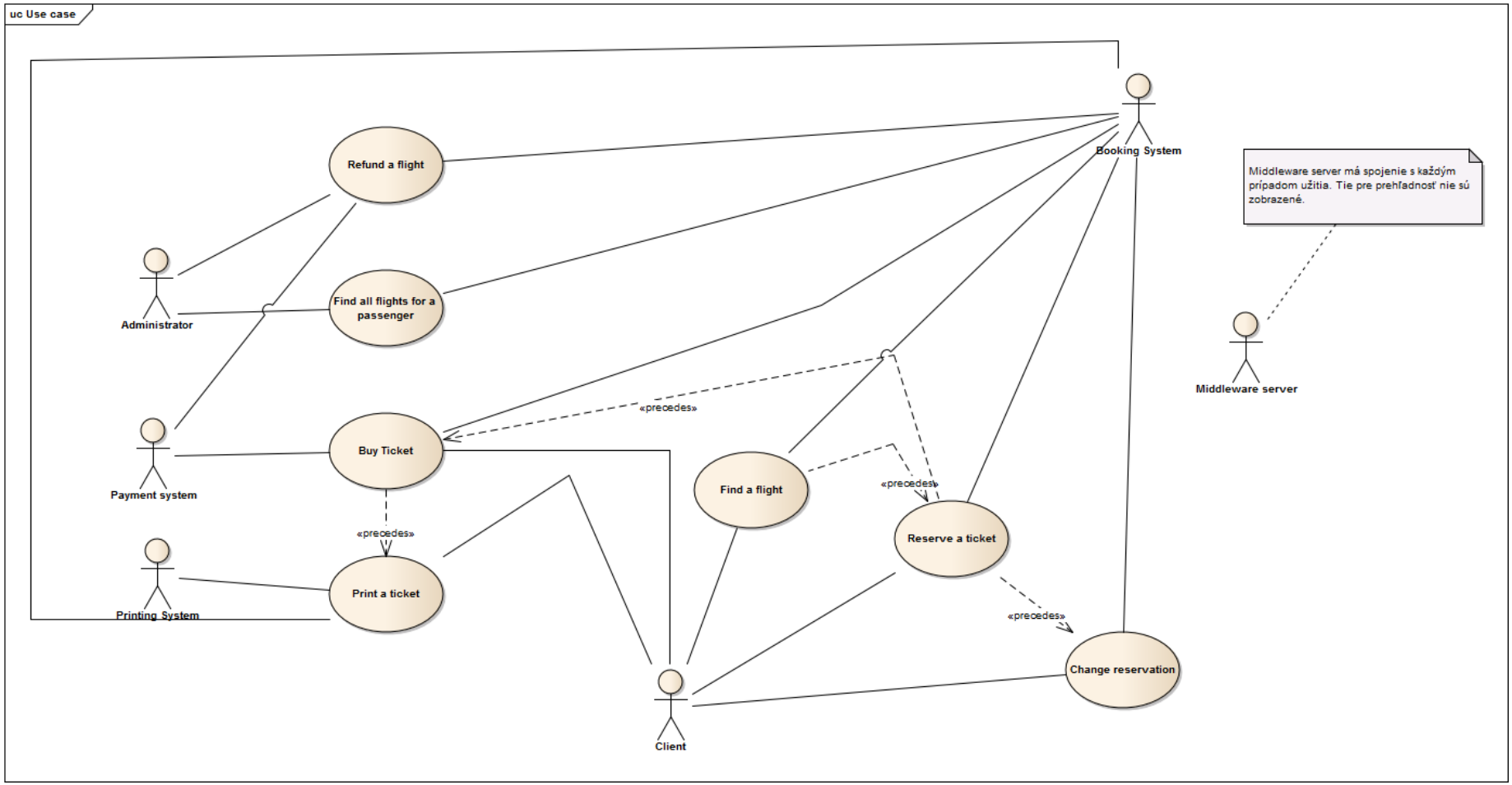
1. Systém musí pozostávať z 3 individualnych častí.
    a. Klient aplikácia
    b. Middleware server
    c. Backendy – 3 osobitné servre
        i. Rezervačný backend
        ii. Platobný backend
        iii. Tlač backend
2. Platobný backend musí umožňovať audit transakcií.
3. Rezervácia a platba za rezerváciu musí prebiehať pomocou transakcie.

NÁVRH

POUŽITÉ TECHNOLÓGIE

Servrové aplikácie budú postavené na technológií Java EE. Klientská aplikácia bude Java SE knižnica. MySQL bute tvoriť databázový systém. Tento softvér bol vybraný, pretože je to slobodný softvér a teda zadarmo. Ako aplikačný server bol zvolený Glassfish 3.1.2 .

USE CASE DIAGRAM



Obrázok 1Use case diagram

METÓDY VZDIALENÝCH ROZHRANÍ

INTERFACE SERVER

- findFlight – vracia letové informácie na základe dátumov a letísk
- bookFlight – rezervuje let pre pasažiera
- changeReservation – zmení, ak to dovoľujú pravidá, rezerváciu
- cancelReservation – zruší, ak to dovoľujú pravidá, rezerváciu
- printTicket – vytlačí lístok ak bol zaplatený
- payForAReservationCash,payForAReservationCard – platobné operácie
- getFreeseatsForFlight – nájde voľné miesta pre let
- getTicketsPerPassenger – vráti vsetky lístky ktoré vlastní pasažier

BOOKING SERVER

Booking server má kvôli prehľadnosti rozdelené metódy na 2 skupiny.

FlightBooker

- getFreeSeats – nájde voľné miesta pre let
- bookFlight – rezervuje let pre pasažiera
- cancelReservation – zruší , ak to dovoľujú pravidá, rezerváciu
- getTicketsForPassenger – vráti vsetky lístky ktoré vlastní pasažier
- changeReservation – zmení, ak to dovoľujú pravidá, rezerváciu
- confirmTicketPaid – zaznamená, že lístok bol zaplatený
- getSpecificTicket – vráti lístok na základe ID
- getPassengerDataForATicket – vráti pasažiera, ktorému patrí lístok

FlightFinder

- findFlightByID – vráti letové informácie na základe ID letu.
- findFlightsByFlightData – vráti letové informácie na základe parametrov: čas odletu, čas príletu, miesto odletu a príletu (IANA formát)

PAYMENT SERVER

Všetky operácie sa pokúsia o platbu a ak je úspešná vráti platbu s upraveným obsahom – príznak úspechu.

- performCashTransaction
- performCardTransaction
- preformBankTransaction

PRINTING SERVER

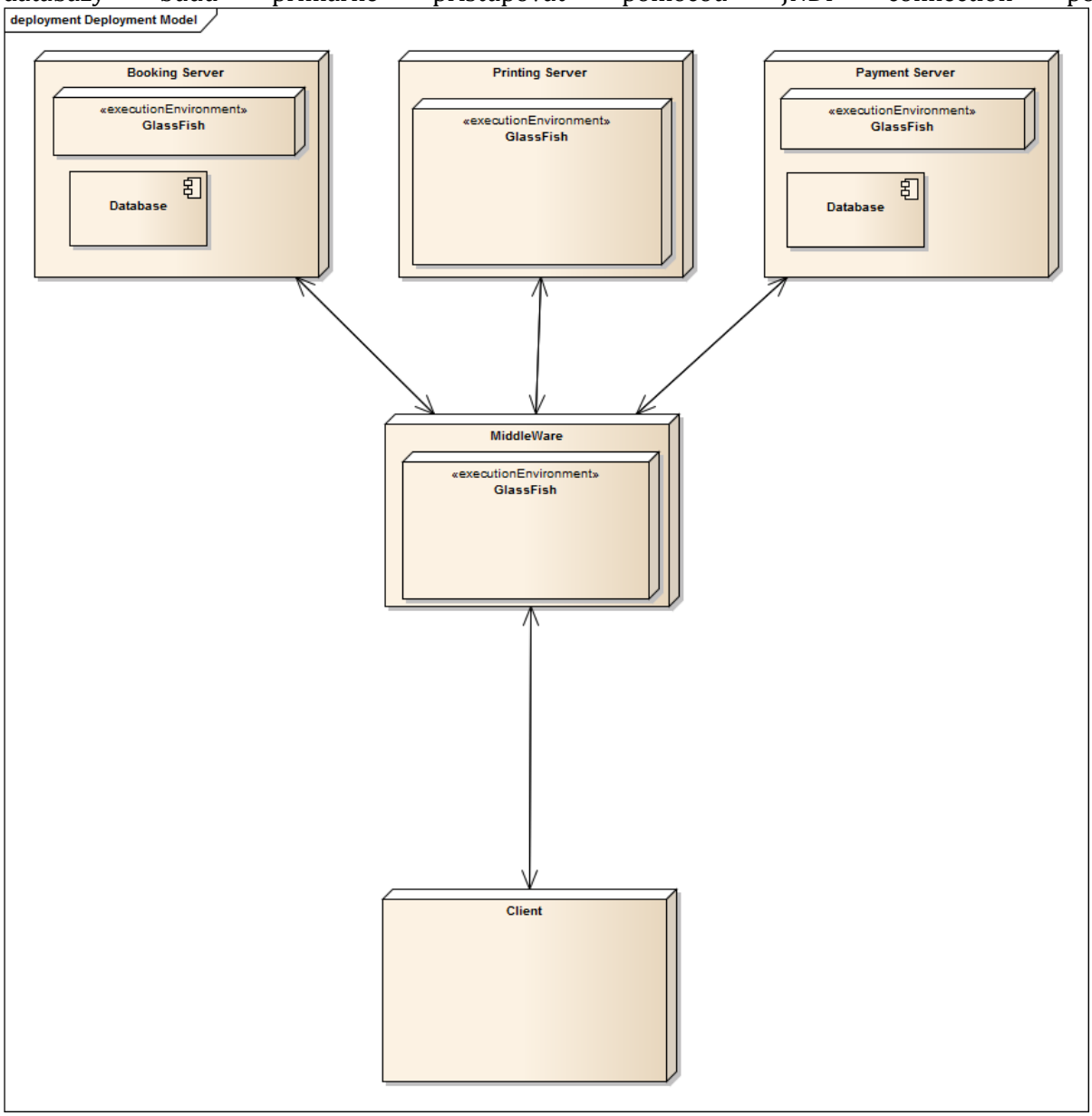
- printTicket – vráti textový súbor ktorý obsahuje letenku.

ARCHITEKTÚRA

Z požiadaviek na systém je evidentné, že systém sa bude skladať z 5 osobitných aplikácií. Tieto aplikácie budú, okrem klientskej, spustené na aplikačných servroch. Klientská aplikácia bude Java knižnica, ktorá bude sprístupňovať služby middleware servera koncovým aplikáciám.

Z bezpečnostných dôvodov budú backendy prístupné iba z middleware servera cez internú sieť a teda nebude nutné šifrovať správy, ktoré si vymieňajú.

Rezervačný a platobný systém budú potrebovať databázu pre ukladanie perzistentných dát. Do databázy budú primárne pristupovať pomocou JNDI connection pools.

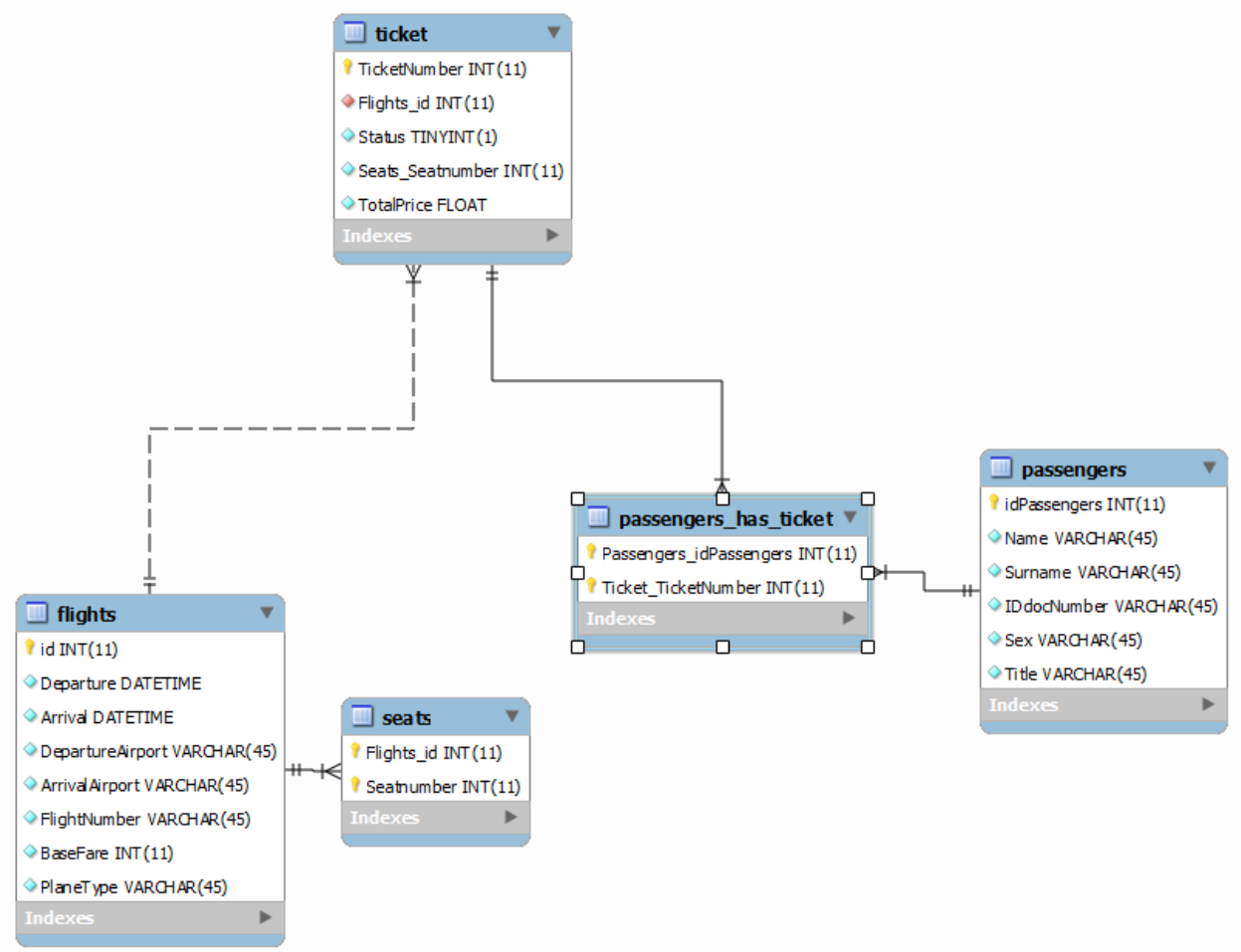


Obrázok 2 diagram nasadenia

DATABÁZOVÝ MODEL

BOOKING SERVER

Server na vykonávanie operácií s letmi potrebuje dáta v persistentnej databáze. Na túto činnosť potrebuje tabuľku letov. Entita let je previazaná s istým počtom sedadiel, ktoré sú identifikované na základe kombinácie ID letu a čísla sedadla. Ďalšími entitami sú samotná letenka a pasažier. Pasažier môže mať viac leteniek. Kvôli tomu bola vytvorená osobitná tabuľka „passengers_has_ticket“.



Obrázok 3 EER Diagram bookovacej databázy

PLATOBNÝ SERVER

Platobný server si bude v perzistentnej vrstve ukladať údaje o všetkých transakciách – úspešných a neúspešných. Dôvodom je spatná kontrola platieb. Keďze jednou bankovou transakciou sa dajú zaplaťiť viaceré letenky, zoznam leteniek, ktoré boli zaplatené v transakcií, je v asociovanej tabuľke.

SEKVENČNÝ DIAGRAM

Na sekvenčných diagramoch je možné vidieť ako prebiehajú volania služieb, ktoré zahŕňajú viac ako 2 servery.

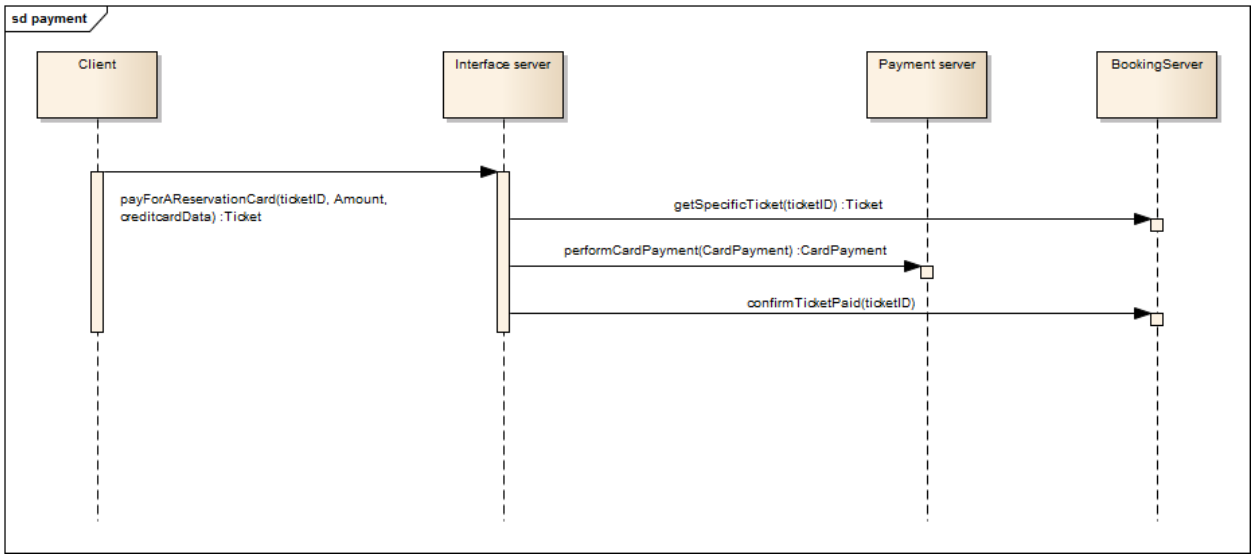


Figure 1 Digram platby pomocou karty

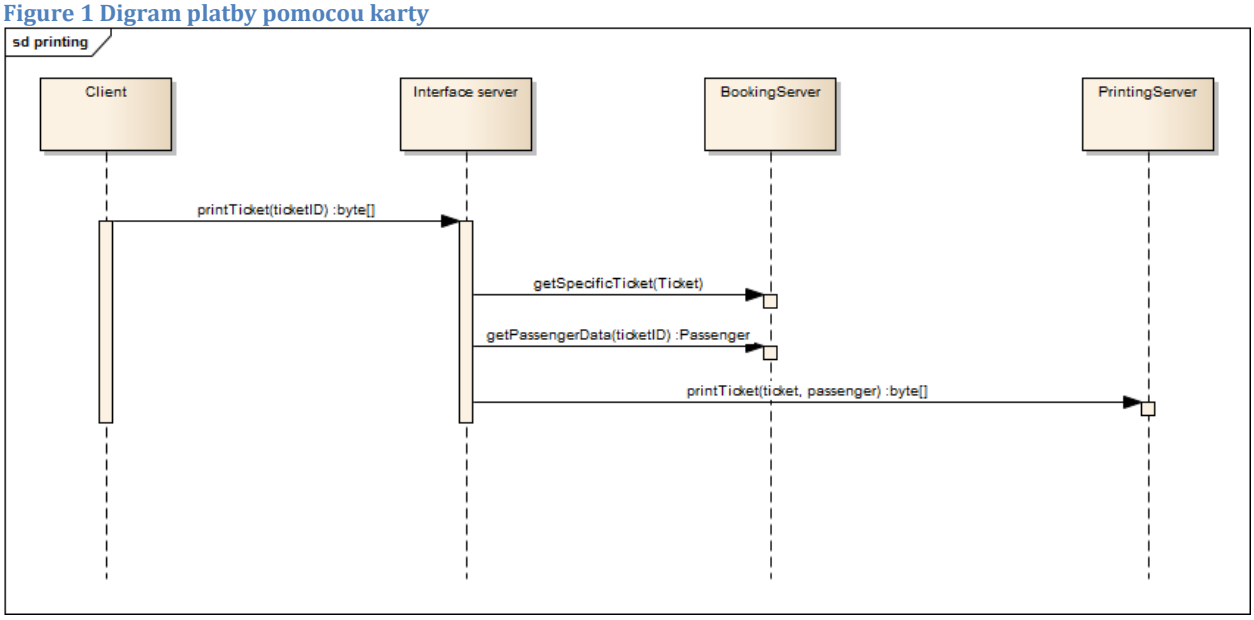


Figure 2 Diagram tlače lístka

IMPLEMENTAČNÉ POZNÁMKY

TRANSAKČNÉ SPRACOVANIE KRITICKÝCH OPERÁCIÍ

Všetky finančné operácie a všetky operácie, kde sa je nutný postupný zápis do databázy sú ošetrené tak,že v prípade pádu databázy, alebo inej chyby nastane rollback a dáta ostanú v konzistentnom stave.

BUSINESS LOGIKA

O business logiku sa stará trieda BusinessPolici v balíčku cz.ctu.fee.murinrad.bookingserver.utils. V tejto triete sa nachádzajú metóda ktorá určuje celkovú cenu letenky a tiež obsahuje počet dní pred odletom, kedy sa ešte dá let zrušiť.

MOŽNOSTI ROZŠÍRENIA

Prioritne je potrebné riešenie bezpečnosti. Tu je nutné nasadiť SSL medzi Interface serverom a klientami.

Systém by sa ďalej dal rozšíriť deploynutím viacerých serverov a rozdenením záťaže medzi ne pomocou osobitného zariadenia. Ďalšia možnosť je oddeliť databázu od servra a používať oddelené servre.