

Analýza pro systém na rezervaci letenek

Karel Čemus, cvičení 9:15

1. Zadání

Pomocí webových služeb realizujte jednoduchý systém pro rezervaci letenek využívající 3 vrstvy serverů. V první vrstvě bude klient realizovaný jako aplikace nebo webová stránka. Druhou vrstvu bude tvořit Interface Server, který představuje jediné spojení klienta se třetí vrstvou, která je tvořena 3 servery. Booking Server se stará o práci s rezervacemi, Payment Server musí přijímat několik druhů plateb a Print Server umožňuje tisk různých dokumentů.

2. Funkcionální požadavky

1. Vypsát dostupné lety
2. Provést rezervaci na daný let
3. Stornovat rezervaci
4. Zaplatit rezervaci přes VISA kartu
5. Zaplatit rezervaci převodem částky z jiné stornované rezervace
6. Tisknout potvrzení o platbě, rezervaci a e-letenku

3. Architektura

Systém je navržen silně modulárně pro zajištění maximální testovatelnosti a minimálního opakování kódu, čímž lze dosáhnout výborné udržitelnosti a snadných aktualizací.

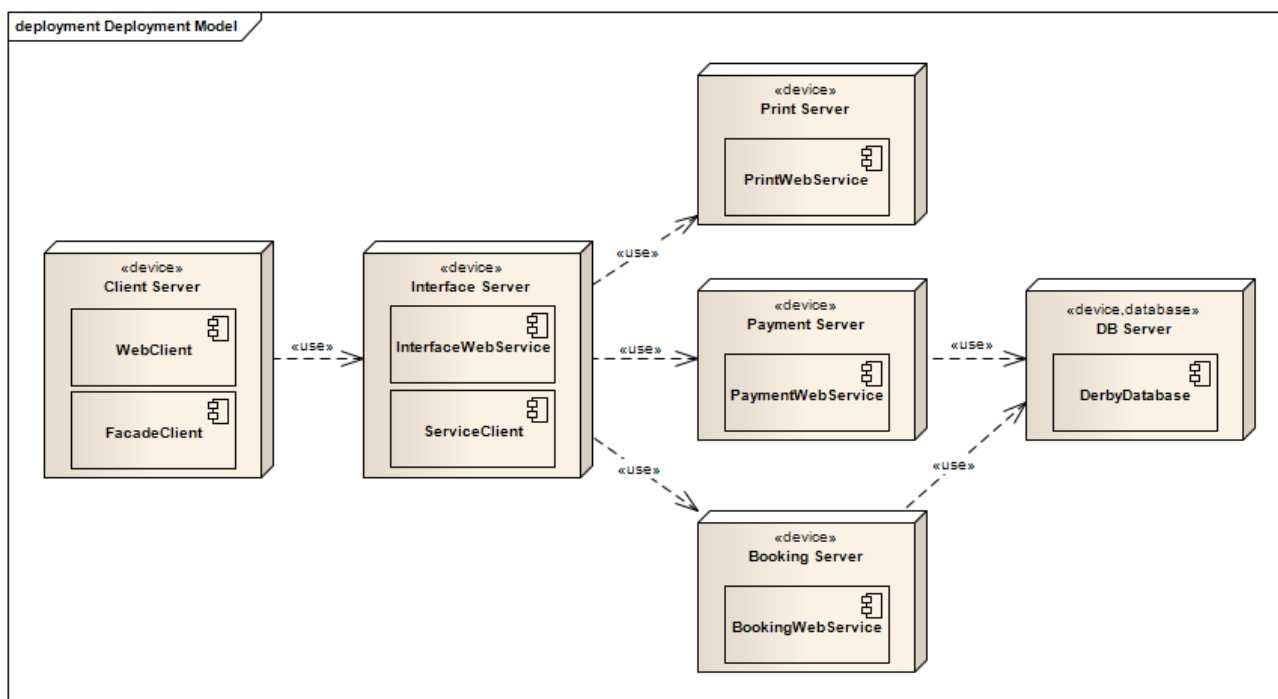


Diagram 1: Deployment diagram systému

Použité komponenty:

1. DB Server
 - persistentní úložiště dat
2. Print Server
 - umožní tisk různých dokumentů podle příchozích požadavků
 - není napojený na databázi, všechny informace dostává přes komunikační rozhraní
 - je stateless
3. Payment Server
 - přijímá požadavky na platby
 - pracuje transakčně
 - převedené peníze ukládá do databáze
4. Booking Server
 - obsahuje informace o letech a destinacích
 - přijímá požadavky pro práci s rezervacemi
 - pracuje transakčně
 - užívá persistentní úložiště
 - přístup k rezervaci je zabezpečen heslem, které je vygenerováno při vytvoření rezervace a je součástí potvrzení o rezervaci. Bez platného hesla není možné s rezervací pracovat
5. Service Client
 - implementuje klienta pro nabízené služby 3. vrstvy, umí volat vzdálená API
6. Interface Server
 - funguje jako fasáda pro služby 3. vrstvy
 - poskytuje API potřebné pro vlastní business aplikace
 - integruje služby 3. úrovně
 - deleguje požadavky na příslušné servery
 - obsahuje adaptér mezi objekty 2. a 3. vrstvy
7. Facade Client
 - implementuje klienta pro API 2. vrstvy, umí volat vzdálený server
8. Web Client
 - nabízí webové rozhraní pro práci se systémem

4. Technologie

Celý systém bude realizovaný pomocí SOAP webových služeb definovaných prostřednictvím WSDL specifikace. Pro implementaci budou použity následující technologie s přihlédnutím k reálným požadavkům zlehčeným o fakt, že se jedná o semestrální projekt.

- Databázový engine Derby, ačkoliv tato databáze má mnoho nedostatků jako špatná podpora constraints či transakcí, tak pro prezentaci funkčnosti to postačuje a díky použitému rozhraní JDBC pak při případném nasazení není komplikované provést výměnu databáze za kvalitnější engine, např. Postgres
- serverový kontejner Tomcat 7
- Hibernate 3 jako ORM provider

- poskytoval webových služeb Apache CXF
- integrační framework Spring 3 pro DI, bean a transaction management
- JSF 2.1 pro realizaci webového klienta
- TestNG pro implementaci testovacích skriptů pro ověření dostupnosti a funkčnosti webových služeb

5. Testovatelnost

Pomocí frameworku TestNG a jednotlivých implementovaných klientských modulů (FacadeClient, ServiceClient) bude vytvořena sada testovacích scénářů, které prověří chování vystavených webových služeb a ověří je proti specifikaci. Testy budou rozděleny pro 3. a 2. vrstvu, aby bylo možné najít chybu co nejbližší u zdroje a testovat tak každou vrstvu zvlášť.

6. Popis služeb

Print Server :: PrintService

- printReservationConfirmation
 - tiskne potvrzení o vytvoření rezervace
- printTicket
 - vytiskne e-letenku
- printPaymentConfirmation
 - generuje soubor s potvrzením o přijetí platby
- printCancelConfirmation
 - generuje potvrzení o stornování rezervace

Booking Server :: DestinationService

- findAllDestinations
 - vrací seznam všech známých lokalit

Booking Server :: FlightService

- find
 - hledá let podle jeho čísla
- findAllFlights
 - vrací všechny definované lety
- findFlightsFrom
 - vrací všechny lety z dané oblasti v požadovaném termínu
- findFlightsTo
 - vrací všechny lety do dané oblasti v požadovaném termínu
- findFlights
 - vrací všechny lety z místa A do místa B v požadovaném termínu

Booking Server :: ReservationService

- find
 - vrací rezervaci, pokud je předána platná kombinace číslo rezervace / heslo
 - SecurityException pokud není platné heslo k dané rezervaci
- create

- vytvoří rezervaci na let a vygeneruje přístupové heslo
- FullFlightException: Pokud je let již obsazen a není možné zarezervovat požadovaný počet míst
- cancel
 - stornuje rezervaci, pokud je předána platná kombinace číslo rezervace / heslo
 - SecurityException pokud není platné heslo k dané rezervaci

Payment Server :: PaymentService

- payVisa
 - přijímá definici platební karty a převádí peníze na cílovou rezervaci
 - pozn: platný verifikační kód odpovídá posledním 3 číslicím čísla karty
 - SecurityException pokud není platné heslo k dané rezervaci
 - InvalidPaymentException: platební karta je propadlá, není platný verifikační kód, rezervace je zrušená
 - NoSuchReservationException: Rezervace neexistuje
- transferFromReservation
 - přijímá identifikátor stornované a cílové rezervace a převádí mezi nimi vložené peníze
 - SecurityException pokud není platné heslo k dané rezervaci
 - NoSuchReservationException: Rezervace neexistuje
 - InvalidPaymentException: Zdrojová rezervace není zrušená, cílová je zrušená, zdrojový účet je prázdný

Interface Server :: FacadeService

- findAllDestinations
 - vrací seznam všech známých lokalit
- findFlightsFrom
 - vrací všechny lety z dané oblasti v požadovaném termínu
- findFlightsTo
 - vrací všechny lety do dané oblasti v požadovaném termínu
- findFlights
 - vrací všechny lety z místa A do místa B v požadovaném termínu
- findReservation
 - vrací rezervaci, pokud je předána platná kombinace číslo rezervace / heslo
 - SecurityException pokud není platné heslo k dané rezervaci
- createReservation
 - vytvoří rezervaci na let a vygeneruje přístupové heslo, vrací potvrzení o vytvoření rezervace
 - FullFlightException: Pokud je let již obsazen a není možné zarezervovat požadovaný počet míst
- cancelReservation
 - stornuje rezervaci, pokud je předána platná kombinace číslo rezervace / heslo, vrací potvrzení o stornování rezervace
 - SecurityException pokud není platné heslo k dané rezervaci

- NoSuchReservationException: Rezervace neexistuje
- payVisa
 - přijímá definici platební karty a převádí peníze na cílovou rezervaci, pokud jsou všechny údaje platné, vrací potvrzení o platbě
 - SecurityException pokud není platné heslo k dané rezervaci
 - InvalidPaymentException: platební karta je propadlá, není platný verifikační kód, rezervace je zrušená
 - NoSuchReservationException: Rezervace neexistuje
- payFromCanceledReservation
 - přijímá identifikátor stornované a cílové rezervace a převádí mezi nimi vložené peníze
 - SecurityException pokud není platné heslo k dané rezervaci
 - NoSuchReservationException: Rezervace neexistuje
 - InvalidPaymentException: Zdrojová rezervace není zrušená, cílová je zrušená, zdrojový účet je prázdný
- printTicket
 - tiskne e-letenku
 - SecurityException pokud není platné heslo k dané rezervaci
 - NoSuchReservationException: Rezervace neexistuje
 - ReservationNotPaidException: Rezervace není zaplacená, není možné tisknout letenku

7. Příklad komunikace

Následující sekvenční diagramy ukazují příklad komunikace v úspěšných scénářích. Pokud by v některé fázi došlo k vygenerování výjimky, byla by místo odpovědi propagovaná příslušná chybová zpráva (fault) viz specifikace jednotlivých web service.

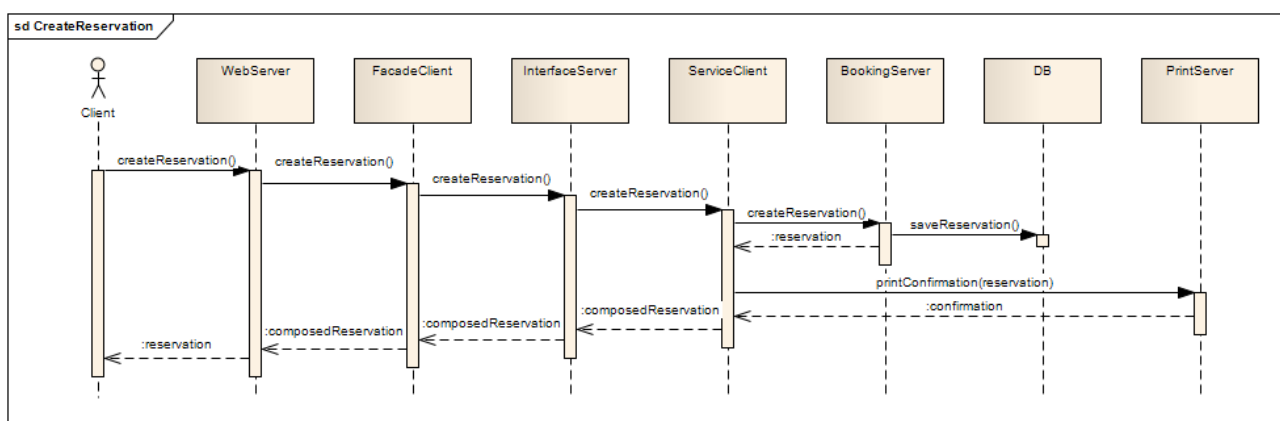


Diagram 2: Sekvenční diagram pro vytvoření rezervace

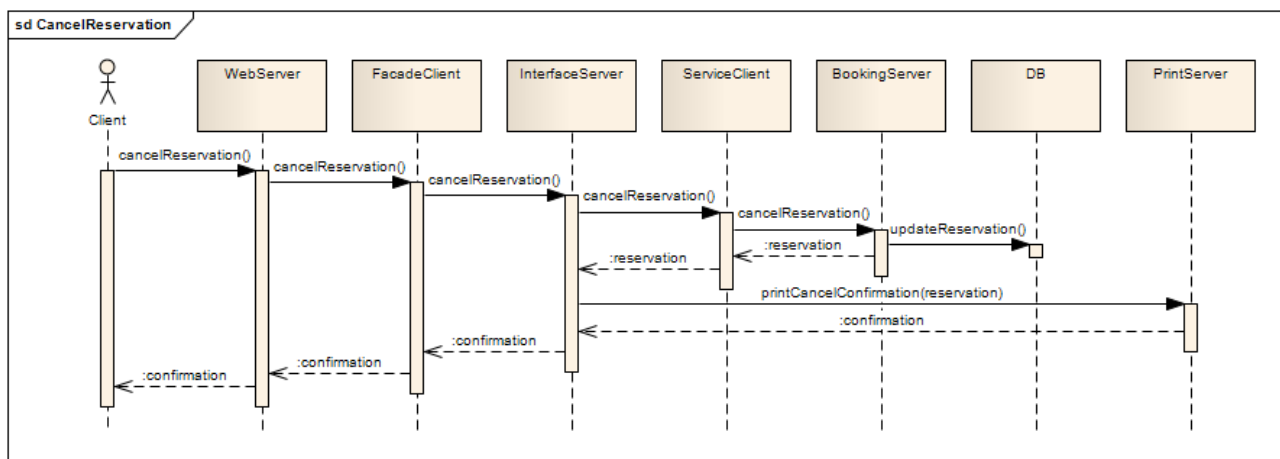


Diagram 3: Sekvenční diagram pro zrušení rezervace

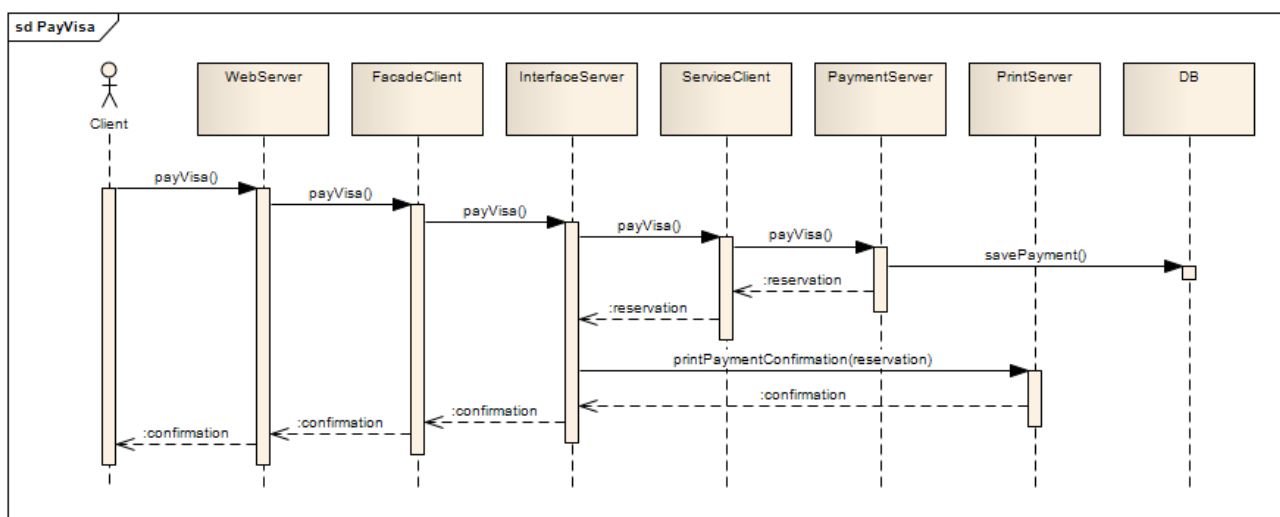


Diagram 4: Sekvenční diagram pro platbu VISA kartou

8. Implementační poznámky (není součástí analýzy)

- implementováno 5 serverů
- 26 testů pro servery 3. vrstvy
- 17 testů pro integrační facade server, testy jsou odvozeny od testů 3. úrovně a reflektují business operace
- testy kontrolují úspěšné i chybové průchody
- testy spoléhají na data v databázi, ale součástí testů jsou i disabled testy pro inicializaci databáze správnými daty
- přístup k rezervaci je hlídán heslem, které je vygenerováno při vytvoření rezervace. Heslo generuje web server, aby bylo možné do budoucna dopsat do UI možnost zadání hesla uživatelem, web service pouze očekávají heslo, které mají přiřadit.
- heslo se do databáze ukládá hashované a osolené
- web services pro práci s rezervací vyžadují heslo, ale klient JSF ho drží v session pro usnadnění práce uživatele

- transakce probíhají pouze nad databází na serverech 3. úrovně (ačkoliv vývojová DB Derby na to není vhodná, tak při výměně DB by to nebyl problém, řídí to Spring pomocí delegování)
- úspěšně předvedeno na cvičení