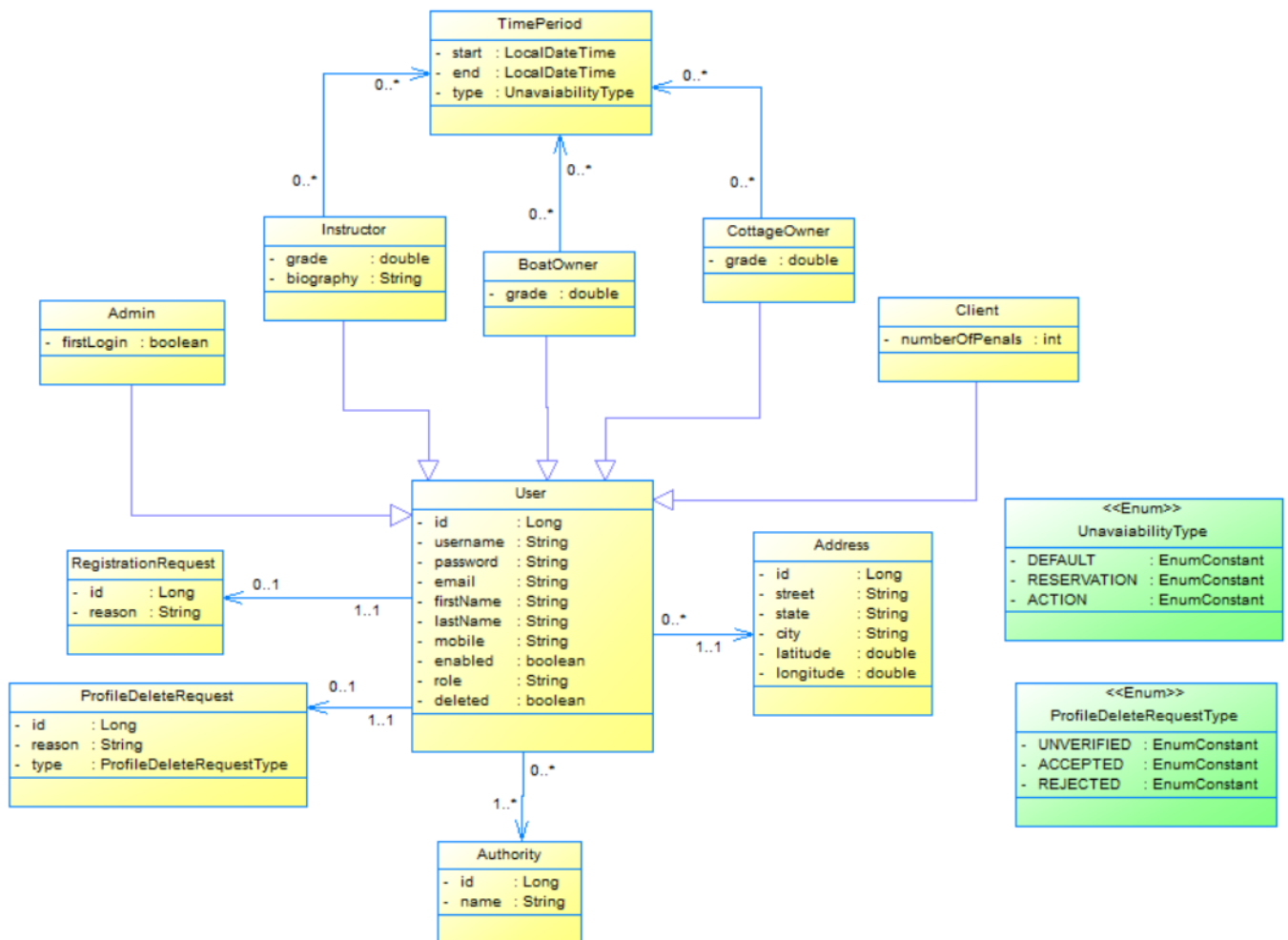


Skalabilna arhitektura aplikacije

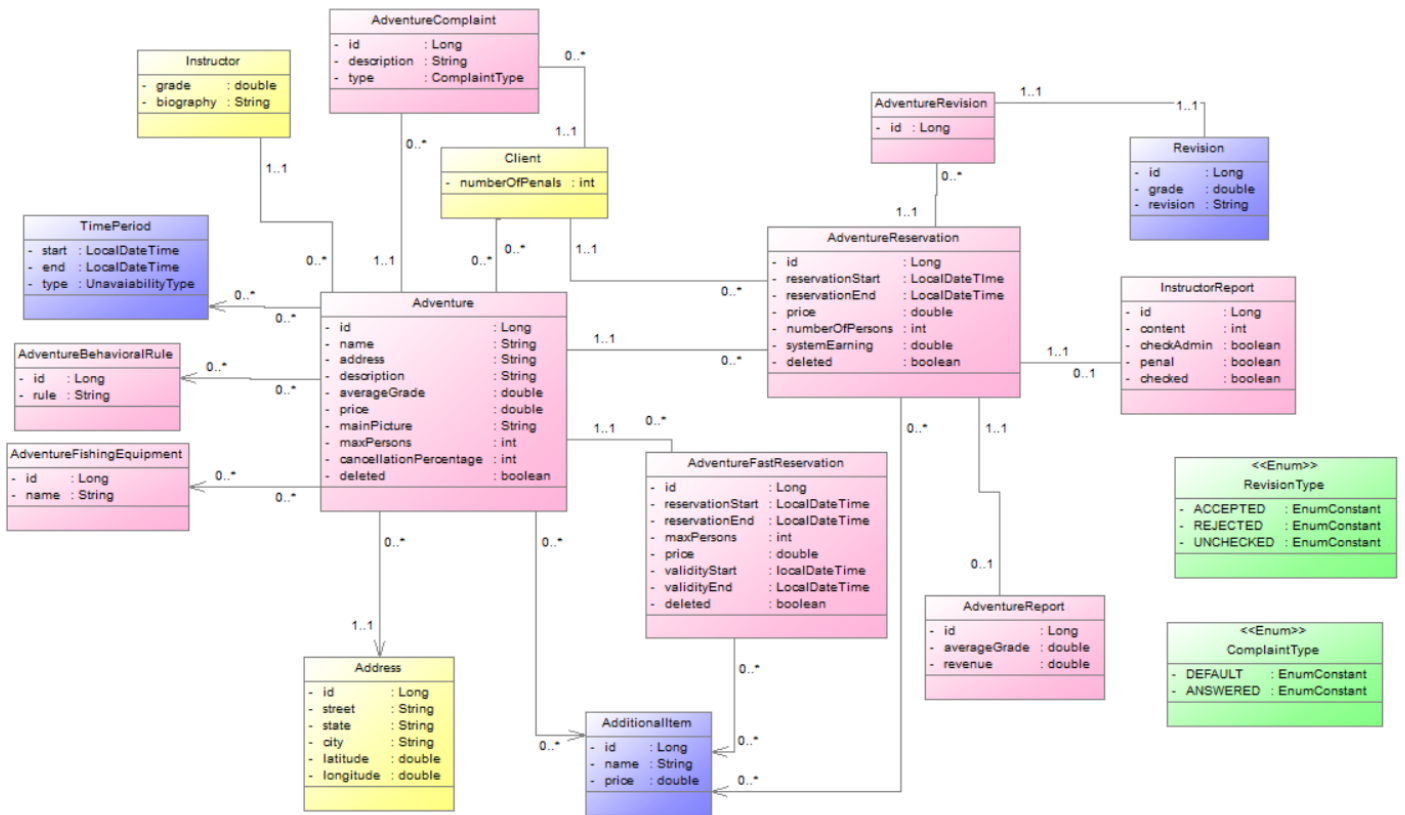
1. Dizajn šeme baze podataka

Dizajn šeme baze podataka je predstavljen putem klasnih dijagrama za slične i jako povezane entitete.

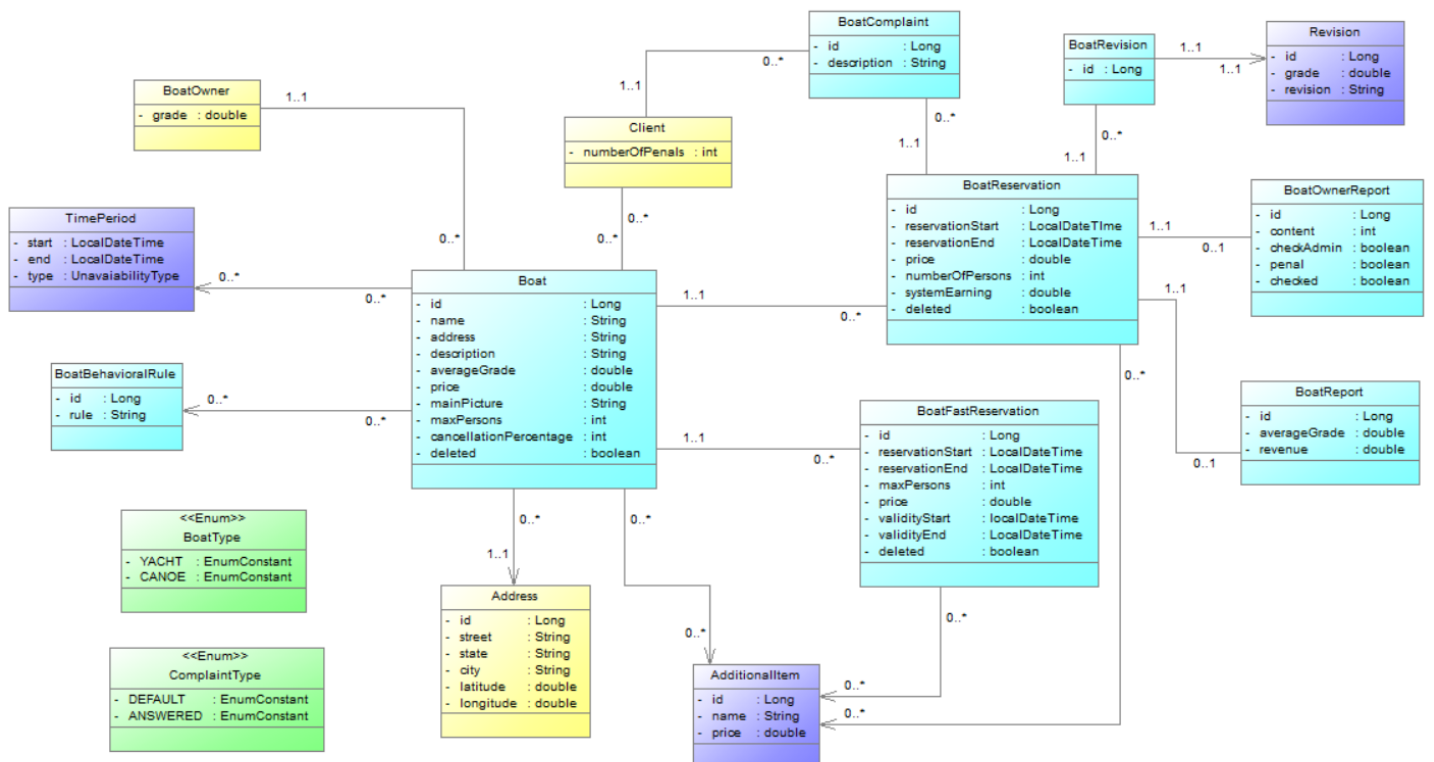
- Korisnici:



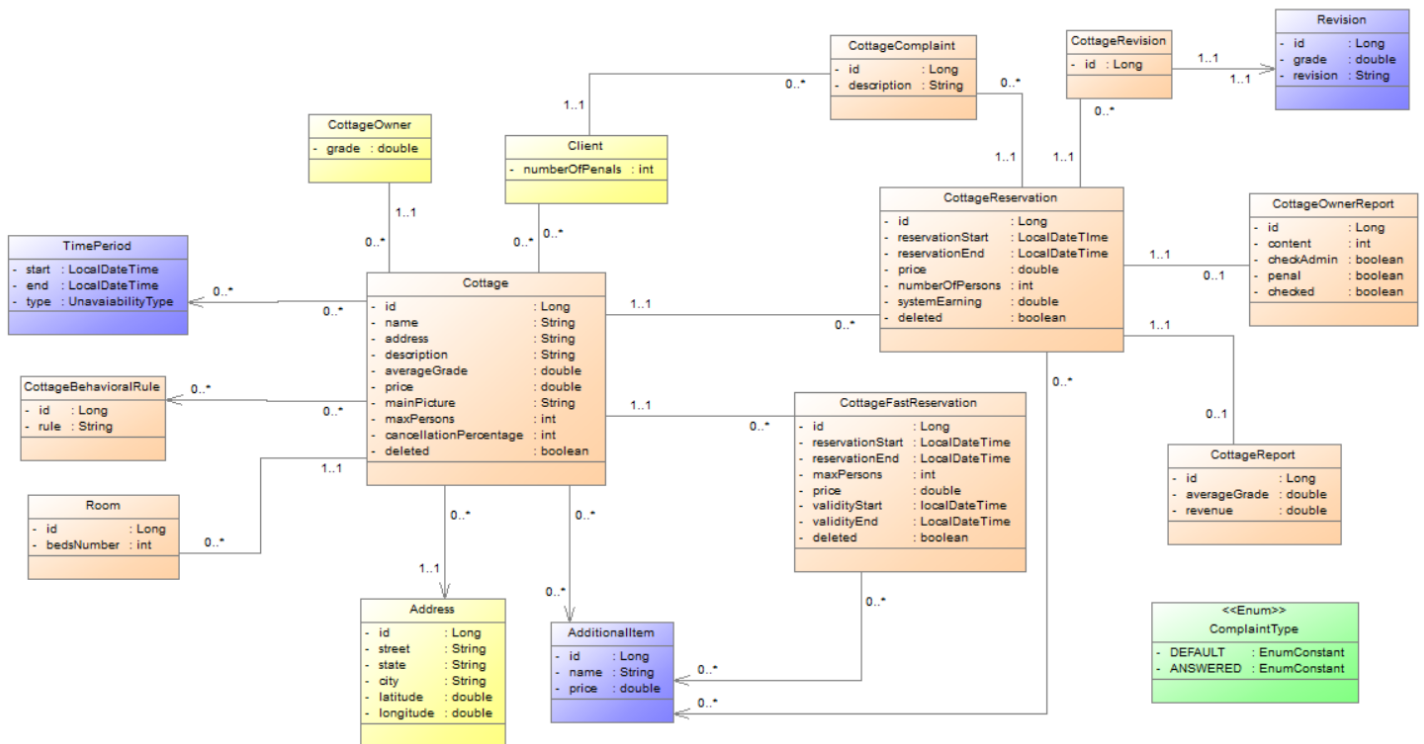
- Avantura:



- Brod:



- Vikendica:



2. Prijedlog za particionisanje podataka

Pod pretpostavkom da će ovu aplikaciju u budućnosti koristiti milioni korisnika, kako bi se postigle dobre performanse aplikacije potrebno je odraditi particionisanje podataka. Particionisanjem bi se podaci raspodijelili na više servera i time bi riješio problem preopterećenja servera zbog velike količine entiteta u bazi. Horizontalnim particionisanjem se tabele u bazi podataka dijele po redovima tj. nastaju odvojeni skupovi entiteta date tabele na osnovu određenog uslova(ključa). U slučaju naše aplikacije dato particionisanje možemo primjeniti nad tabelama za rezervacije vikendica, brodova i avantura zbog velike količine entiteta koja bi se tu čuvali na osnovu svakodnevnih rezervacija od strane korisnika. Particionisanje bi se moglo vršiti na osnovu ključa određene vikendice(broda ili instruktora avanture). Na osnovu ključa vlasnika bi se odredilo kojoj particiji odnosno serveru baze podataka se treba pristupiti.

Particionisanje prethodno navedenih tabela se može izvršiti i na osnovu datuma rezervacija, gdje bi se razdvojile završene od predstojećih rezervacija kojima bi se češće pristupilo. Isti način particionisanja bi se iskoristio i za akcije, gdje bi se

onima sa isteklim vremenom važenja pristupalo samo radi analiza pa bi se one smještale na hardver lošijeg kvaliteta. Tabelu sa klijentima particionisali bi na osnovu država iz koje dolaze.

Vertikalnim particionisanjem dijelimo tabelu na osnovu skupova određenih obilježja odnosno kolona. Tabele sa klijentima i vlasnicima možemo vertikalno particionisati tako da podatke kao što su username, password,email,.. čuvamo odvojene od ostalih podataka, jer su osjetljivi i pristupa im se rijede. S obzirom da se datumima rezervacija pristupa često, radi provjere zauzetosti i prikaza u kalendarima zauzeća, tabele sa rezervacijama se particionišu na navedene i ostale stavke. Entitete vikendica, broj i avantura dijelimo vertikalno na osnovne podatke i dodatne stavke o revizijama,akcijama,..

3. Prijedlog strategije za replikaciju baze i obezbjeđivanje otpornosti na greške

Kako bi se izbjegla prevelika opterećenost baze podataka sa velikim brojem korisnika, vrši se replikacija baze. Pristup koji bi bio iskorišćen je master-slave. On podrazumijeva da postoji jedna glavna (master) baza podataka i pomoćne (slave) koje su njene kopije. Kako bi cjelokupan sistem bio u konzistentnom stanju između svih baza se vrši sinhronizacija. Ukoliko se radi o upisu podataka, ono se izvršava na master bazi koja te podatke propagira na slave baze. Za razliku od upisa čitanje se odvija na svim serverima, čime se postižu bolje performanse i brži rad sa podacima. Proces replikacije može da se odvija sinhrono ili asinhrono.

U slučaju sinhronog u jednoj transakciji se obavlja ažuriranje podataka i na master i slave bazama, ali mana je što se proces zaustavlja ako je jedan server nedostupan. Iz tog razloga je izabran asinhron pristup kojim se šalje skup transakcija. Na ovaj način je omogućena otpornost na greške jer postoji mogućnost konfiguracije slave baza da preuzmu ulogu master baze ukoliko dođe do njenog otkaza. Time je obezbjeđeno očuvanje svih podataka. Kao dodatna mogućnost postoji i opcija klasterovanja korisnika na osnovu geografskog položaja, gdje bi imali više mašina za svaku grupu koje bi međusobno bile povezane i sinhronizovane. To bi dodatno omogućilo brži pristup podacima

4. Prijedlog strategije za keširanje podataka

Zarad većeg poboljšanja performansi aplikacije koristimo keširanje podataka, kojim se obezbjeđuje brže čitanje i dobavljanje podataka. Podaci koji su pogodni za keširanje su oni koji se ne ažuriraju često i koji nose veliku količinu informacija. U tom slučaju za našu aplikaciju bi bilo pogodno keširati podatke o glavnim entitetima kao što su informacije o klijentima ,vlasnicima, vikendicama,brodovima i avanturama, jer sadrže veliki broj obilježja koja su statička.

Također bilo bi poželjno keširati i podatke kao što su slike i druge vrste medija. Pored navedenih u keširane podatke bi smjestili i podatke koji se često dobijavaju odnosno samo čitaju. Tu opet pripadaju podaci o brodovima, avanturama i vikendicama i informacije o rezervacijama i periodima dostupnosti. U slučaju da se neki podaci moraju ažurirati, možemo ažurirati keš zapravo, jer je to jeftinija operacija u odnosu na one koje bi radile direktno sa bazom podataka.

U našoj aplikaciji implementirano je keširanje osnovnih podataka o avanturama, brodovima, vikendicama i periodima njihove zauzetosti. Za implementaciju je iskorišten EnCache.

5. Okvirna procjena za hardverske resurse potrebne za skladištenje svih podataka u narednih 5 godina

Osnovne pretpostavke:

- ukupan broj korisnika aplikacije je 100 miliona,
- broj rezervacija svih entiteta na mjesečnom nivou je milion
- sistem mora biti skalabilan i visoko dostupan

Podaci o korisnicima:

- Podaci o korisniku zauzimaju oko 1,7 KB
- Skladištenje svih korisnika će zauzeti oko: $1.7\text{KB} * 100\,000\,000 = \underline{170\text{GB}}$

Podaci o avanturama, brodovima i vikendicama:

- Za date entitete potrebno je oko 3 KB memorije
- Pretpostavka da od 100 000 000 korisnika njih 5% spada u vlasnike i da svaki vlasnik posjeduje u prosjeku po 2 entiteta
- Ukupno zauzeće bi bilo $100\,000\,000 * 0,05 * 2 * 3\text{KB} = \underline{30\text{GB}}$

Podaci o rezervacijama:

- Obična rezervacija zauzima oko 2.8 KB, a akcija oko 8KB
- Pretpostavka je da imamo oko 1 million rezervacija na mjesečnom nivou, a oko 200 000 definisanih akcija mjesečno
- Skladištenje rezervacija i akcija nakon 5 godina zauzima oko $1\,000\,000 * 12 * 5 * 2.8\text{KB} + 200\,000 * 12 * 5 * 8\text{KB} = 168\text{GB} + 96\text{GB} = \underline{264\text{GB}}$

Podaci o ocjenama i izvještajima:

- Za čuvanje ocjena od clijenata je potrebno oko 2KB, a za čuvanje izvještaja od strane vlasnika oko 8.2KB
- Pretpostavka da je od svih rezervacija 70% završeno i da će na 50 % njih klijenti dati ocjenu
- Pretpostavka da će instruktori ostaviti izvještaj na 10% završenih rezervacija

- Ukupna potrošnja memorije nakon 5 godina bi bila: $1\,000\,000 * 12 * 5 * 0,7 * 0,5 * 2\text{KB} + 1\,000\,000 * 12 * 5 * 0,7 * 0,1 * 8,2\text{KB} = 42\text{GB} + 34,4\text{GB} = \underline{76,4\text{GB}}$

Ostali podaci:

- Pretpostavka da će za ostale podatke u toku 5 godina biti potrebno oko 20GB

Za održavanje sistema u toku 5 godina potrebno je oko 550GB.

6. Prijedlog strategije za postavljanje load balansera

Cilj korišćenja load balansera jeste obezbjeđenje ravnomjerne raspodjele poslova na serverima. On bi se postavio između klijenta i aplikativnog servera tako da se korisnički zahtjevi šalju onom serveru koji je trenutno dostupan i predložen. Algoritam koji bi se iskoristio je "Weighted Least Connection". Na osnovu datog algoritma balanser bi uputio zahtjev na server koji ima najmanje aktivnih konekcija. Prednost je i mogućnost definisanje težina odnosno značajnosti svakog servera pri čemu bi se moglo podesiti da serveri sa boljim performansama imaju veću vjerovatnoću prihvatanja zahtjeva od korisnika.

7. Prijedlog koje operacije korisnika treba nadgledati u cilju poboljšanja sistema

Zarad boljih performansi date aplikacije kao i korisničkog iskustva predlaže se uvođenje Event Sourcing-a. Na osnovu bilježenja podatka o broju rezervacija za svaki entitet i drugih statistika, klijentima bi se mogle ponuditi opcije poput najpopularnijih avantura, brodova i vikendica kao i preporuka događaja koji su im u blizini. Također na osnovu pretrage klijenata i klikova, svaki vlasnik bi mogao imati uvid u podatke o tome koja usluga mu se najviše isplati i u šta treba dodatno da ulaže.

Na osnovu statistika vlasnicima bi se mogli preporučivati periodi pogodni za definisanje akcija kao i vrijednosti njihovog popusta.

Pored navedenog radi boljeg korisničkog iskustva trebale bi se uvesti i notifikacije o predstojećim rezervacijama kao i podsjetnici za davanje ocjene i izvještaja.

Za sam rad aplikacije analitike o podacima bi bile korisne i za donošenje odluke koje podatke bi trebalo keširati ili izbaciti iz upotrebe.

8. Kompletan crtež dizajna predložene arhitekture (aplikativni serveri, serveri baza, serveri za keširanje, itd)

