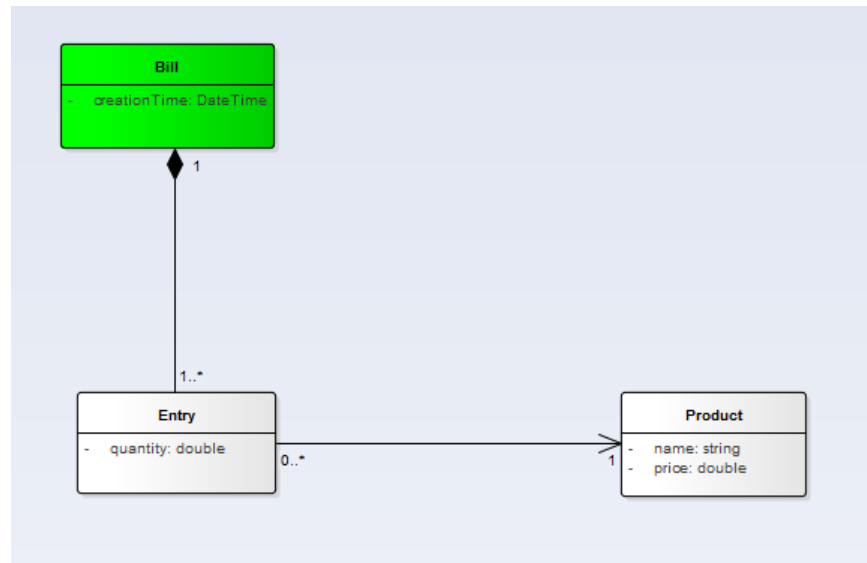


# Cloud Computing u elektroenergetskim sistemima

## Opis projekta

U okviru dijagrama klasa je definisano nekoliko klasa. Svaka klasa odgovara entitetu koji je potrebno čuvati u Azure Table skladištu. Za svaki entitet je potrebno sačuvati sve podatke koje taj entitet potencijalno može da ima (minimalno 2 atributa po klasi, atributi se definišu samostalno ukoliko ih ima manje od 2). Veze između klasa, ograničenje referencijalnog integriteta ne treba implementirati.



Postoje dva tipa entiteta: Primarni entiteti (zelene klase) i sekundarni entiteti (ostale klase). Korisnik može da napravi, izmeni, izlista, i obriše i primarne i sekundarne entitete (CRUD) operacije. Kroz upotrebu Web role omogućiti klijentu da kroz korisnički interfejs (Web stranice) može da manipuliše definisanim entitetima, odnosno da se omogući dodavanje, izmena, brisanje i ispis entiteta.

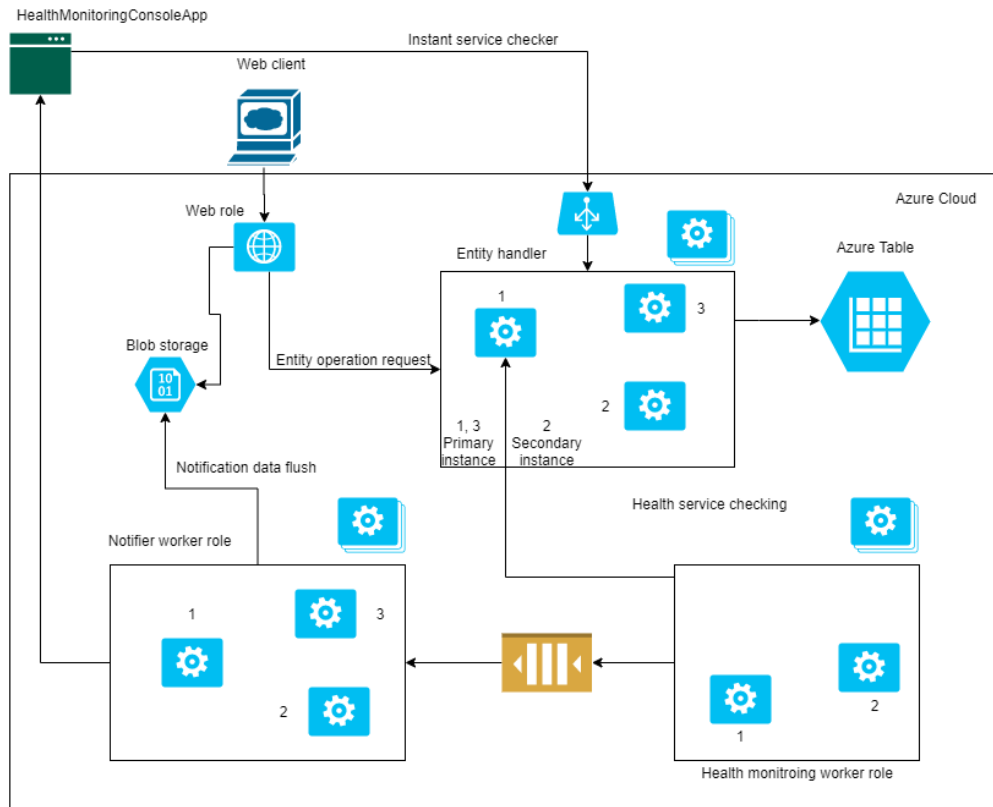
## Arhitektura

Potrebno je u skladu sa datim entitetima implementirati distribuirani sistem prikazan na dijagramu.

Web rola ne komunicira direktno sa Azure Table skladištem već šalje odgovarajući zahtev Entity handler worker roli koja preuzima komunikaciju sa Azure Table skladištem.

Potrebno je u okviru worker role (Entity handler worker role) implementirati interfejs za web role za manipulaciju sa primarnim i sekundarnim entitetima.

Primarni entiteti se šalju primarnim instancama Entity handler worker roli, dok se sekundarni entiteti šalju sekundarnim instancama, **naznačeno na dijagramu**.



## Health Monitoring

Potrebno je uz dati sistem za manipulisanje sa entitetima da se implementira i odgovarajući Health Monitoring sistem. Za ove potrebe kreirati Worker rolu (Health monitoring worker role) koja propituje na određenom intervalu Entity handler instancu worker role i nakon uspešnog ili neuspešnog odgovora, stavlja odgovor u Azure Queue koji prati/osluškuje Notifier worker rola. Nakon što dobije podatke iz Queue-a, Notifier worker rola upisuje informaciju u Azure Blob Storage u **blob** za odgovarajući dan. Pored ovoga, tu informaciju i šalje konzolnoj aplikaciji (HealthMonitoringConsoleApp) koja ispisuje trenutni status instance.

Konzolna aplikacija (HealthMonitoringConsoleApp) je u stanju da samostalno propita bilo koju instancu Entity handler-a pri čemu ispisuje status instance.

Potrebno je prilikom obrade zahteva u Web roli proveravati stanje instance, pročitati informaciju iz bloba za odgovarajući dan. Ukoliko je instanca nedostupna, potrebno je zahteve čuvati u blobu. Periodično proveravati da li je instanca dostupna, ukoliko jeste svi sačuvani zahtevi da se obrade.

## Dokumentacija

Uz projekat je potrebno dostaviti i projektnu dokumentaciju. Dokumentacija treba da sadrži sledeće elemente:

- Listing glavnih delova implementacije čuvanja primarne instance
- Listing implementacije komunikacije Write instanci i Logger-a
- Navesti nazive resursa u Cloud-u (nazive tabela, kontejnera i sl.), parametri u okviru Queue-a

- Opisati nekoliko test slučajeva koji se mogu izvršiti kako bi se svi aspekti sistema pokrili i testirali korak po korak npr jedan test za primarne instance:
  - Kakav zahtev se šalje
  - Koja instanca i zašto preuzima zahtev
  - Dalji koraci...

## Bodovanje

Web klijent (web stranice, kontroleri)	10 bodova
EntityHandler CRUD operacije	15 bodova
HealthServiceChecking/InstanceServiceChecking	10 bodova
NotifierWorkerRole Queue komunikacija	10 bodova
NotifierWorkerRole notifikacija konzolnoj aplikaciji	10 bodova
NotifierWorkerRole upis u blob	5 bodova
Provera dostupnosti instance i čuvanje zahteva u blob	10 bodova