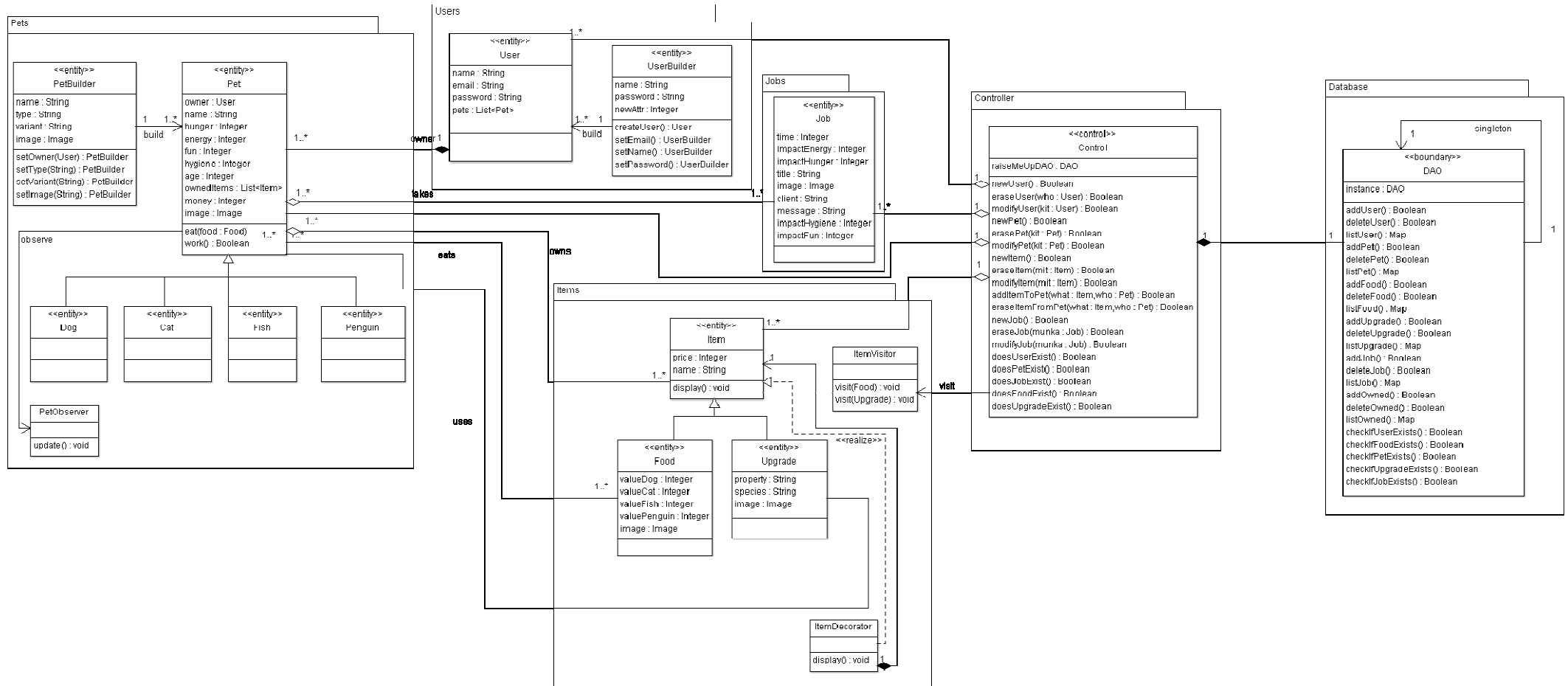


## Tervezési minták

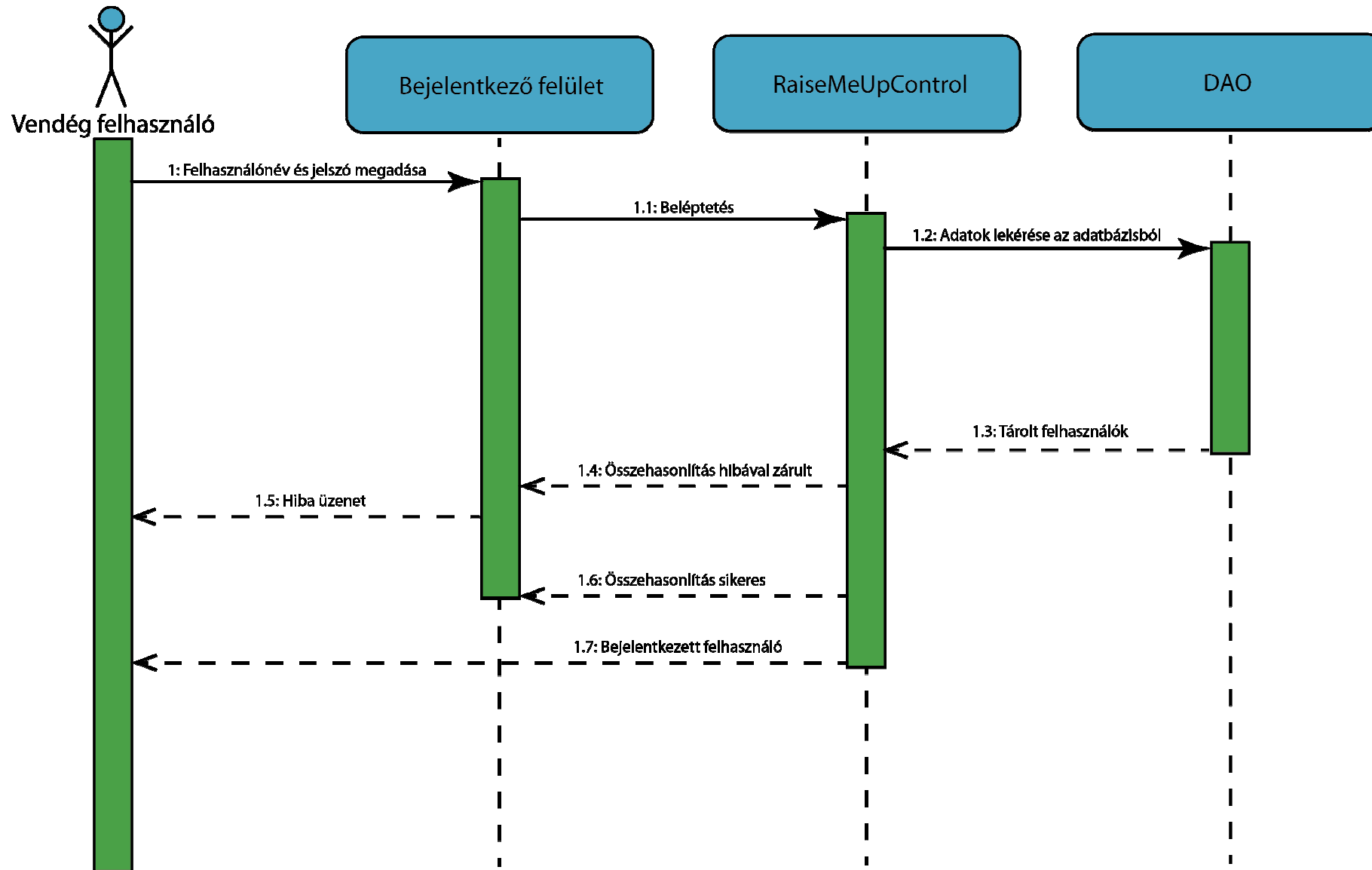
- **BUILDER** - a Builder elkészült a felhasználóhoz (UserBuilder) és a házi kedvenchez (PetBuilder).
- **DECORATOR** - ItemDecoratort fogunk tenni a kedvencek által birtokolható eszközökre, tehát az ételhez (Food) és fejlesztésekhez (Upgrade). Az alapmegjelenítésen csak a házi kedvenc fog szerepelni és a további bővítésekkel kerülnek rá a panelre a Food, illetve Upgrade elemek.
- **VISITOR** - lesz lehetőség boltban vásárolni a kedvencek számára, a boltban található termékeket fogjuk ItemVisitor-ral "meglátogatni" és kilistázni, az elemek vásárlása előtt.
- **SINGLETON** - adatbázisban van tárolva minden, tehát a játékhoz szükséges állapotokat is ott tároljuk le. Maga a DAO lesz Singleton.
- **OBSERVER** - a kedvencfigyelő felületen meg lesz jelenítve többek között a házi kedvenc és az, hogy éppen milyen "állapotban" van, értve ezalatt az örömet, az energiát, a higiénia és az éhség szintjét. Ahogy telik az idő ezek az értékek folyamatosan csökkennek és dinamikusán fogjuk frissíteni PetObserver segítségével a megjelenített progressBar-okon az állapotot (%-ban) és frissíteni az adatbázisban is ezeket az értékeket (ezzel biztosítva a folyamatos mentést).

### A refaktorált Class diagram:

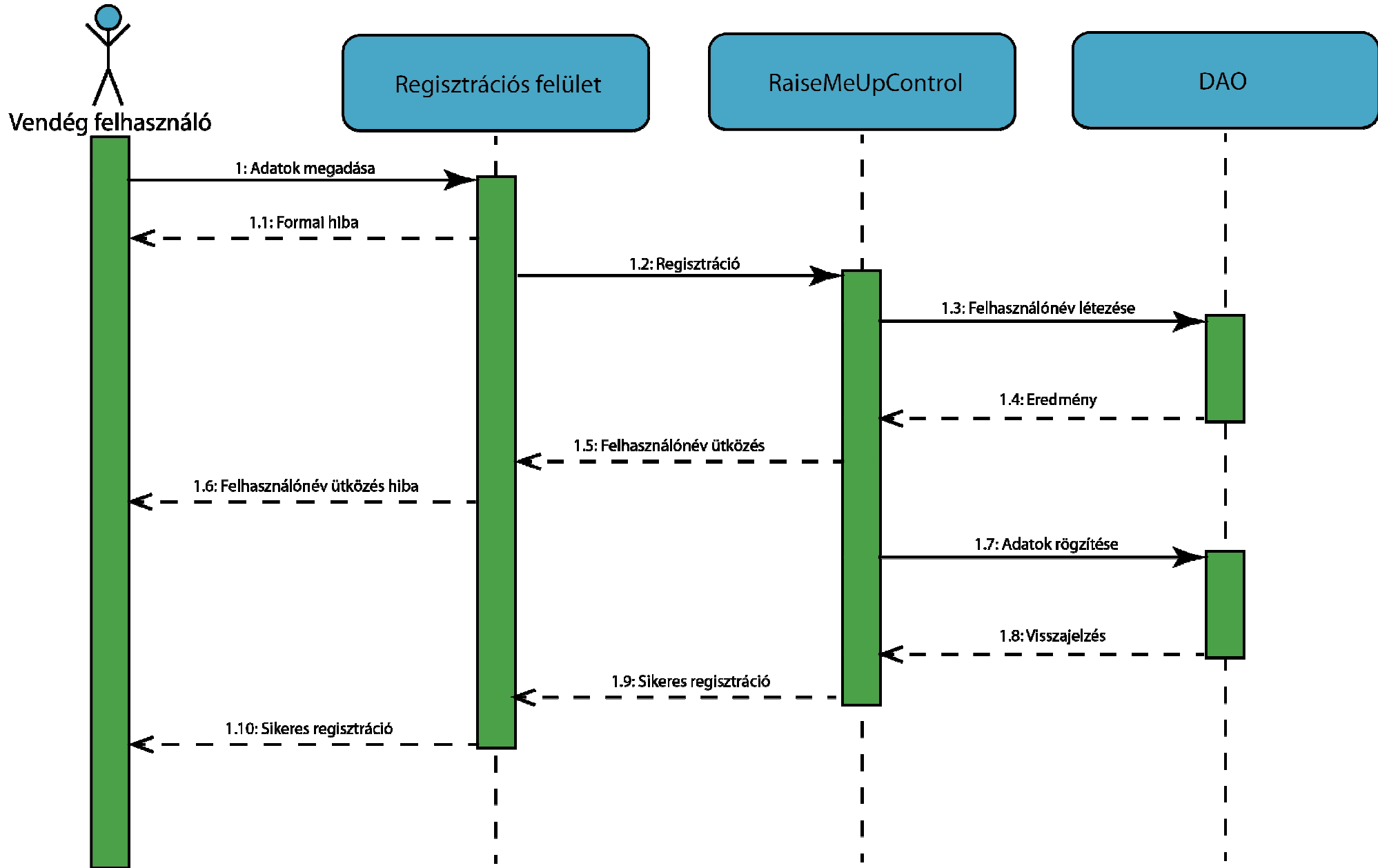


## Sequence diagramok:

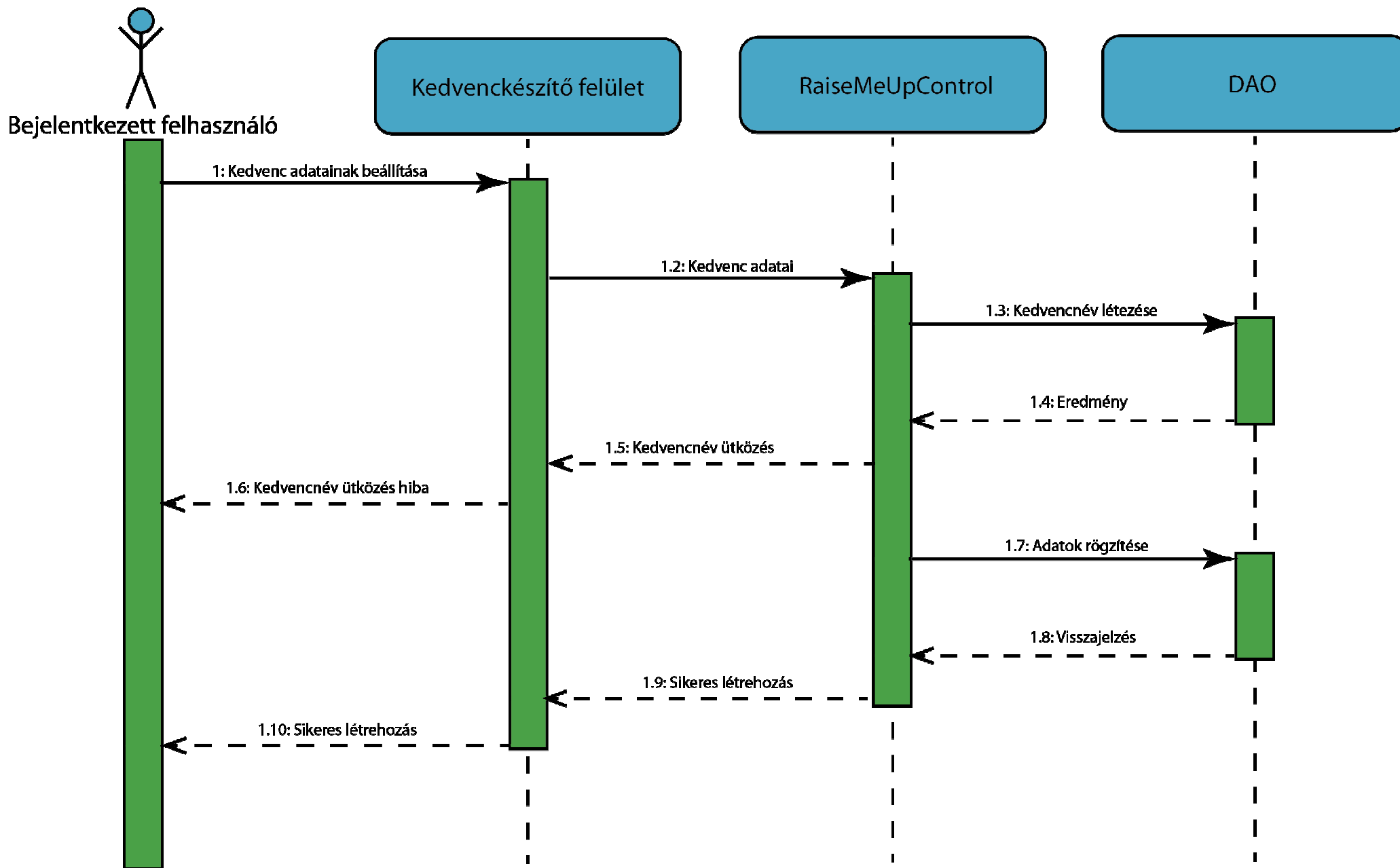
### Bejelentkezés:



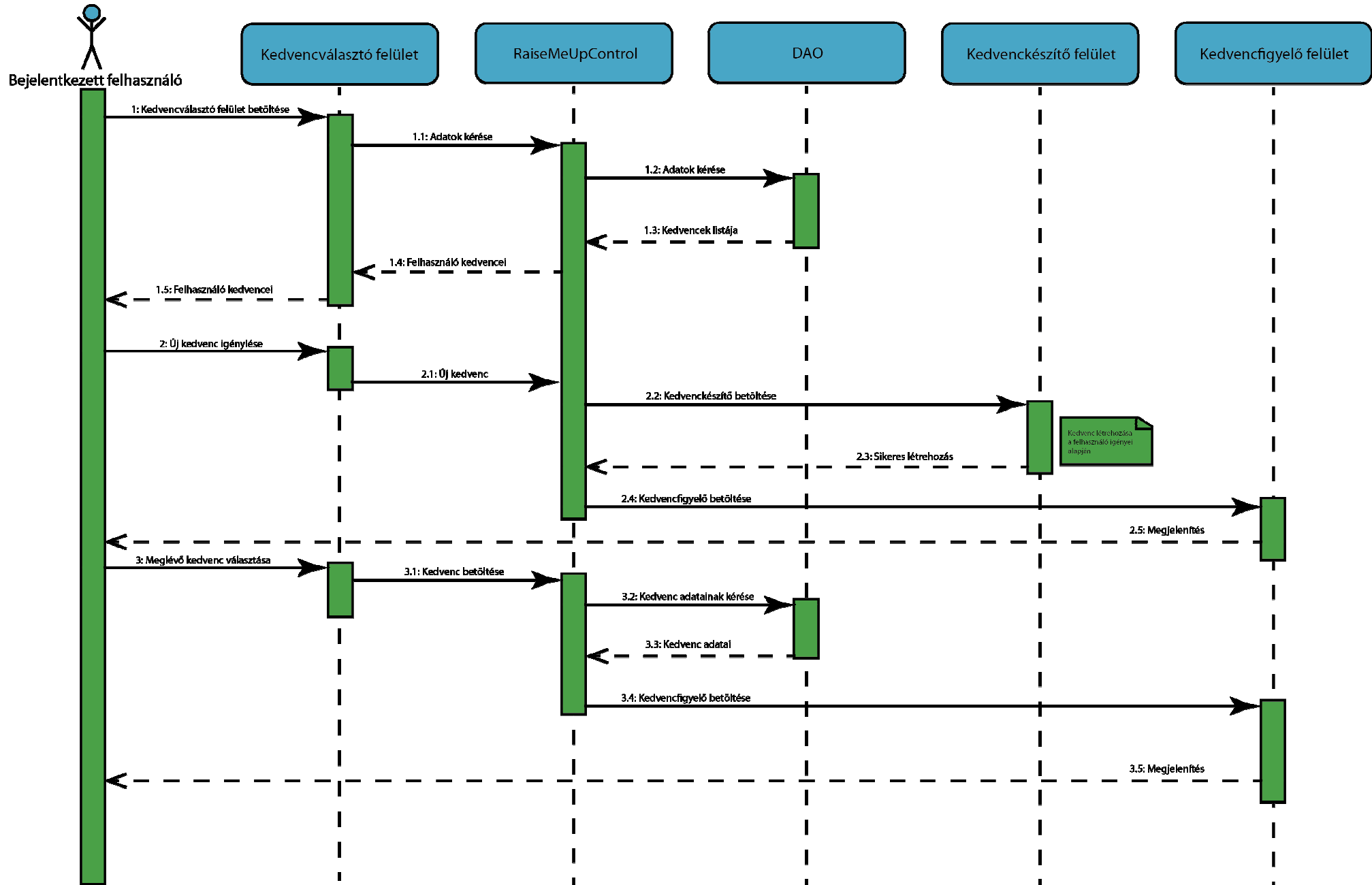
## Regisztráció:



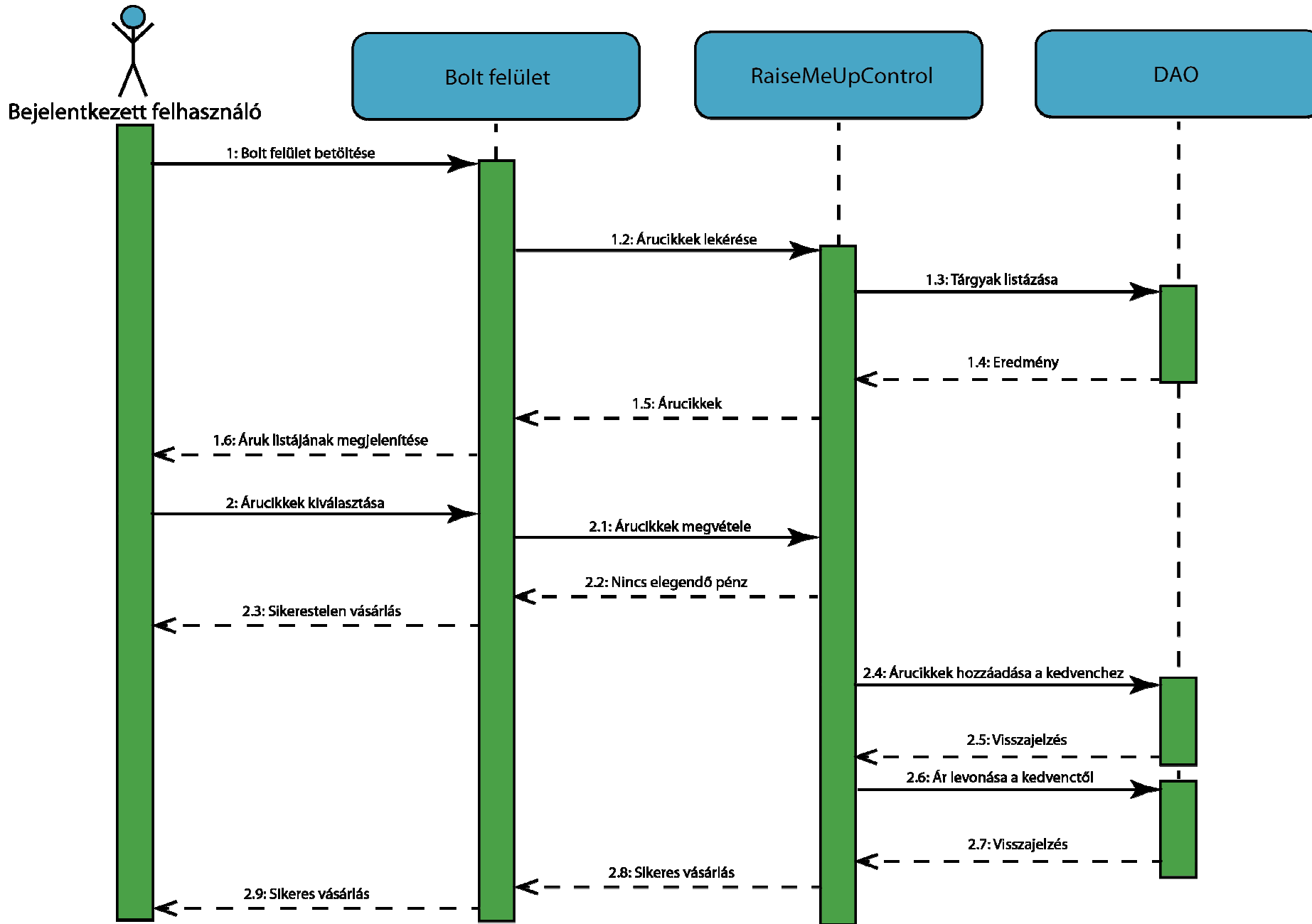
## Kedvenckészítés:



## Kedvencválasztás:



## Vásárlás:



### Deployment diagram:

