

PROIEKTUAREN DOKUMENTAZIOA

Egileak:

Urtzi Espinosa
Unax Amunarriz
Zuriñe Behobide

AURKIBIDEA

• Web Zerbitzua eta Internalizazioa.....	3
• Sarrera.....	3
• Eskakizun bilketa eta diseinua.....	3
○ Entitateak.....	3
○ Domeinuaren eredua eta bere erlazioak.....	4
○ Erabilpen kasuen eredua.....	6
○ Erabilpen kasu bakoitzeko bere gertaera fluxua eta interfaze grafikoa.....	7
○ Klase diagrama osoa.....	35
• Inplementazioa.....	38
• Ondorioak.....	42
• Bideoaren URL-a.....	42
• Kodearen zip fitxategia.....	42

SOFTWARE INGENIARITZA

- **Web Zerbitzua eta Internalizazioa**

Internalizazioa garatu dugu, web zerbitzuak ez.

- **Sarrera**

Hiru mailako software arkitektura batean diseinatutako bidaiak konpartitzeko aplikazio bat garatu dugu.

Aplikazio honetan erabiltzaile erregistratuentzako hainbat funtzionalitate agertzen dira, baina honako funtzio nagusi hauek azpimarratuko ditugu:

- Bidaiak sortzea.
- Bidaiak erreserbatzea.
- Bidaiak bilatzea.

- **Eskakizun bilketa eta diseinua**

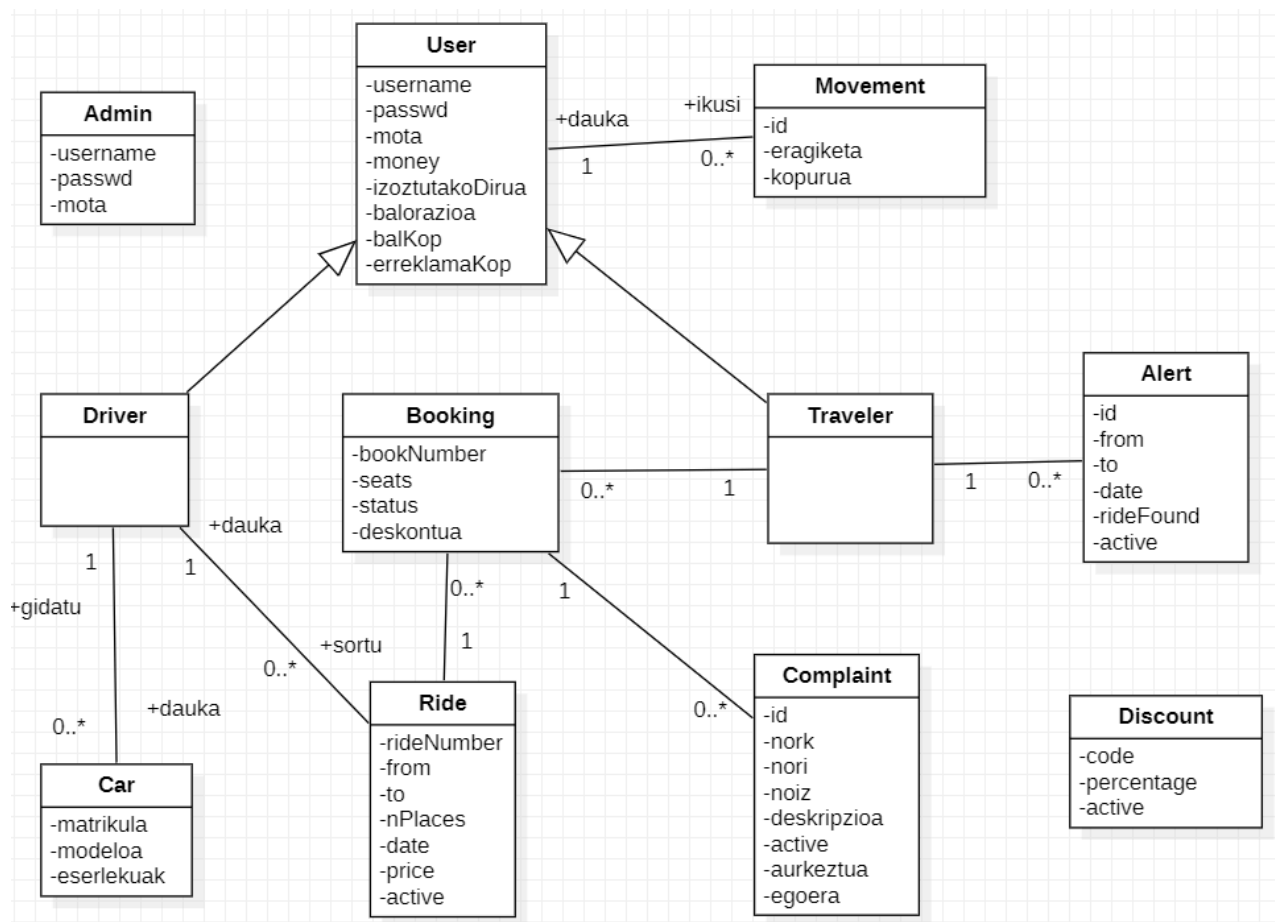
- Entitateak

Guztira lau entitate definitu ditugu: *Ez-erregistratua*, *User*, *Traveler* eta *Driver*.

Lehenengo entitatea, aplikazioan erregistratu ez diren erabiltzaileei dagokio. Entitate hau ez dago datu basuan gordeta, beraz, aplikazioaren barruan ez du inongo eraginik egingo erregistratua izan arte. Beraz, hiru entitate nagusiak *User*, *Traveler* eta *Driver* dira. *Traveler* eta *Driver* klaseek atributu berdinak konpartitzen dituztenez, *User* klasearen umeak dira eta hainbat funtzionalitate konpartitzen dituzte.

SOFTWARE INGENIARITZA

- Domeinuaren eredua eta bere erlazioak



Diseinatu dugun domeinuaren ereduan, 11 klase definitu ditugu: *Admin*, *User*, *Movement*, *Driver*, *Booking*, *Traveler*, *Alert*, *Car*, *Ride*, *Complaint* eta *Discount*.

User klaseak erabiltzaile erregistratuak biltzen ditu eta bi klase ume dauzka: *Driver* eta *Traveler*. *User* klasea definitu dugu aurreko bi klase hauek atributu berdinak konpartitzen dituztelako. *User* klaseak *Movement* klaseko lista bat dauku, bertan, diruarekin egindako mugimendu eragiketa guztiak agertuko dira eta *Movement* klase bakoitza *User* bakar bati lotuta egongo da.

Driver klaseak *Car* motako lista bat dauka eta *Car* bakoitzak *Driver* bakarra gidatzeko aukera izango du. Horretaz gain, *Driverrak* sortutako bidaien lista bat dauka, *Ride* izenarekin definitutakoa.

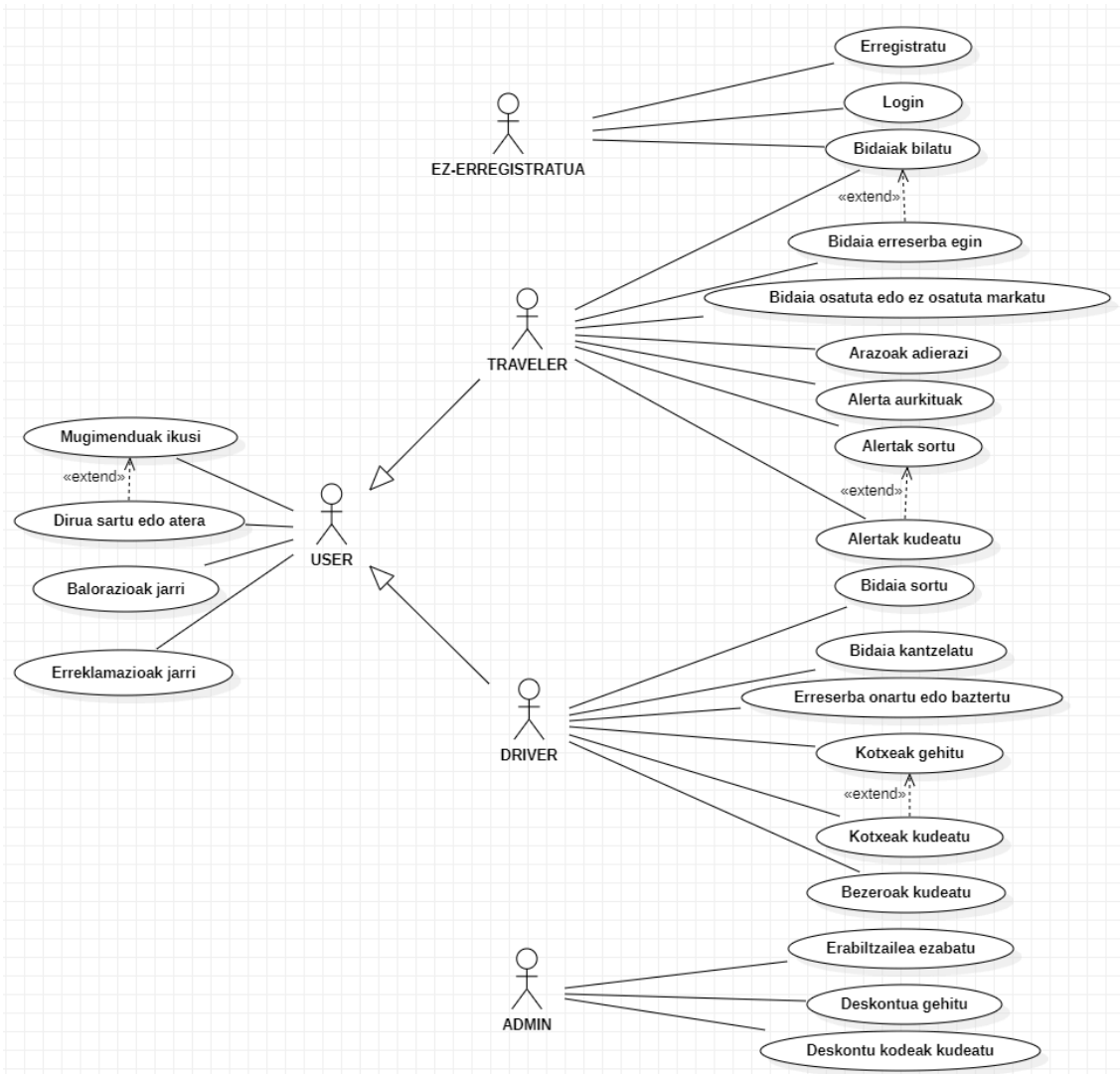
SOFTWARE INGENIARITZA

Traveler klaseak hainbat *Alert* kudeatzeko lista bat dauka eta *Alert* bakoitza *Traveler* bakar bati lotuta dago. *Traveler* klaseak baita *Booking* klaseari lotuta dago, erreserbatutako bidaia guztiak gordetzeko. *Booking* klase honek beste bi klaseekin lotuta dago. Alde batetik, *Ride* klasearekin, erreserba bakoitza bidaia bati esleituta egoteko eta *Ride* batek hainbat erreserba desberdinak edukitzeko. Bestetik, *Booking* bakar bat hainbat erreklamazio eduki ditzakeenez, *Complaint* klaseari lotuta dago.

Azkenik, loturik gabeko bi klase gelditzen dira: *Admin* eta *Discount*. *Admin* klaseak aplikazio barruan administratzaile bezala jokatzeko du eta funtzio nagusiak jasaten ditu. *Discount* klasea, *Travelerrek* bidaia baten erreserbaren prezioari aplikatu diezaioketen deskontua da.

SOFTWARE INGENIARITZA

○ Erabilpen kasuen eredua



SOFTWARE INGENIARITZA

- Erabilpen kasu bakoitzeko bere gertaera fluxua eta interfaze grafikoa

Erregistratu

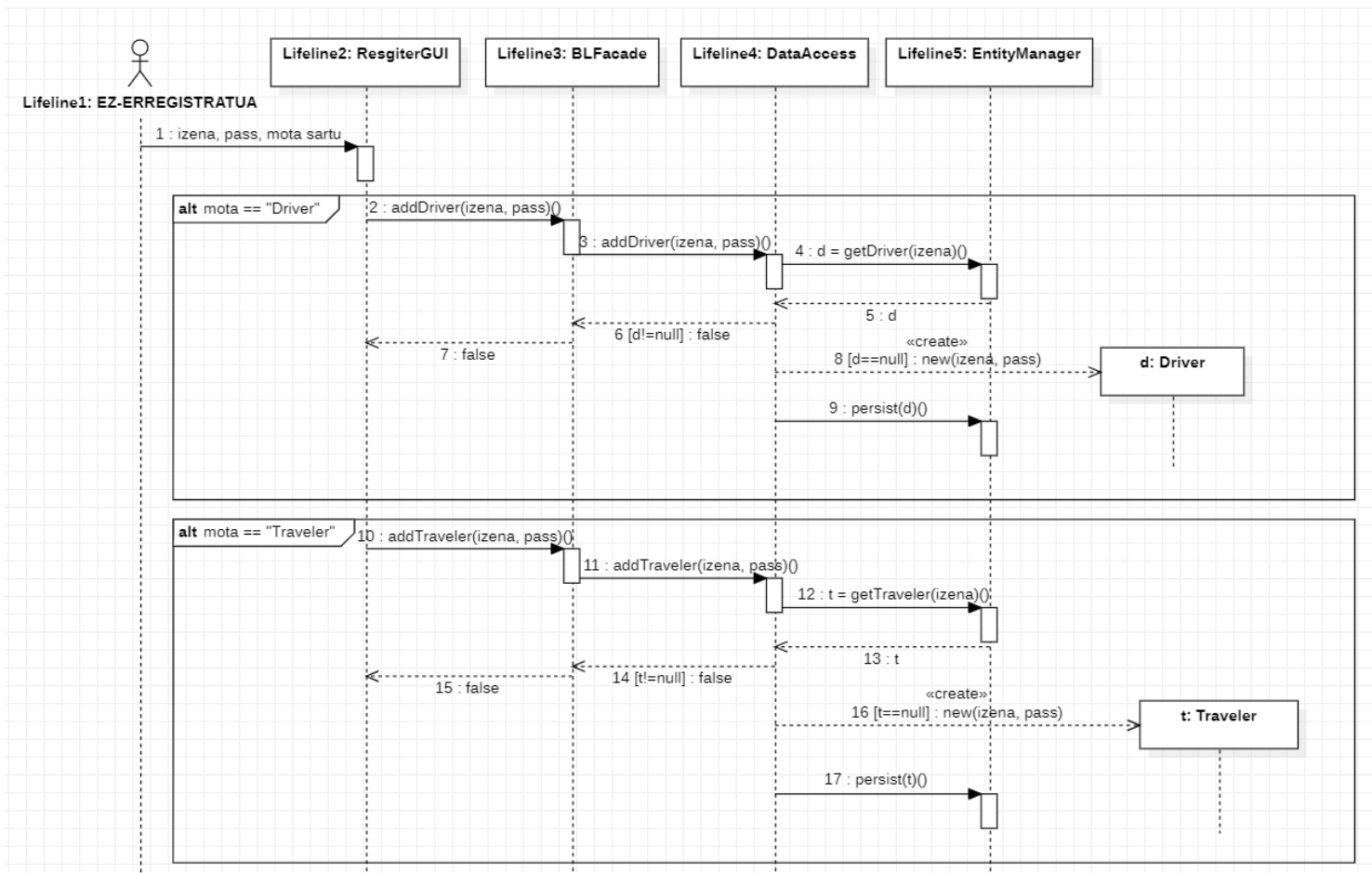
Gertaera fluxuak

Basic flow

1. Ez-erregistratua izena eta pasahitza sartuz, bidaiaria edo gidaria aukeratu eta sartu.
2. Sistemak aukeratutako rollaren arabera erabiltzaile hori dagoeneko existitzen den frogatzen du. Ez bada existitzen erabiltzailerik, kontu bat sortzen du.

Alternative flow

1. Datuak falta dira. Ez da erregistratzen. Sistemak erabiltzailea abisatzen du.
2. Dagoeneko erabiltzailea existitzen da. Sistemak erabiltzailea abisatzen du.



SOFTWARE INGENIARITZA

Login

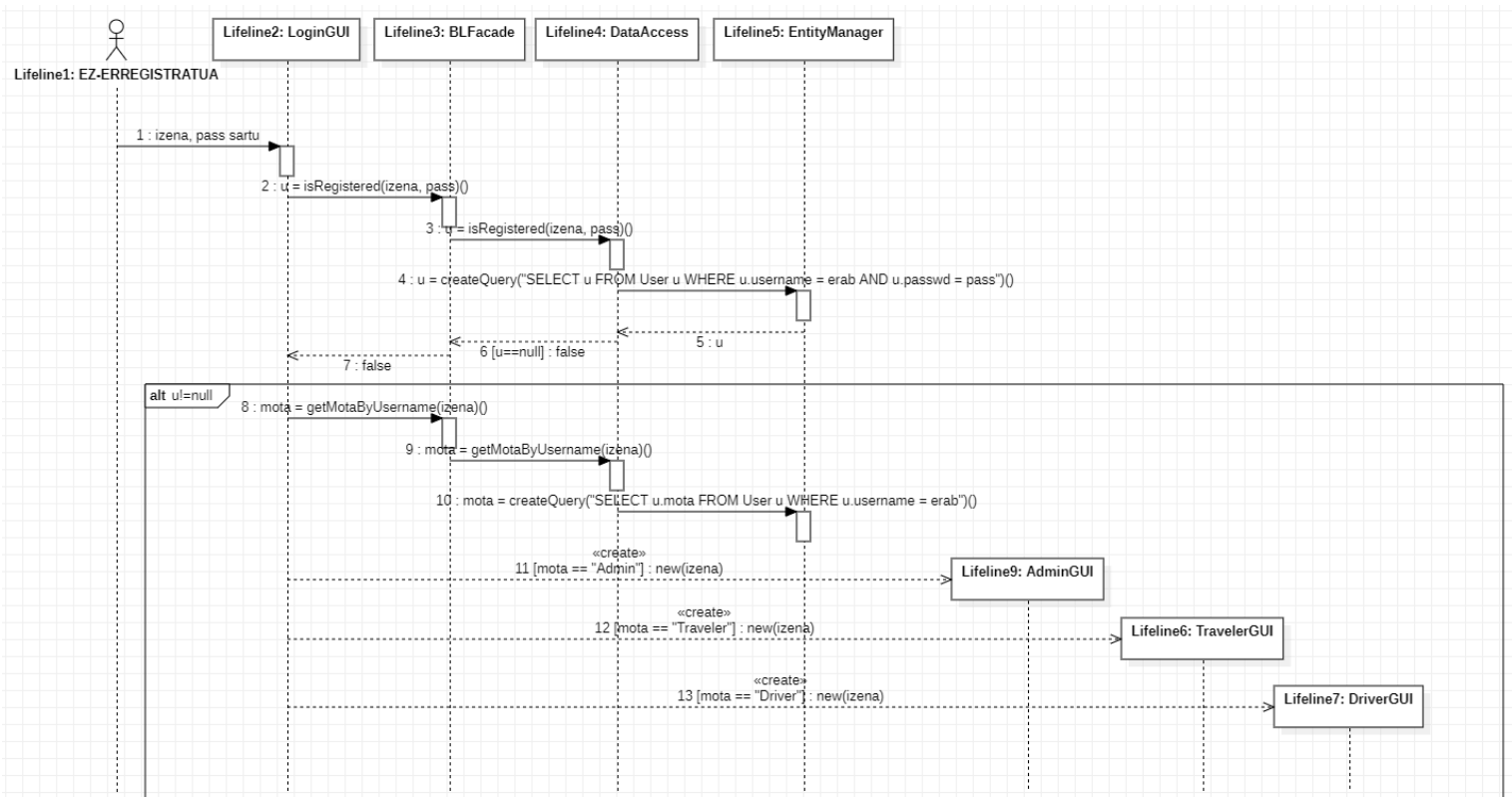
Gertaera fluxuak

Basic flow

1. Ez-erregistratua izena eta pasahitza datuak ipiniz, aplikaziora sartu.
2. Sistemak jarritako datuen arabera kontu bat existitzen bada frogatzen du. Jadantik existitzen bada, erabiltzaileen rollaren arabera aplikaziora sartzen da.

Alternative flow

1. Datuak falta dira. Sistemak erabiltzailea abisatzen du.
2. Ez da existitzen erabiltzailerik. Sistemak erabiltzailea abisatzen du.



SOFTWARE INGENIARITZA

Bidaia bilatu

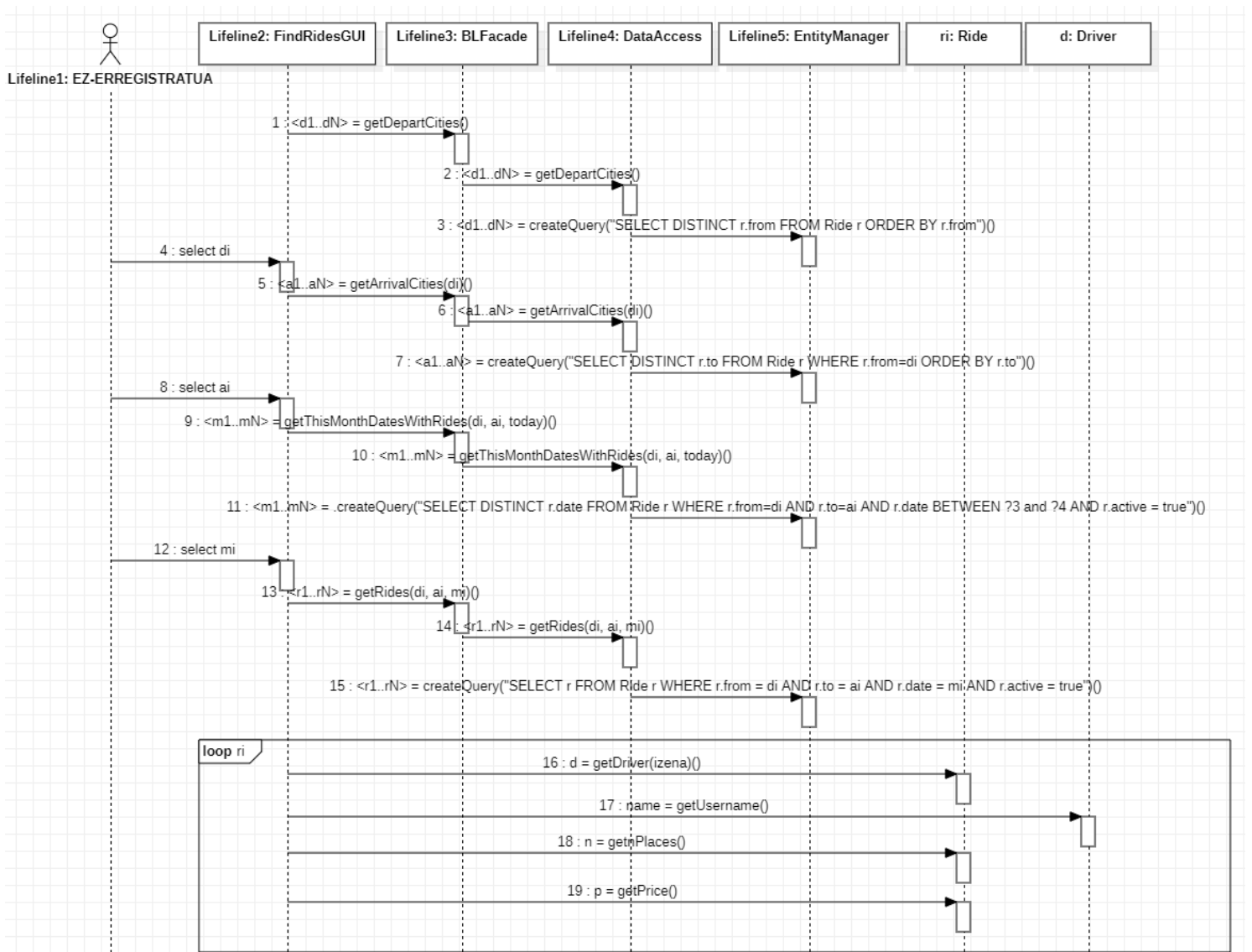
Gertaera fluxuak

Basic flow

1. Sistema nondik norako hiri guztiak erakutsi eta egutegia ipini.
2. Bidaia edo ez-erregistratua sistemak jarritako jatorrien artean bat aukeratu.
3. Sistema jatorriko hiriaren arabera helmuga aukerak eguneratu.
4. Bidaia edo ez-erregistratua helmuga aukeratu.
5. Sistema nondik norako bidaia aukeratu hiriaren arabera eskuragarri dauden egunak irudikatu.
6. Bidaia edo ez-erregistratua eskuragarri dauden egunetarik bat aukeratu.
7. Sistema aukeratuak eguna kolore batekin markatu eta egun horretako bidaia datuak (gidariaren izena, eserlekuak eta prezioa) erakutsi.

Alternative flow

1. Gidariak aukeratuak egunean ez dago bidaiarik. Bidaia lista hutsa dago.



SOFTWARE INGENIARITZA

Bidaia erreserba egin

Gertaera fluxuak

Basic flow

1. *Sistema* nondik norako hiri guztiak erakutsi eta **egutegia** ipini.
2. *Bidaia* sistemak jarritako jatorrien artean bat aukeratu.
3. *Sistema* jatorriko hiriaren arabera helmuga aukerak eguneratu.
4. *Bidaia* helmuga aukeratu.
5. *Sistema* nondik norako bidaiaren aukeratutako hirien arabera eskuragarri dauden egunak irudikatu.
6. *Bidaia* eskuragarri dauden egunetatik bat aukeratu.
7. *Sistema* aukeratutako eguna kolore batekin markatu eta egun horretako bidaien datuak (gidariaren izena, eserlekuak, gidariaren balorazioa eta prezioa) erakutsi.
8. *Bidaia* bidaia bat aukeratu.
9. *Bidaia* deskontu bat sartzeko aukera dauka.
10. *Sistema* bidaiaria deskonturen bat sartu duen konprobatzen du eta aktiboa edo ez dagoen begiratzeko. Deskontua existitzen bada eta aktiboa badago, bidaiaren prezioa eguneratzen du.
11. *Bidaia* izena eta eserlekuak sartu.
12. *Bidaia* aukeratutako bidaiaren erreserba egiten du.
13. *Sistema* bidaiaria existitzen den zihurtatzen du.
14. *Sistema* erreserba sisteman gordetzen du eta bidaiariari esleitzen dio.
15. *Sistema* bidaiaren eserlekuak eguneratzen ditu eta bidaiariari dirua kentzen zaio.
16. *Sistema* erreserba ondo burutu bada, bidaiariaren kontuan diruarekin sortutako mugimendua gordetzen du.

Alternative flow

1. Ez daude bidaiak aukeratutako egunean, beraz ezin da erreserba egin.
2. Deskonturik sartzen ez badu, 0 balioa hartzen du.
3. Deskontuaren kodea ez da existitzen. Sistema erabiltzailea abisatzen du.
4. Ez dago diru nahikoa erreserba egiteko.
5. Libre dauden eserleku baino gehiago erreserbatzen saiatzen ari da. Sistema erabiltzailea abisatzen du.
6. Erreserba ez da ondo gorde. Sistema erabiltzailea abisatzen du.

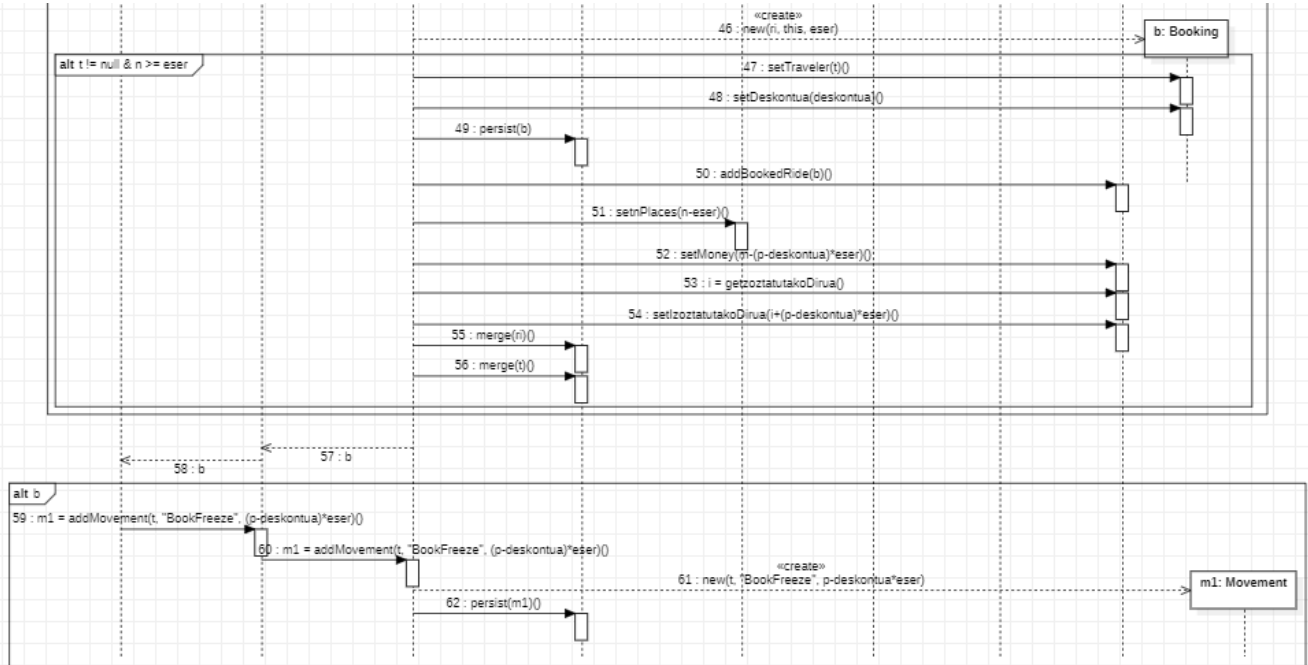
```

sequenceDiagram
    participant TRAVELER as Lifeline1: TRAVELER
    participant BookGUI as Lifeline2: BookGUI
    participant BLFacade as Lifeline3: BLFacade
    participant DataAccess as Lifeline4: DataAccess
    participant EntityManager as Lifeline5: EntityManager
    participant ri as ri: Ride
    participant d as d: Driver
    participant desk as desk: Discount
    participant t as t: Traveler

    TRAVELER->>BookGUI: 1 : <d1.dN> = getDepartCities()
    activate BookGUI
    BookGUI->>BLFacade: 2 : <d1.dN> = getDepartCities()
    deactivate BookGUI
    activate BLFacade
    BLFacade->>DataAccess: 3 : createQuery("SELECT DISTINCT r.from FROM Ride r ORDER BY r.from")()
    deactivate BLFacade
    activate DataAccess
    DataAccess->>EntityManager: 
    deactivate DataAccess
    activate EntityManager
    EntityManager->>ri: 
    deactivate EntityManager
    ri->>d: 
    deactivate ri
    d->>t: 
    deactivate d
    t->>BookGUI: 4 : select di
    deactivate t
    activate BookGUI
    BookGUI->>BLFacade: 5 : <a1.aN> = getArrivalCities(di)()
    deactivate BookGUI
    activate BLFacade
    BLFacade->>DataAccess: 6 : <a1.aN> = getArrivalCities(di)()
    deactivate BLFacade
    activate DataAccess
    DataAccess->>EntityManager: 7 : createQuery("SELECT DISTINCT r.to FROM Ride r WHERE r.from=di ORDER BY r.to")()
    deactivate DataAccess
    activate EntityManager
    EntityManager->>ri: 
    deactivate EntityManager
    ri->>d: 
    deactivate ri
    d->>t: 
    deactivate d
    t->>BookGUI: 8 : select ai
    deactivate t
    activate BookGUI
    BookGUI->>BLFacade: 9 : <m1.mN> = getThisMonthDatesWithRides(di, ai, today)()
    deactivate BookGUI
    activate BLFacade
    BLFacade->>DataAccess: 10 : <m1.mN> = getThisMonthDatesWithRides(di, ai, today)()
    deactivate BLFacade
    activate DataAccess
    DataAccess->>EntityManager: 11 : createQuery("SELECT DISTINCT r.date FROM Ride r WHERE r.from=di AND r.to=ai AND r.date BETWEEN ?3 and ?4 AND r.active = true")()
    deactivate DataAccess
    activate EntityManager
    EntityManager->>ri: 
    deactivate EntityManager
    ri->>d: 
    deactivate ri
    d->>t: 
    deactivate d
    t->>BookGUI: 12 : select mi
    deactivate t
    activate BookGUI
    BookGUI->>BLFacade: 13 : <r1.rN> = getRides(di, ai, m)()
    deactivate BookGUI
    activate BLFacade
    BLFacade->>DataAccess: 14 : <r1.rN> = getRides(di, ai, m)()
    deactivate BLFacade
    activate DataAccess
    DataAccess->>EntityManager: 15 : createQuery("SELECT r FROM Ride r WHERE r.from = di AND r.to = ai AND r.date = mi AND r.active = true")()
    deactivate DataAccess
    activate EntityManager
    EntityManager->>ri: 
    deactivate EntityManager
    ri->>d: 
    deactivate ri
    d->>t: 
    deactivate d
    t->>BookGUI: loop ri
    deactivate t
    activate BookGUI
    BookGUI->>EntityManager: 16 : d = getDriver()
    deactivate BookGUI
    activate EntityManager
    EntityManager->>d: 
    deactivate EntityManager
    d->>t: 17 : name = getUsername()
    deactivate d
    activate t
    t->>BookGUI: 
    deactivate t
    activate BookGUI
    BookGUI->>EntityManager: 18 : n = getnPlaces()
    deactivate BookGUI
    activate EntityManager
    EntityManager->>d: 
    deactivate EntityManager
    d->>t: 
    deactivate d
    activate t
    t->>BookGUI: 19 : p = getPrice()
    deactivate t
    activate BookGUI
    BookGUI->>EntityManager: 20 : balo = getBalorazioa()
    deactivate BookGUI
    activate EntityManager
    EntityManager->>>d: 
    deactivate EntityManager
    d->>t: 
    deactivate d
    activate t
    t->>BookGUI: 21 : select ri
    deactivate t
    activate BookGUI
    BookGUI->>BLFacade: 22 : deskontua sartu
    deactivate BookGUI
    activate BLFacade
    alt deskontua!=null
        BLFacade->>DataAccess: 23 : desk = getDesk(deskontua)()
        deactivate BLFacade
        activate DataAccess
        DataAccess->>EntityManager: 24 : desk = getDesk(deskontua)()
        deactivate DataAccess
        activate EntityManager
        EntityManager->>desk: 25 : desk = createQuery("SELECT d FROM Discount d WHERE d.kodea = deskontua")()
        deactivate EntityManager
        activate desk
        desk->>t: 26 : desk
        deactivate desk
        activate t
        t->>desk: 27 : a = isActive()
        deactivate t
        activate desk
        desk->>t: 28 : a
        deactivate desk
        activate t
        t->>desk: 30 : false
        deactivate t
        activate desk
        alt desk!=null & a
            desk->>t: 31 : por = getPozteztaila()
            deactivate desk
            activate t
            t->>desk: 32 : setPrice(b-(por*p))()
            deactivate t
            activate desk
        else
            desk->>t: 33 : izena_eser sartu
            deactivate desk
            activate t
            t->>BLFacade: 34 : m = getActualMoney(izena)()
            deactivate t
            activate BLFacade
            BLFacade->>DataAccess: 35 : m = getActualMoney(izena)()
            deactivate BLFacade
            activate DataAccess
            DataAccess->>EntityManager: 36 : m = createQuery("SELECT u.money FROM User u WHERE u.username = izena")()
            deactivate DataAccess
            activate EntityManager
            EntityManager->>t: 37 : m
            deactivate EntityManager
            activate t
            t->>BLFacade: 38 : false
            deactivate t
            activate BLFacade
            alt m >= (p - deskontua)*eser
                BLFacade->>DataAccess: 40 : b = bokkRide(izena, ri, eser, deskontua)()
                deactivate BLFacade
                activate DataAccess
                DataAccess->>EntityManager: 41 : b = bokkRide(izena, ri, eser, deskontua)()
                deactivate DataAccess
                activate EntityManager
                EntityManager->>t: 42 : t = getTraveler(izena)()
                deactivate EntityManager
                activate t
                t->>BLFacade: 43 : t
                deactivate t
                activate BLFacade
            else
                BLFacade->>DataAccess: 44 : false
                deactivate BLFacade
                activate DataAccess
                DataAccess->>EntityManager: 45 : false
                deactivate DataAccess
                activate EntityManager
            end
        end
    end
    deactivate BLFacade
    deactivate DataAccess
    deactivate EntityManager
    deactivate ri
    deactivate d
    deactivate desk
    deactivate t
    Note over BookGUI, BLFacade, DataAccess, EntityManager, ri, d, desk, t: «create»
    BookGUI->>BLFacade: 46 : new(n, this, eser)
    deactivate BookGUI
    activate BLFacade
    BLFacade->>DataAccess: 
    deactivate BLFacade
    activate DataAccess
    DataAccess->>EntityManager: 
    deactivate DataAccess
    activate EntityManager
    EntityManager->>ri: 
    deactivate EntityManager
    ri->>d: 
    deactivate ri
    d->>t: 
    deactivate d
    activate t
    t->>BookGUI: b: Booking
    deactivate t
    activate BookGUI

```

SOFTWARE INGENIARITZA



Bidaia osatuta edo ez osatuta markatu

Gertaera fluxuak

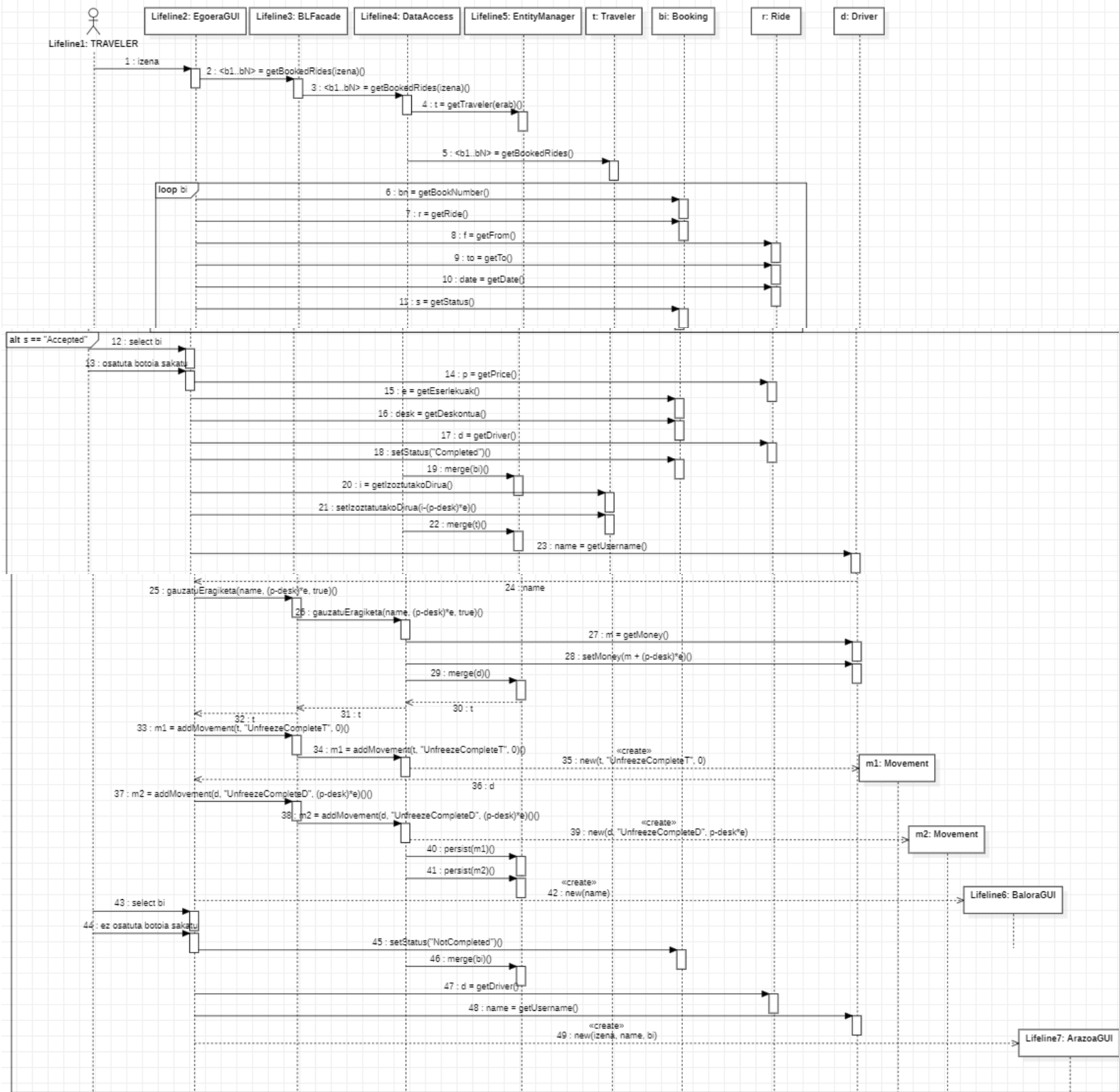
Basic flow

1. *Bidaia* izena eta egoera sartzen ditu.
2. *Sistemak* bidaia iragarri dagokion erreserben lista erakusten du, hain zuzen ere, erreserben zenbakia, nondik nora bidaia, data eta erreserbaren egoera.
3. *Bidaia* "Accepted" egoeran dagoen erreserba eskaera bat aukeratu eta **osatuta** edo **ez osatuta** dagoen markatzen du.
4. *Sistemak* erreserbaren egoera eguneratzen du.
5. *Sistemak osatuta* markatzerakoan, gidari bidaia prezioa kontuan hartu eta bidaia dirua kontutik atera. Aurreko bi diru-mugimenduak sisteman go *BaloraGUI* pantailara bideratu.
6. *Sistemak ez osatuta* markatzerakoan, *ArazoaGUI* pantailara bideratu.

Alternative flow

1. Bidaia iragarri ez du aukeratu. *Sistemak* erabiltzailea abisatzen du.
2. Aukerutako erreserbaren egoera ez dago zehaztuta. *Sistemak* erabiltzailea abisatzen du.
3. Bidaia gaur baino lehenagokoa izan behar da. *Sistemak* erabiltzailea abisatzen du.

SOFTWARE INGENIARITZA

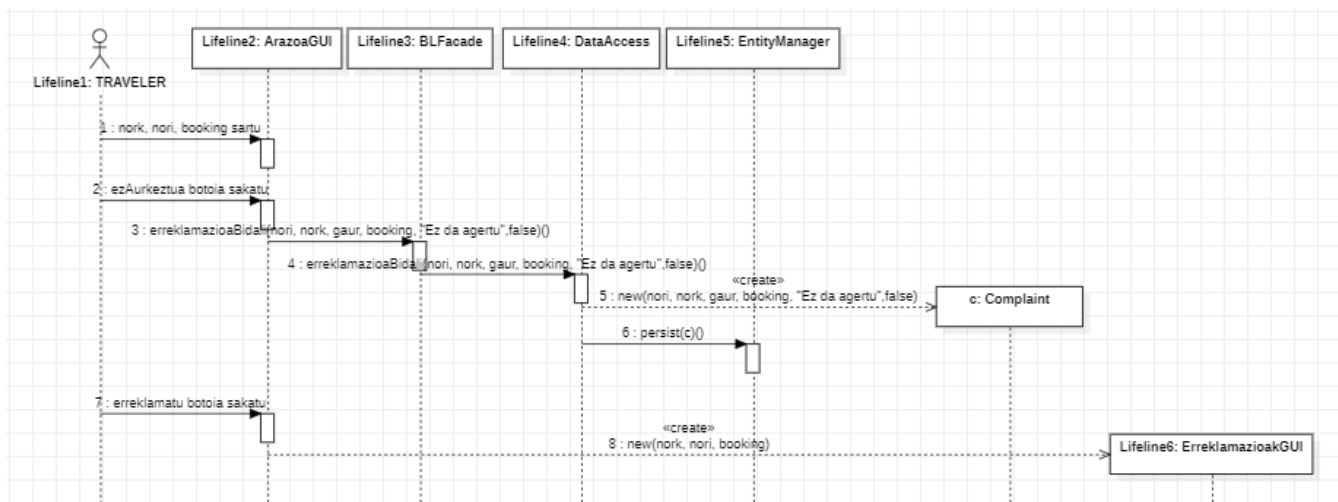


Arazoak adierazi

Gertaera fluxuak

Basic flow

1. *Bidaia*riak *nork*, *nori* eta *erreserba* sartu.
2. *Bidaia*riak *ezAurkeztua* botoia sakatu, *bidaia*ko egunean ez delako agertu.
3. *Sistemak* erreklamazio bat bidaliko du *gidariaren* izenean adierazteko *bidaia*ria ez dela agertu eta sortutako erreklamazioa gorde.
4. *Bidaia*riak *erreklamatu* botoia sakatu, erreklamazioa jartzeko.
5. *Sistemak* *ErreklamazioakGUI* pantailara bideratu.



SOFTWARE INGENIARITZA

Alerta aurkituak

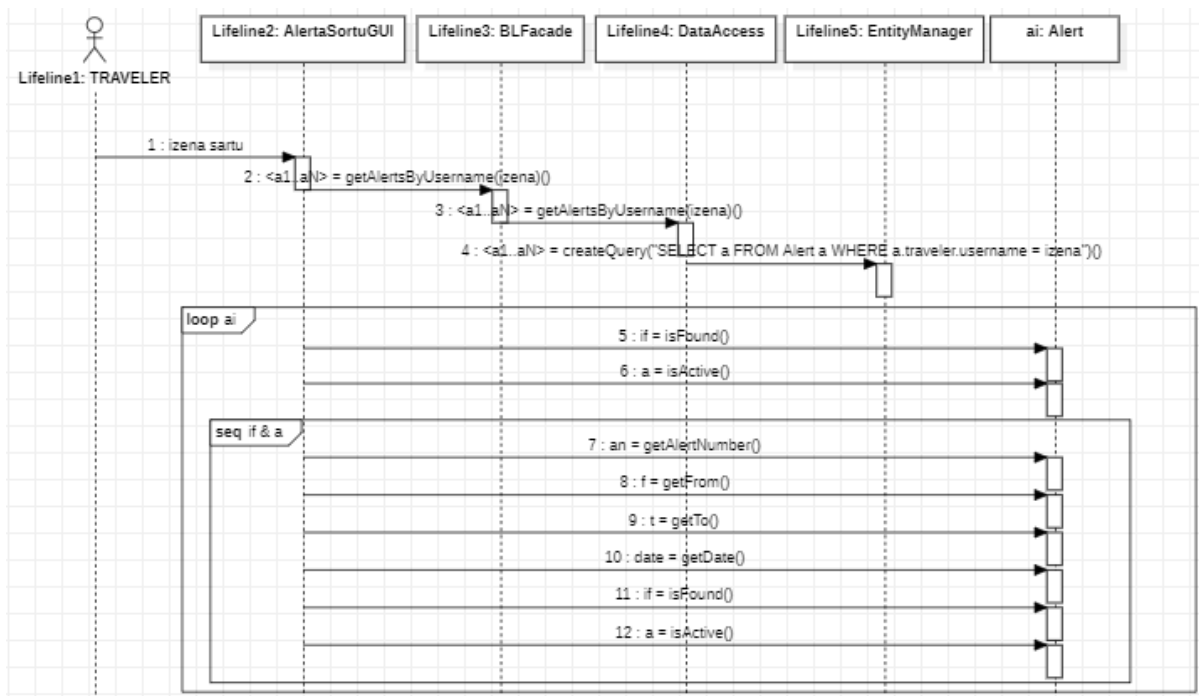
Gertaera fluxuak

Basic flow

1. Bidaiariak izena sartzen du.
2. Sistemak bidaiariari dagokion alerten lista erakusten du, hain zuzen ere, alerta zenbakia, nondik, nora, data, aurkituta dagoen eta aktibatuta dagoen.

Alternative flow

1. Alertarik ez da agertzen. Sistemak erabiltzailea abisatzen du.



Alertak sortu

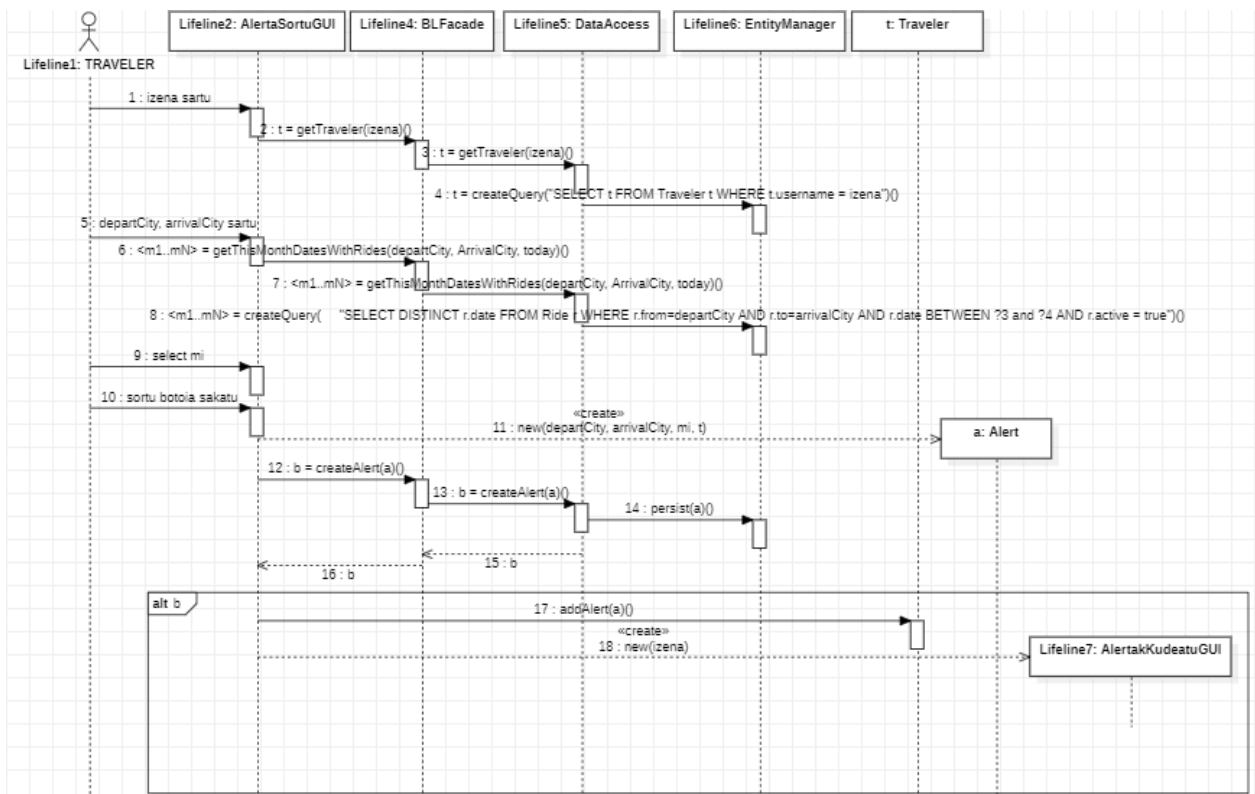
Gertaera fluxuak

Basic flow

1. *Bidaia*riak nondik, norako bidaia sartzen du.
2. *Sistemak* egutegia erakusten du.
3. *Bidaia*riak egun bat aukeratu.
4. *Bidaia*riak izena sartzen du.
5. *Bidaia*riak sortu botoia sakatzen du.
6. *Sistemak* alerta berria sortu eta gordetzen du.
7. *Sistemak* alerta ondo sortuta konprobatzen du eta *bidaia*riari esleitu.
8. *Sistemak* *AlertakKudeatuGUI* pantailara bideratu.

Alternative flow

1. Alerta berria ez da sortu. *Sistemak* erabiltzailea abisatzen du.



SOFTWARE INGENIARITZA

Alertak kudeatu

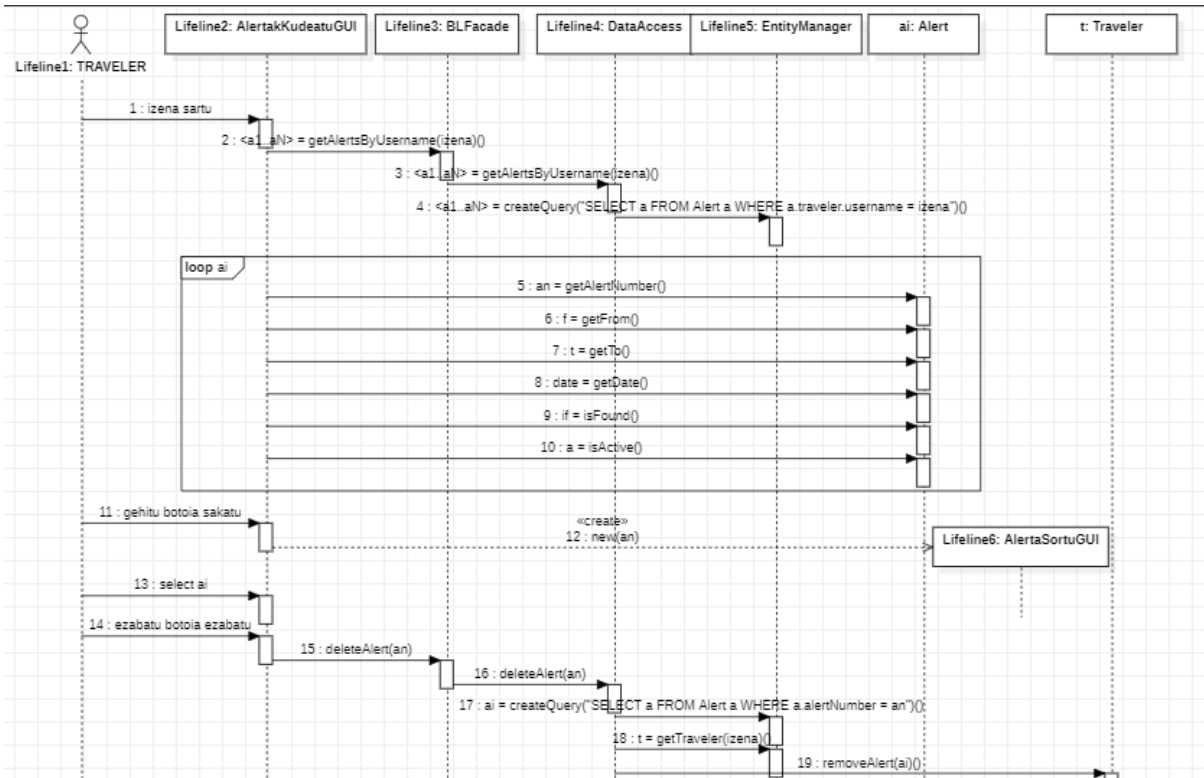
Gertaera fluxuak

Basic flow

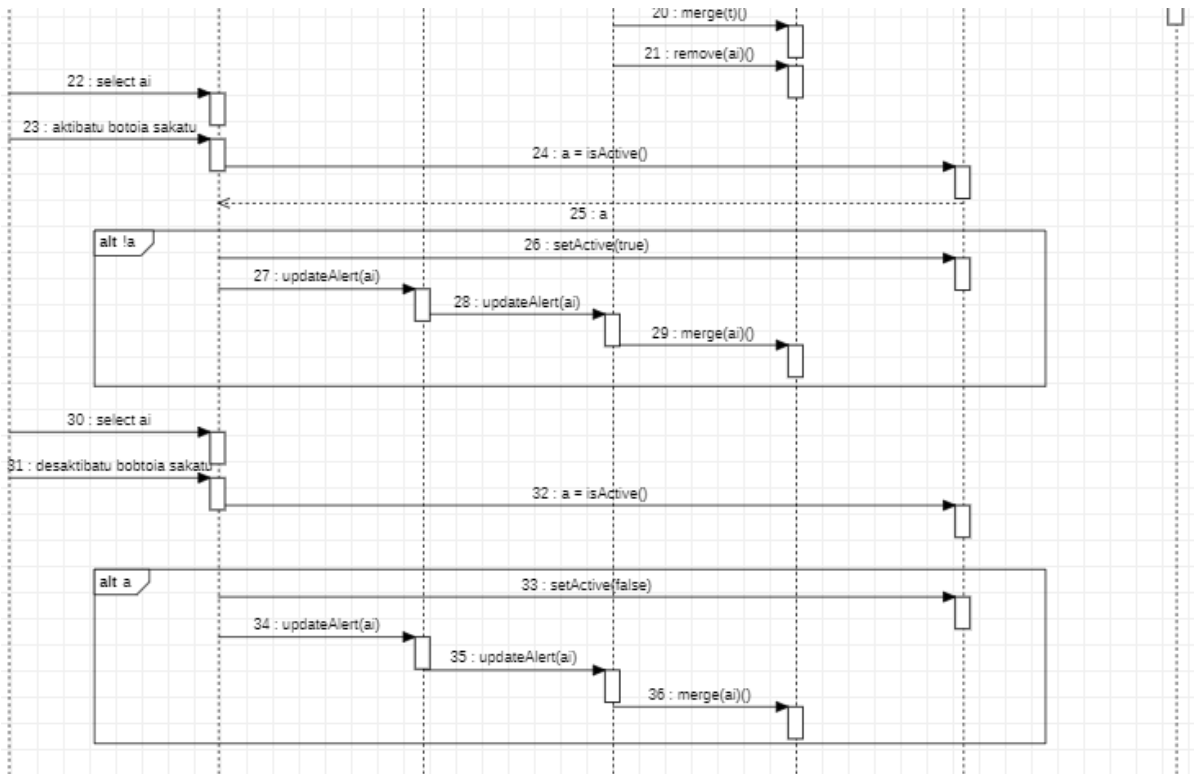
1. Bidaiaariak izena sartzen du.
2. Sistemak bidaiaariari dagokion alerten lista erakusten du, hain zuzen ere, alerta zenbakia, nondik, nora, data, aurkituta dagoen edo ez eta aktibatuta dagoen edo ez.
3. Bidaiaariak gehitu botoia sakatu.
4. Sistemak *AlertaSortuGUI* pantailara bideratu.
5. Bidaiaariak alerta bau aukeratu eta ezabatu botoia sakatu.
6. Sistemak bidaiaariari alerta kendu, eguneratu eta sistematik kendu.
7. Bidaiaariak alerta bau aukeratu eta aktibatu botoia sakatu.
8. Sistemak aukeratutako alertaren egoera konprobatu, aktibatuta ez badago, aktibatu eta eguneratu.
9. Bidaiaariak alerta bau aukeratu eta desaktibatu botoia sakatu.
10. Sistemak aukeratutako alertaren egoera konprobatu, aktibatuta badago, desaktibatu eta eguneratu.

Alternative flow

1. Bidaiaariak erreserbarik ez du aukeratu. Sistemak erabiltzailea abisatzten du.



SOFTWARE INGENIARITZA



Bidaia sortu

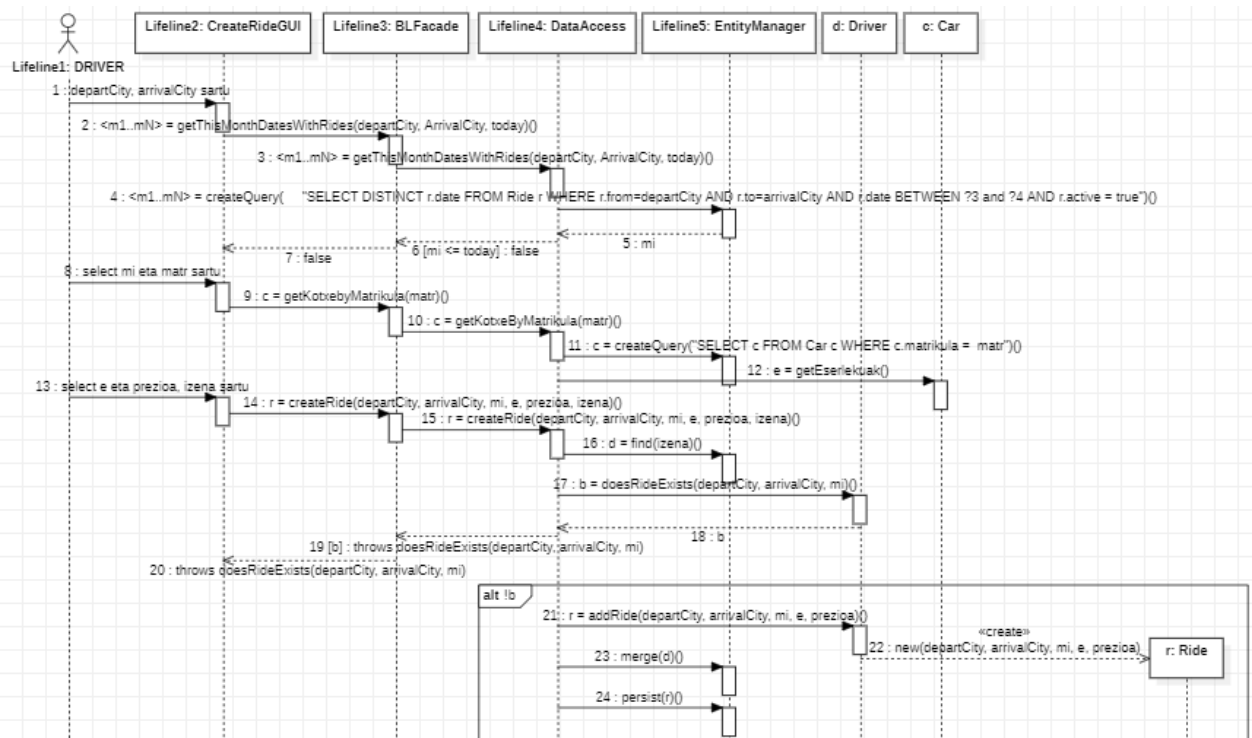
Gertaera fluxuak

Basic flow

1. *Gidaria* jatorria, helmuga, prezioa eta izena sartu.
2. *Sistema* egutegia erakutsi.
3. *Gidaria* bidaia eguna aukeratu.
4. *Gidaria* kotxearen matrikula sartu.
5. *Sistema* matrikula hori dagokion kotxea erakutsi.
6. *Gidaria* kotxea eta eserleku kopurua aukeratu.
7. *Sistemak* bidaia sortzen du eta gidariari esleitzen dio.

Alternative flow

1. Datuak falta dira. Amaitu.
2. eserleku kopurua edo prezioa ez dira zenbakiak. Sistemak erabiltzailea abisatzen du.
3. bidaia eguna pasa da. Sistemak erabiltzailea abisatzen du.
4. bidaia dagoeneko existitzen da gidari honentzat. Bidaia ez da sortzen. Sistemak erabiltzailea abisatzen du.



SOFTWARE INGENIARITZA

Bidaia kantzeltatu

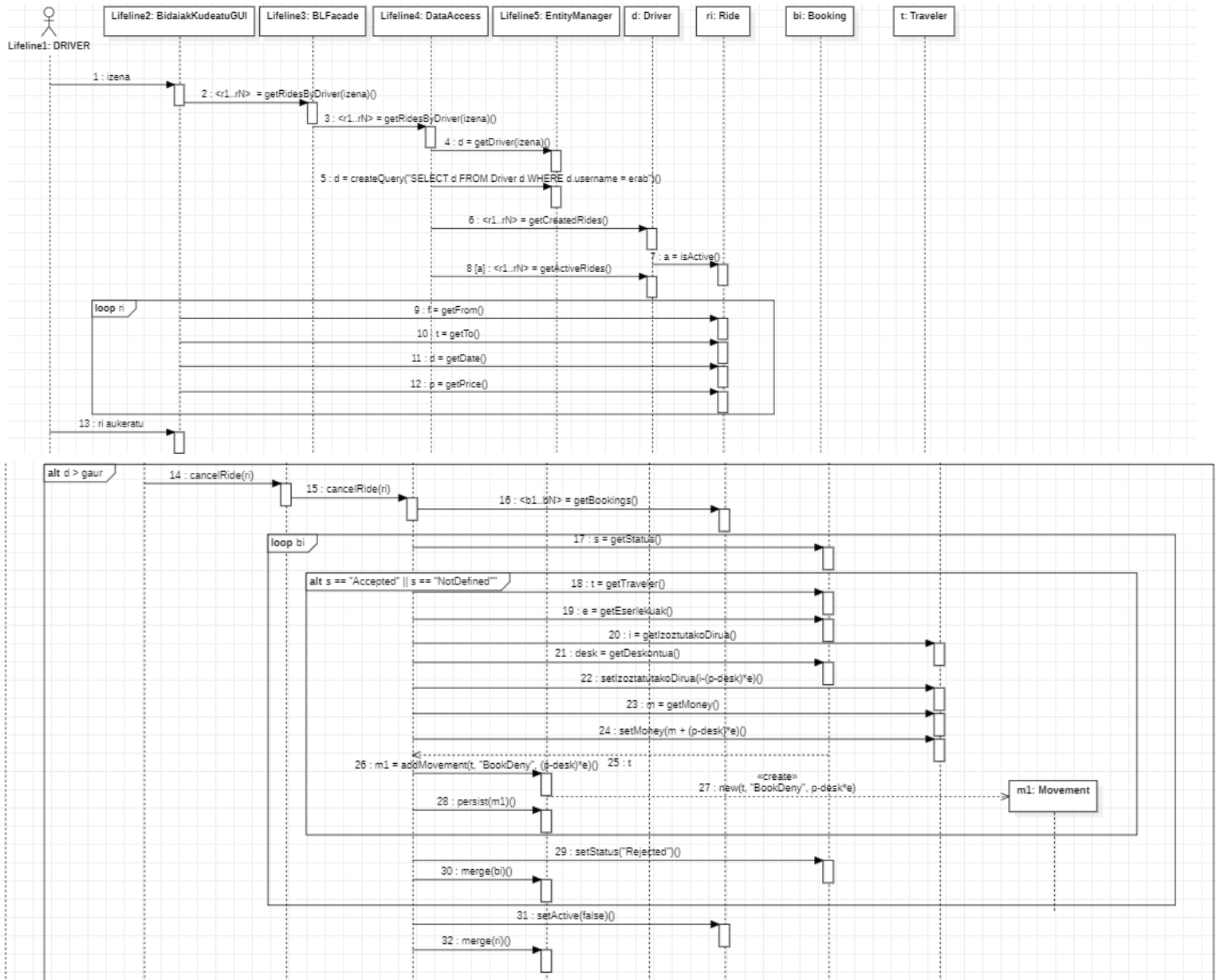
Gertaera fluxuak

Basic flow

1. Gidaria izena sartzen du.
2. Sistemak gidariari dagokion bidaia lista erakusten du, hain zuzen ere, nondik norako bidaia, data eta prezioa.
3. Gidaria bidaia aukeratu.
4. Sistemak dagokion bidaia kantzeltatu du. Horretarako bidaia erreserben lista aztertu behar du.
5. Sistemak "Accepted" edo "NotDefined" egoera dauden erreserba aukeratu eta bidaia bidaia prezioa kontuan sartu eta sortutako diru-mugimendua gorde.
6. Sistemak erreserben egoera "Rejected" egoerara aldatu eta eguneratu.
7. Sistemak bidaia egoera false egoerara eguneratu.

Alternative flow

1. Gidariak bidaia ez du aukeratu. Sistemak erabiltzailea abisatzeko du.
2. Aukeratu bidaia egoera jadanik zehaztuta dago. Sistemak erabiltzailea abisatzeko du.
3. Dagoeneko bidaia data pasa da. Sistemak erabiltzailea abisatzeko du.



SOFTWARE INGENIARITZA

Erreserba onartu edo baztertu

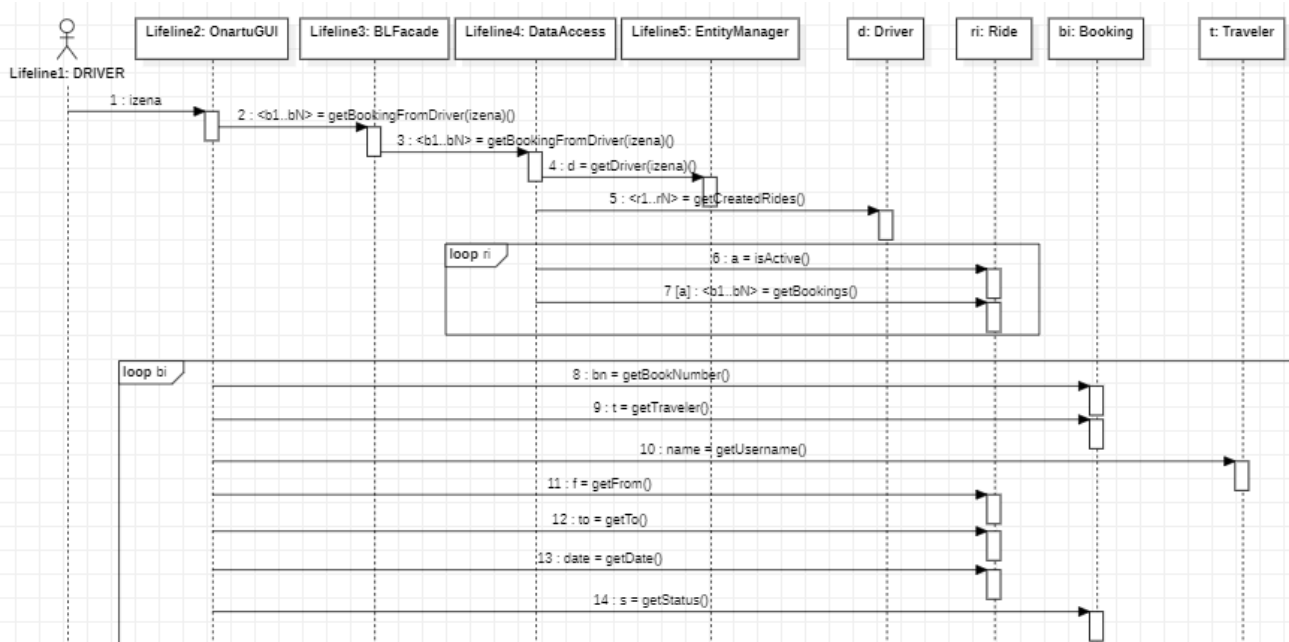
Gertaera fluxuak

Basic flow

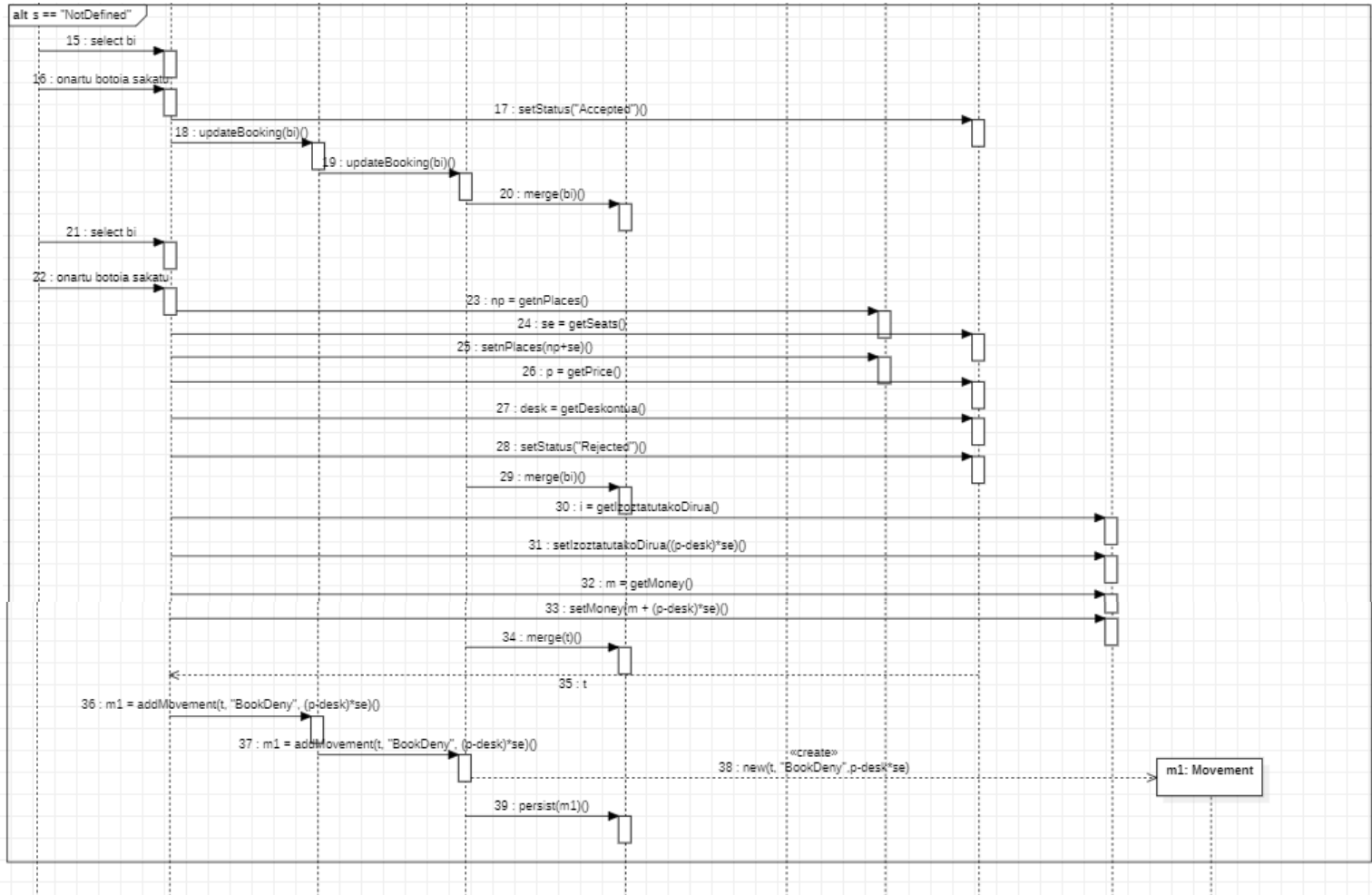
1. *Gidaria* izena eta egoera sartu
2. *Sistemak* gidariari dagokion erreserben lista erakusten du, hain zuzen ere, erreserben zenbakia, bidaiariaren izenak, nondik nora bidaiak, data eta erreserbaren egoera.
3. *Gidaria* erreserba eskaera bat aukeratu eta **onartu** edo **baztertu** du.
4. *Bidaiaria* "NotDefined" egoeran dagoen erreserba eskaera bat aukeratu eta **onartu** edo **ez onartu** markatzen du.
5. *Sistemak* **onartu** markatzerakoan, erreserbaren egoera "Accepted" egoerara aldatu eta eguneratu.
6. *Sistemak* **ez onartu** markatzerakoan, erreserbaren egoera "Rejected" egoerara aldatu eta eguneratu. Bidaiaren eserlekuak eguneratu. Bidaiariari bidaiaren prezioa kontuan sartu eta sortutako diru-mugimendua sisteman gorde.

Alternative flow

1. Gidariak erreserbarik ez du aukeratu. *Sistemak* erabiltzailea abisatzen du.
2. Aukeraturako erreserbaren egoera jadanik zehaztuta dago. *Sistemak* erabiltzailea abisatzen du.



SOFTWARE INGENIARITZA



SOFTWARE INGENIARITZA

Kotxea gehitu

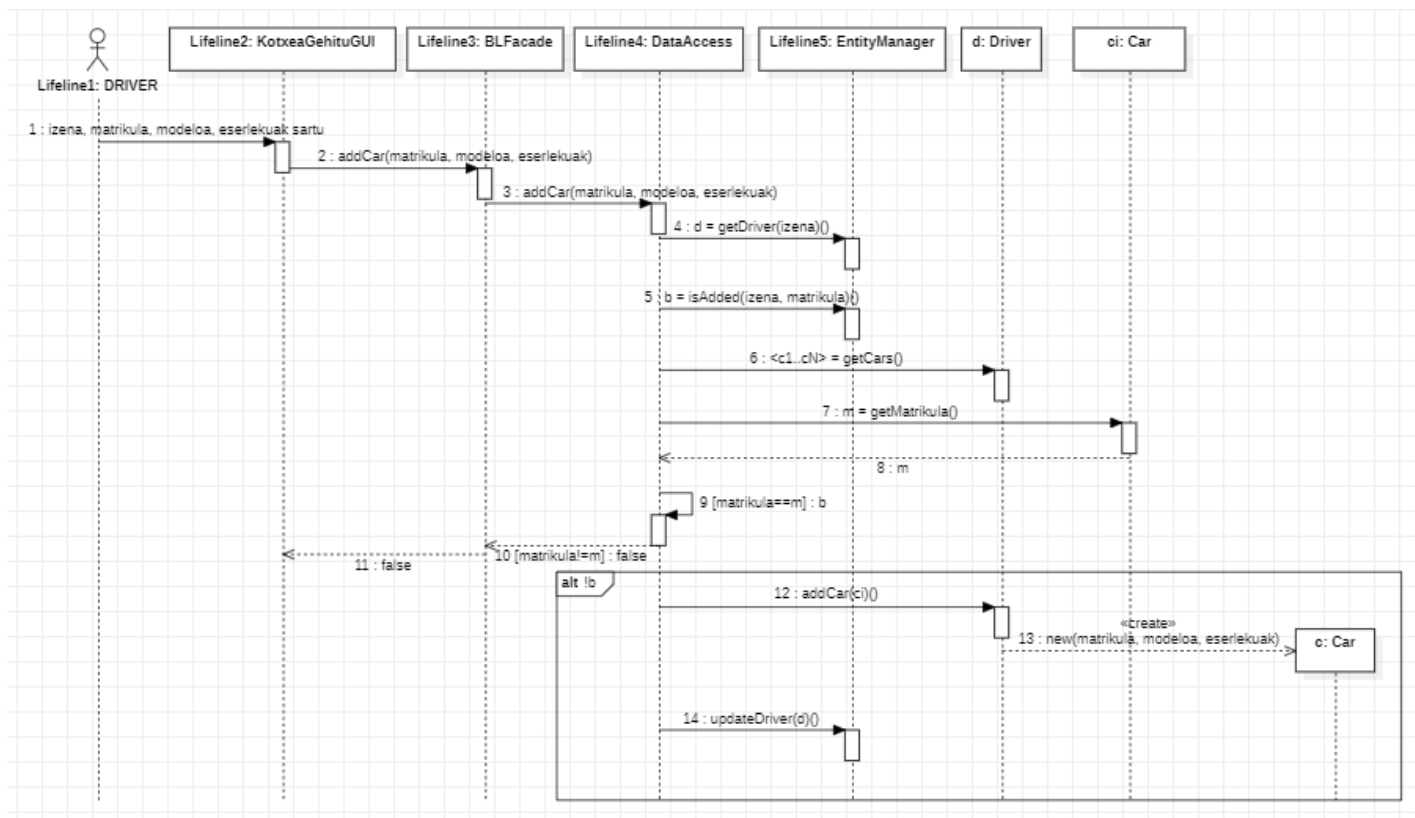
Gertaera fluxuak

Basic flow

1. *Gidaria* matrikula, modeloa eta eserleku kopurua ipini.
2. *Sistemak* aukeratutako datuen arabera kotxea gehitzen du.
3. *Sistemak* kotxea gehitzeko dagoeneko sortuta badago konprobatzen du. Sortuta ez badago gidariari kotxea esleitu eta eguneratu.

Alternative flow

1. Datuak falta dira. *Sistemak* erabiltzailea abisatzen du.
2. eserleku kopurua ez da zenbakia. *Sistemak* erabiltzailea abisatzen du.
3. kotxea jada existitzen da gidari honentzat. Kotxea ez da sortzen. *Sistemak* erabiltzailea abisatzen du.



Kotxeak kudeatu

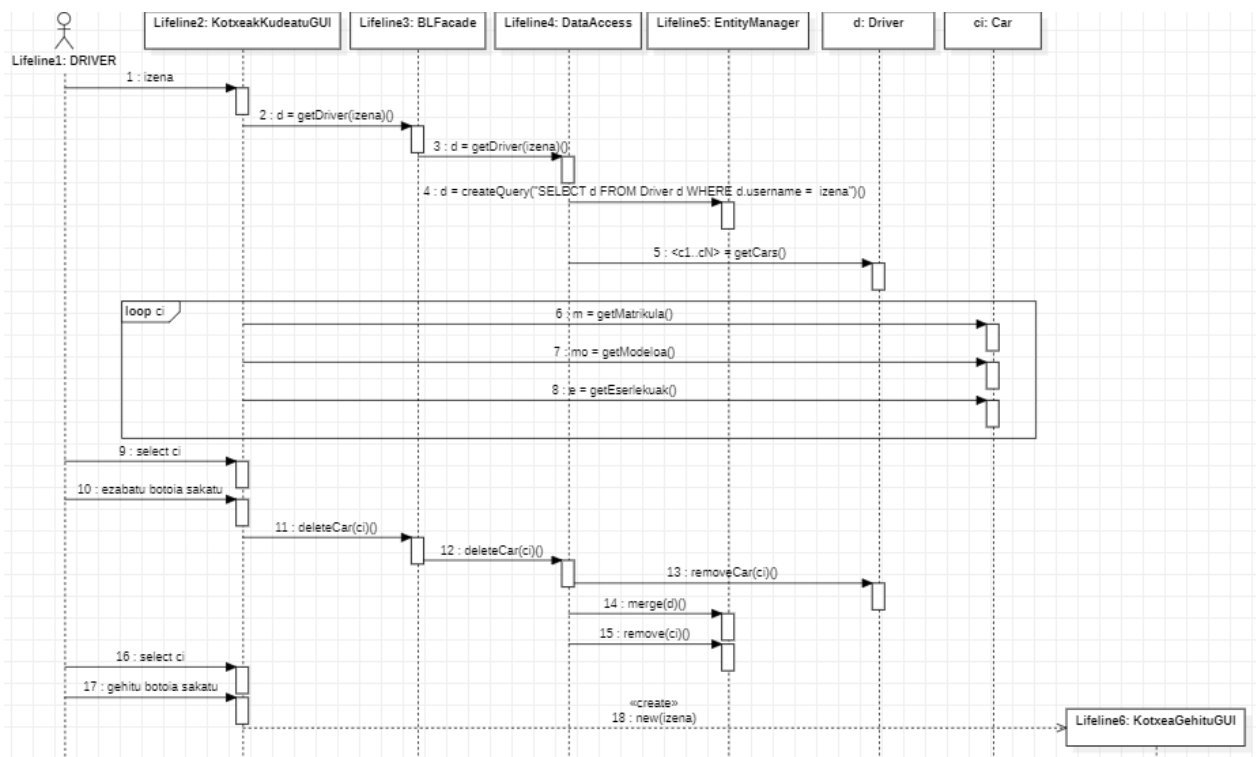
Gertaera fluxuak

Basic flow

1. *Gidariak izena sartzen du.*
2. *Sistemak* gidariari dagokion kotxeen lista erakusten du, hain zuzen ere, matrikula, modeloa eta eserlekuak.
3. *Gidariak* kotxe bat aukeratu eta ezabatu botoia sakatu.
4. *Sistemak* kotxea ezabatzen du.
5. *Gidariak* gehitu botoia sakatu eta *KotxeaGehituGUI* pantailara bideratu.

Alternative flow

1. Kotxeen lista hutsa da. Ez da kotxerik agertzen.



SOFTWARE INGENIARITZA

bezeroak kudeatu

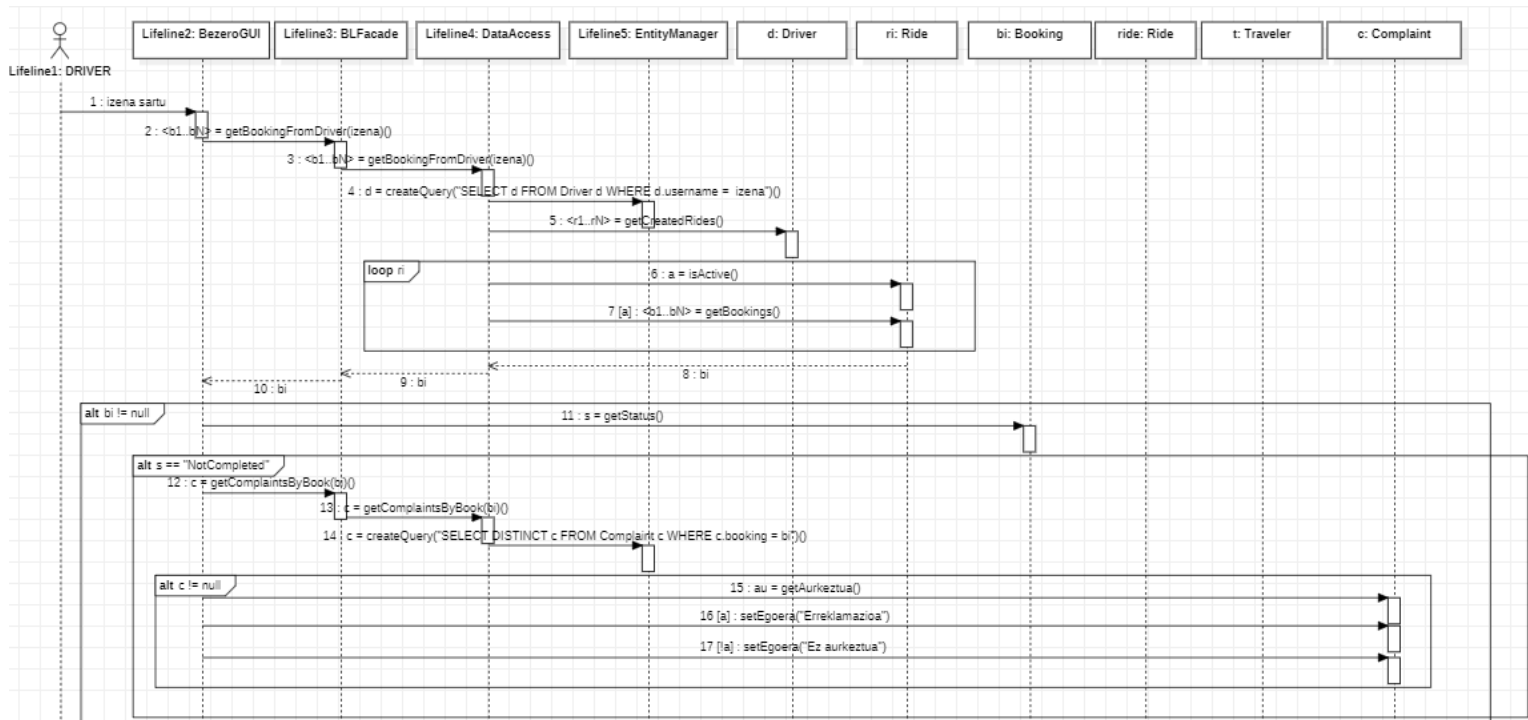
Gertaera fluxuak

Basic flow

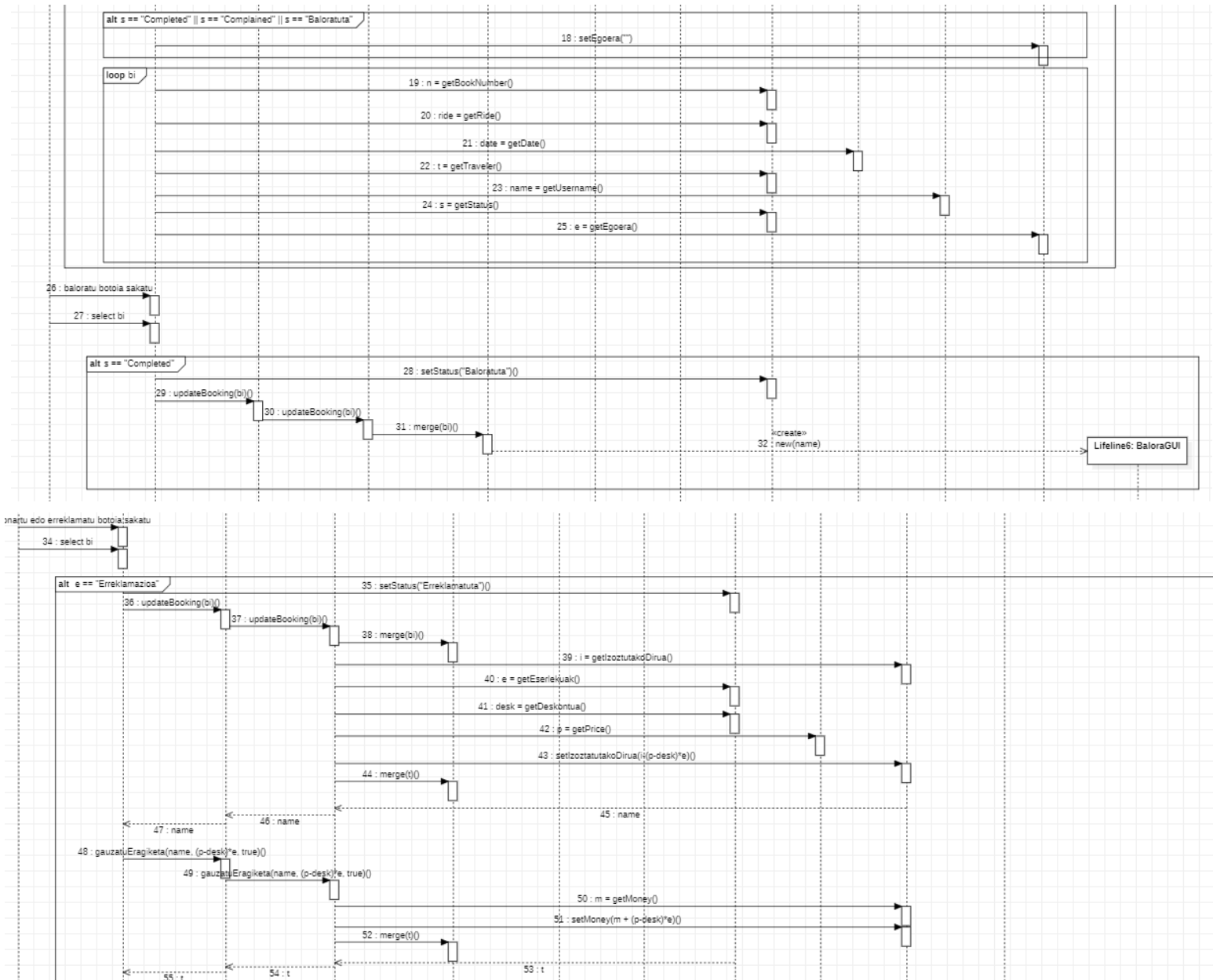
1. *Gidariak* izena sartzen du.
2. *Sistemak* gidariari dagokion erreserben lista eskuratzen du eta gidariak sortutako bidaia guztietatik aktibo egoeran dauden erreserbak hartu.
3. *Sistemak* erreserben egoera eskuratu.
4. *Sistemak* "NotCompleted" egoeran dauden kasuetan, erreklamazioak hartu eta aurkeztua edo ez dagoen konprobatu. Aurkeztua izan den kasuetan, erreklamazioaren egoera "Erreklamazioa" jarri, bestela "Ez aurkeztua".
5. *Sistemak* "Completed", "Complained" edo "Baloratu" egoeran daudenei, erreklamazioaren egoera ezer ez jarri.
6. *Sistemak* erreserben datu hauek erakutsi: erreserba zenbakia, bidaia data, bidaia izena, erreserbaren egoera eta erreklamazioaren egoera.
7. *Gidariak* erreserba bat aukeratu.
8. *Gidariak* baloratu botoia sakatu.
9. *Sistemak* erreserbaren egoera "Completed" badago konprobatu, **BaloraGUI** pantailara bideratu eta erreserbaren egoera "Baloratuta" egoerara eguneratu.
10. *Gidariak* onartu/erreklamatu botoia sakatu.
11. *Sistemak* konprobatu erreklamazioaren egoera.
12. *Sistemak* erreklamazioaren egoera "Erreklamazioa"-n badago, erreserbaren egoera eguneratu, bidaia irudi bueltatu eta sortutako diru-mugimenduak gorde. Gidariaren erreklamazio kopuruari bat gehitu.
13. *Sistemak* erreklamazioaren egoera "Ez aurkeztua"-n badago, erreserbaren egoera eguneratu, bidaia irudi kendu, gidariari dirua bueltatu eta sortutako diru-mugimenduak gorde. Bidaia irudiaren erreklamazio kopuruari bat gehitu.

Alternative flow

1. Erreserben lista hutsa da. Ez da erreserbarik agertzen.
2. Ez dago erreklamaziorik. Ez da agertzen.



SOFTWARE INGENIARITZA



```

sequenceDiagram
    actor Actor
    participant m2 as m2: Movement
    participant m3 as m3: Movement
    participant m4 as m4: Movement

    alt "Ez aurkeztua"
        Actor->>m2: 63: m2 = addMovement(d, "UnfreezeCompleteD", (p-desk)*e)()
        m2->>m2: 64: m2 = addMovement(d, "UnfreezeCompleteD", (p-desk)*e)()
        m2->>m2: 65: new(t, "UnfreezeCompleteD", p-desk)*e)
        m2->>m2: 66: persist(m2)()
        m2->>m2: 67: k = getErreklamakop()
        m2->>m2: 68: setErreklamakop(k+1)()
        m2->>m2: 69: merge(d)()
    else
        Actor->>m2: 70: setStatus("Erreklamatur")()
        m2->>m2: 71: updateBooking(b)()
        m2->>m2: 72: updateBooking(b)()
        m2->>m2: 73: merge(b)()
        m2->>m2: 74: e = getEseriekak()
        m2->>m2: 75: i = getizoztutakoDirua()
        m2->>m2: 76: desk = getDeskptua()
        m2->>m2: 77: setizoztutakoDirua((p-desk)*e)()
        m2->>m2: 78: k = getErreklamakop()
        m2->>m2: 79: setErreklamakop(k+1)()
        m2->>m2: 80: merge()()
    else
        Actor->>m2: 81: gauzatuEraketa(name, (p-desk)*e, true)()
        m2->>m2: 82: gauzatuEraketa(name, (p-desk)*e, true)()
        m2->>m2: 83: m = getMoney()
        m2->>m2: 84: setMoney(m + (p-desk)*e)()
        m2->>m2: 85: t
        m2->>m2: 86: t
        m2->>m2: 87: t
        m2->>m2: 88: m3 = addMovement(t, "UnfreezeCompleteT", 0)()
        m2->>m2: 89: m3 = addMovement(t, "UnfreezeCompleteT", 0)()
        m2->>m2: 90: new(t, "UnfreezeCompleteT", 0)
        m2->>m2: 91: persist(m3)()
        m2->>m2: 92: d
        m2->>m2: 93: d
        m2->>m2: 94: d
        m2->>m2: 95: m4 = addMovement(d, "UnfreezeCompleteD", (p-desk)*e)()
        m2->>m2: 96: m4 = addMovement(d, "UnfreezeCompleteD", (p-desk)*e)()
        m2->>m2: 97: new(d, "UnfreezeCompleteD", p-desk)*e)
        m2->>m2: 98: persist(m4)()
    end

```

SOFTWARE INGENIARITZA

Erabiltzailea ezabatu

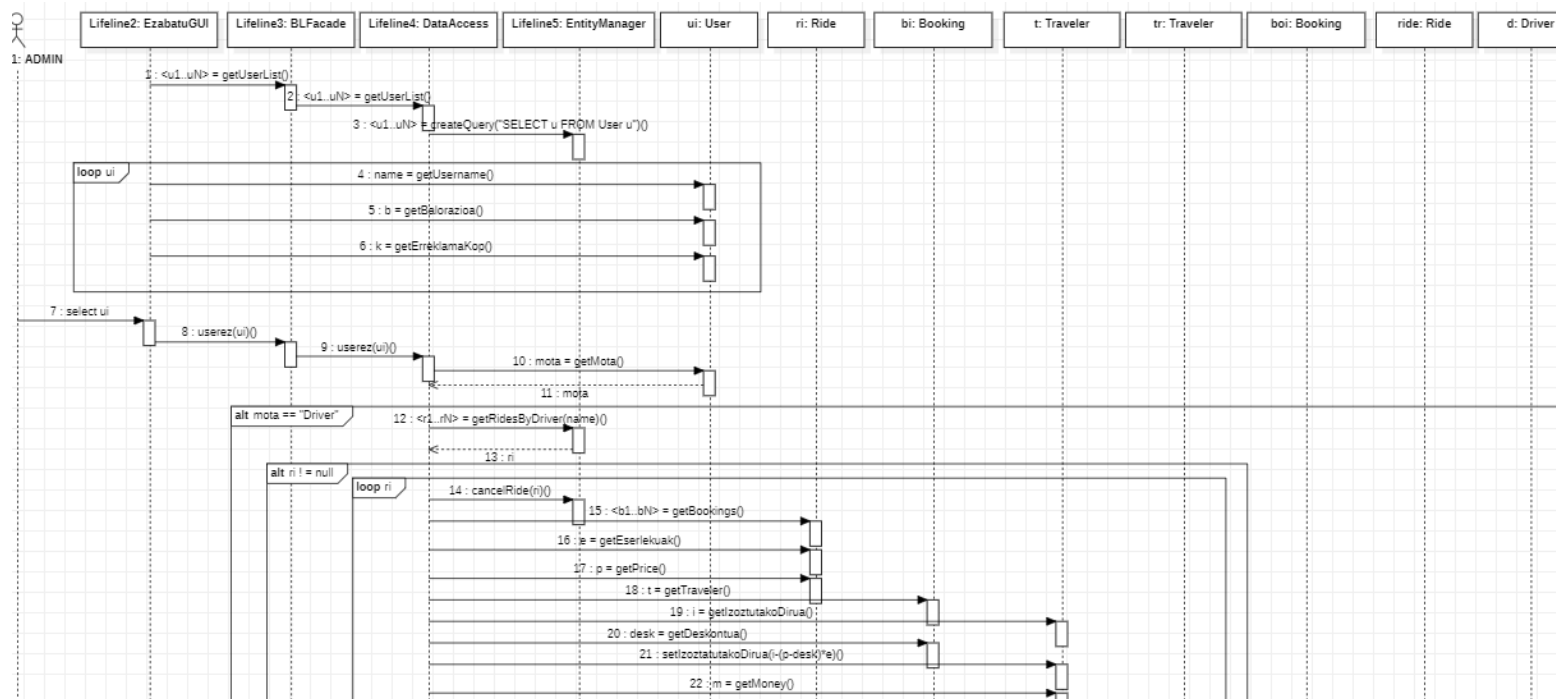
Gertaera fluxuak

Basic flow

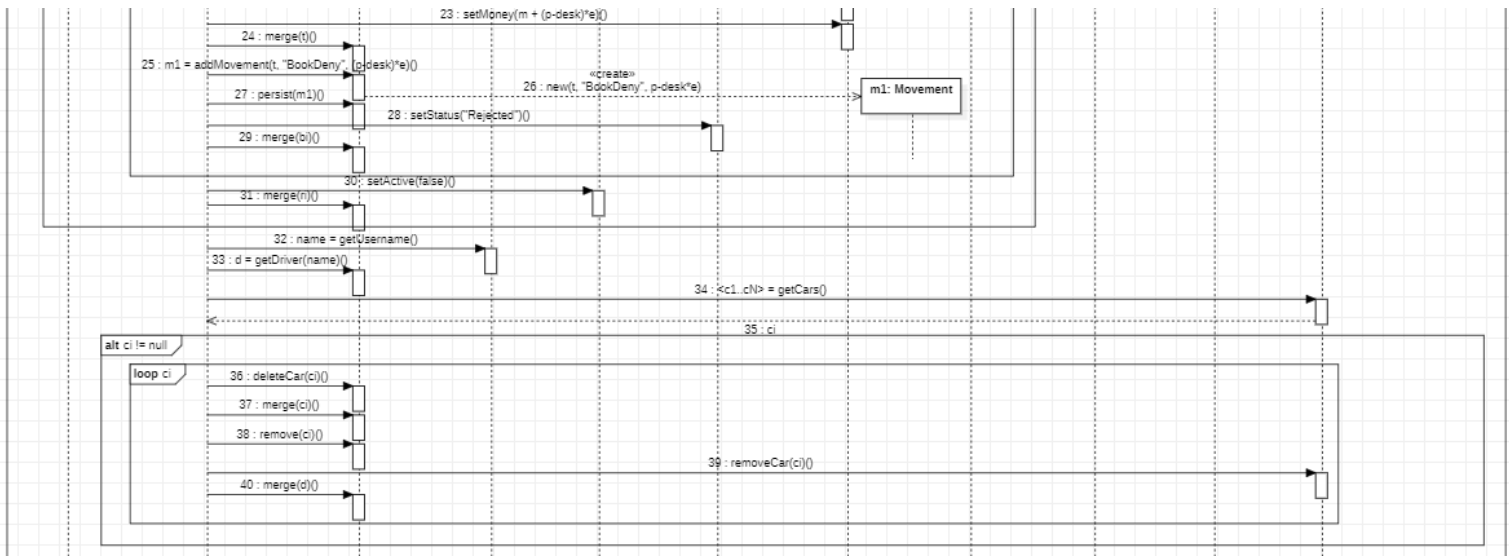
1. *Sistemak* erabiltzaileen lista erakusten du, hain zuzen ere, izena, balorazioa eta erreklamazio kopurua.
2. *Adminak* erabiltzaile bat aukeratu.
3. *Sistemak* erabiltzailea ezabatzeko, mota konprobatu.
4. *Sistemak* gidaria bada, bidaia kantzeltzen du. Horretarako, bidaia erreserba egin duten bidaiariei dirua bueltatu eta bidaiaria eguneratu. Sortutako diru-mugimendua gorde eta erreserba eguneratu.
5. *Sistemak* gidariak kotxerik duen konprobatu eta ezabatu.
6. *Sistemak* bidaiaria bada, erreserbatutako bidaia aukeratu eta erreserbaren egoera eguneratu. Azkenik erreserbatutako bidaia eselerkuak bidaiari gehitu.
7. *Sistemak* erabiltzailea eguneratu eta ezabatu.

Alternative flow

1. Erabiltzaileen lista hutsa da. Ez da kotxerik agertzen.
2. Gidariaren bidaia lista hutsa da. Ez da listarik agertzen.
3. Kotxeen lista hutsa da. Ez da listarik agertzen.
4. Bidaiariaren erreserben lista hutsa da. Ez da listarik agertzen.



SOFTWARE INGENIARITZA



Deskontua gehitu

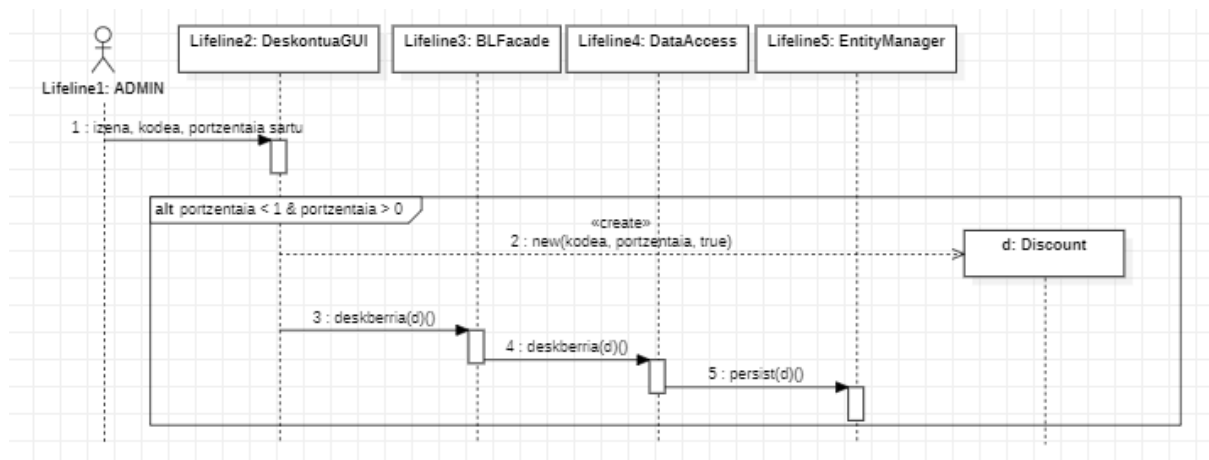
Gertaera fluxuak

Basic flow

1. Admin izena, kodea eta portzentaia sartu.
2. Sistemak sartutako datuekin deskontua gorde.

Alternative flow

1. Datuak falta dira. Sistemak erabiltzailea abisatzen du.
2. Portzentaia 0 eta 1 arteko balioa izan behar da.



SOFTWARE INGENIARITZA

Deskontu kodeak kudeatu

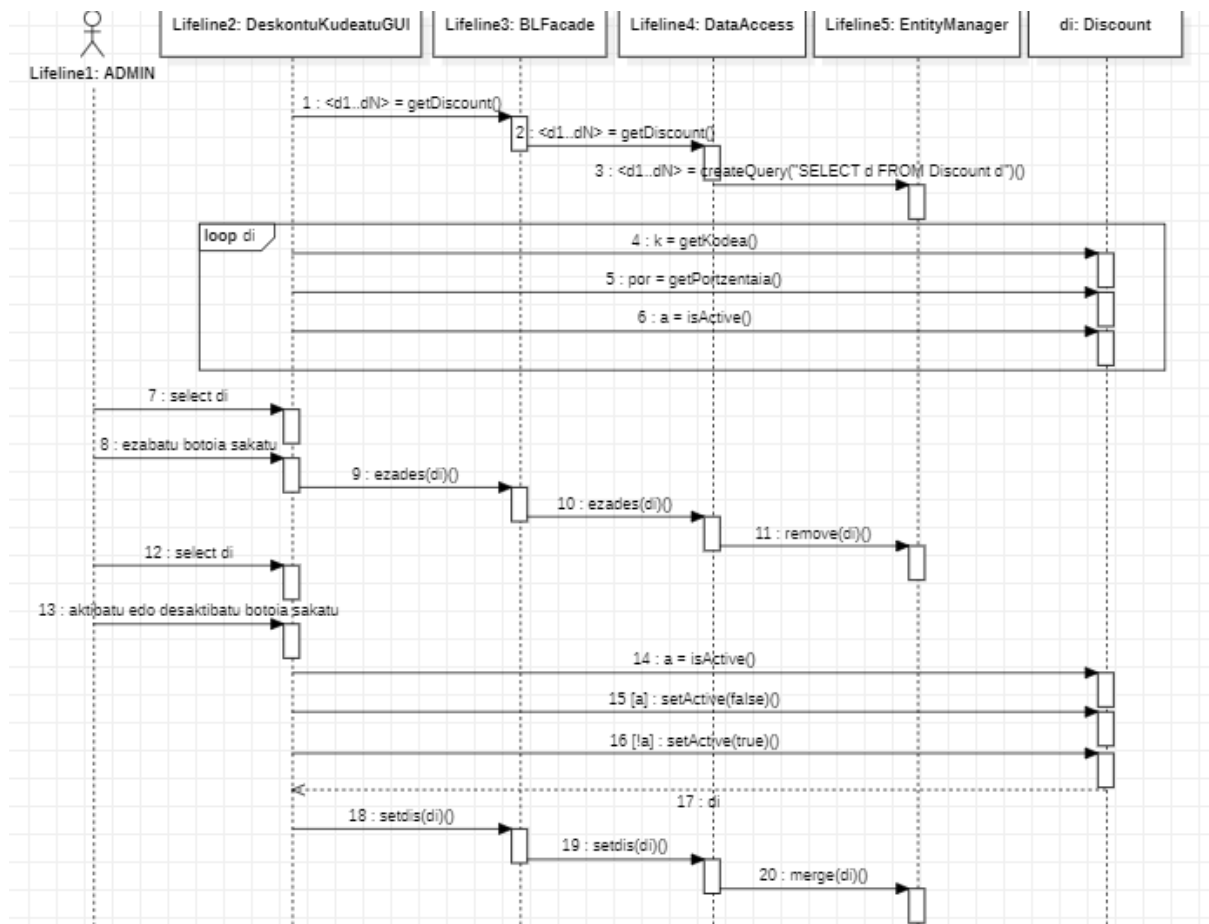
Gertaera fluxuak

Basic flow

1. Sistemak deskontuen lista erakusten du.
2. Adminak deskontu bat aukeratu eta botoi bat sakatu.
3. Sistemak ezabatu botoia sakatzerakoan, deskontua ezabatu.
4. Sistemak aktibatu/desaktibatu botoia sakatzerakoan, deskontuaren egoera konprobatzen du, aktiboa badago desaktibatzen du eta desaktibatuta badago aktibatzen du. Azkenik, deskontua eguneratzen du.

Alternative flow

1. Deskontuen lista hutsa da. Ez da kotxerik agertzen.

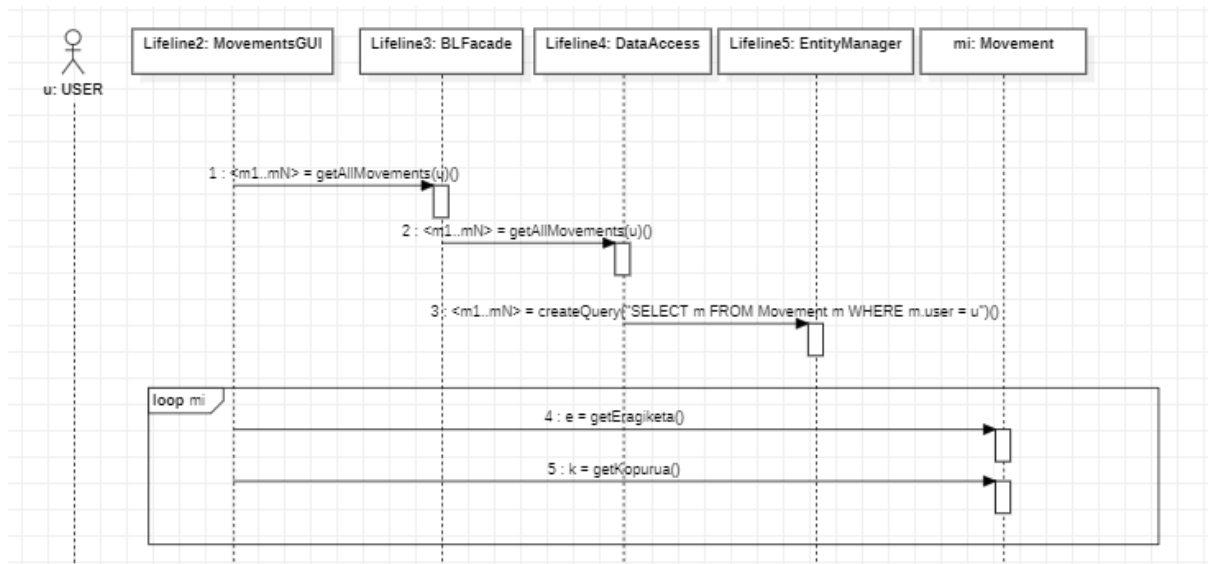


Mugimenduak ikusi

Gertaera fluxuak

Basic flow

1. Sistema eragiketa mota eta diru kopuruaren arabera lista bat erakusten du.
2. Erregistratua bere mugimenduak ikusten ditu.



SOFTWARE INGENIARITZA

Dirua sartu edo atera

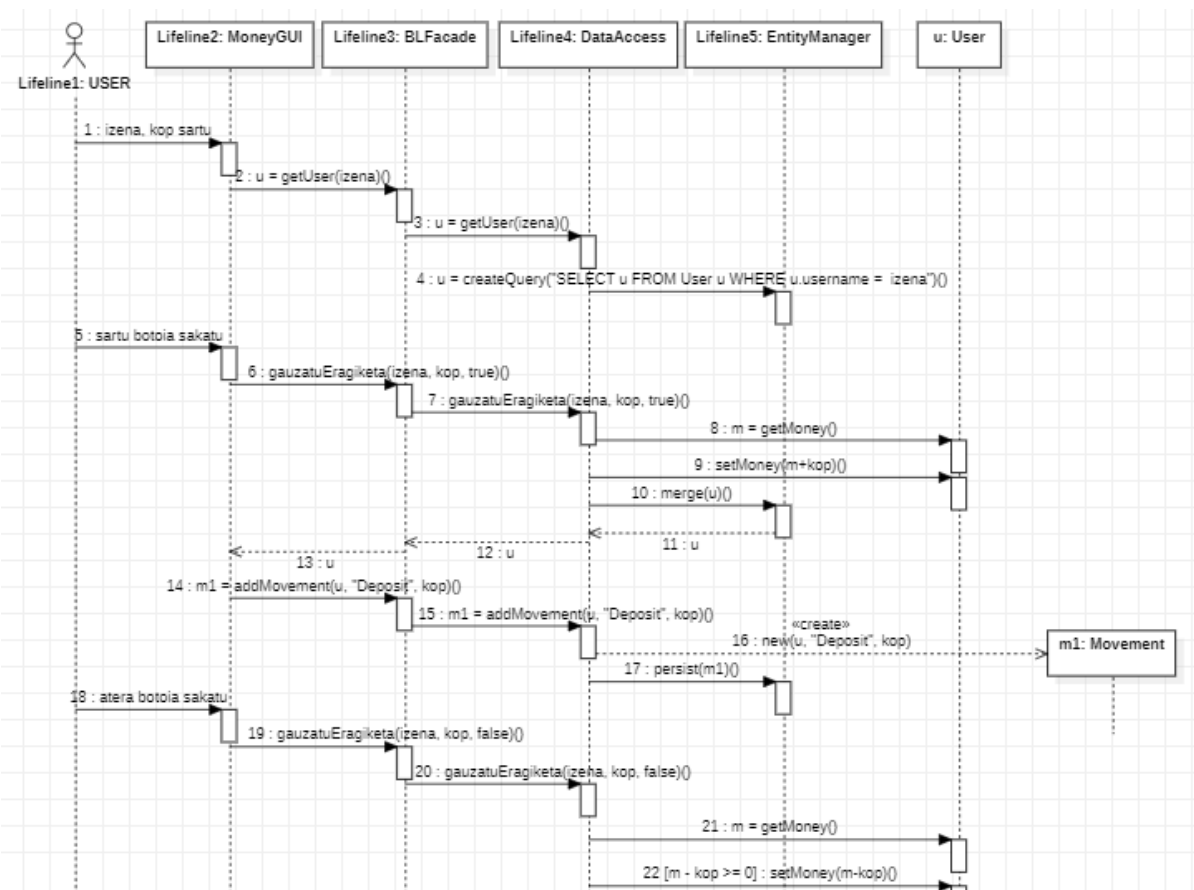
Gertaera fluxuak

Basic flow

1. Erregistratuak izena eta diru kopuru bat sartzen du.
2. Erregistratuak sartu edo atera botoia sakatu.
3. Sistemak sartu botoia sakatzerakoan, sartutako diru kopurua erabiltzaileari gehitu eta sortutako diru-mugimendua gorde.
4. Sistemak atera botoia sakatzerakoan, sartutako diru kopurua erabiltzaileari kendu eta sortutako diru-mugimendua gorde.
5. Erregistratuak mugimenduak botoia sakatu.
6. Sistemak *MovementsGUI* pantailara bideratu.

Alternative flow

1. Diruaren laukian ez da zenbaki egoki bat sartu. Sistemak abisua ematen du.
2. Atera nahi den diru kopurua orain dagoena baino handiagoa da. Sistemak abisua ematen du.




```

sequenceDiagram
    participant L6 as Lifeline6: MovementsGUI
    participant M2 as m2: Movement
    participant L8 as Lifeline8: MovementsGUI

    L6->>L6: 32 : tugimenduak botoia sakatu
    activate L6
    L6->>M2: «create» 30 : new(u, "Withdrawal", kop)
    deactivate L6
    activate M2
    M2->>L8: 31 : persist(m2)()
    deactivate M2
    activate L8
    L8->>L6: 29 : m2 = addMovement(u, "Withdrawal", kop)()
    deactivate L8
    activate L6
    L6->>L6: 28 : m2 = addMovement(u, "Withdrawal", kop)()
    deactivate L6
    L6->>L6: 27 : u
    L6->>L8: 26 : u
    L6->>L8: 25 : u
    L8->>L6: 24 : merge(u)()
    L8->>L6: 23 [m - kop < 0] : setMoney(0)()
    deactivate L8
    deactivate L6

```

Basic flow

-
- ```

sequenceDiagram
 participant User as Lifeline1: USER
 participant BaloraGUI as Lifeline2: BaloraGUI
 participant BLFacade as Lifeline3: BLFacade
 participant DataAccess as Lifeline4: DataAccess
 participant EntityManager as Lifeline5: EntityManager
 participant User as u: User

 User->>BaloraGUI: 1: izena, balorazioa sartu
 activate BaloraGUI
 BaloraGUI->>BLFacade: 2: u = getUser(izena())
 deactivate BaloraGUI
 activate BLFacade
 BLFacade->>DataAccess: 3: u = getUser(izena())
 deactivate BLFacade
 activate DataAccess
 DataAccess->>DataAccess: 4: u = createQuery("SELECT u FROM User u")()
 deactivate DataAccess
 DataAccess->>EntityManager: 5: b = getBalorazioa()
 deactivate DataAccess
 activate EntityManager
 EntityManager->>User: 6: setBalorazioa(b + balorazioa())
 deactivate EntityManager
 activate User
 User->>User: 7: k = getBalkop()
 deactivate User
 activate User
 User->>User: 8: setBalkop(k+1())
 deactivate User
 activate User
 User->>EntityManager: 9: merge(u)()
 deactivate User
 activate EntityManager
 EntityManager->>DataAccess: 10: merge(u)()
 deactivate EntityManager
 activate DataAccess
 DataAccess->>BaloraGUI: 11: sartu()
 deactivate DataAccess
 deactivate BaloraGUI

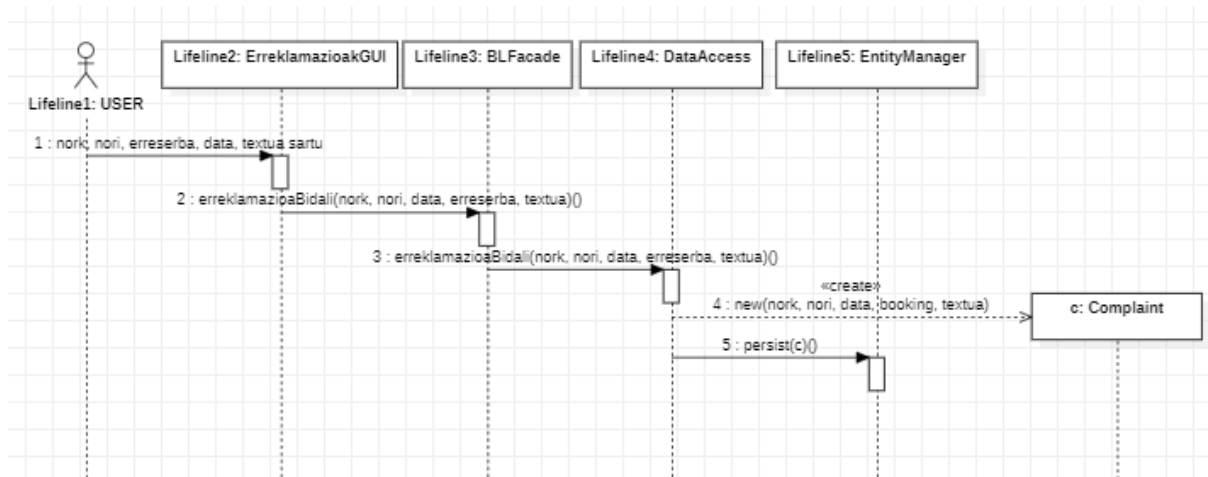
```

## Erreklamazioak jarri

### Gertaera fluxuak

#### Basic flow

1. *Erregistratua* nork, nori, gaur, booking eta textua.
2. *Sistemak* sartutako datuen arabera erreklamazioa sortu eta gorde.

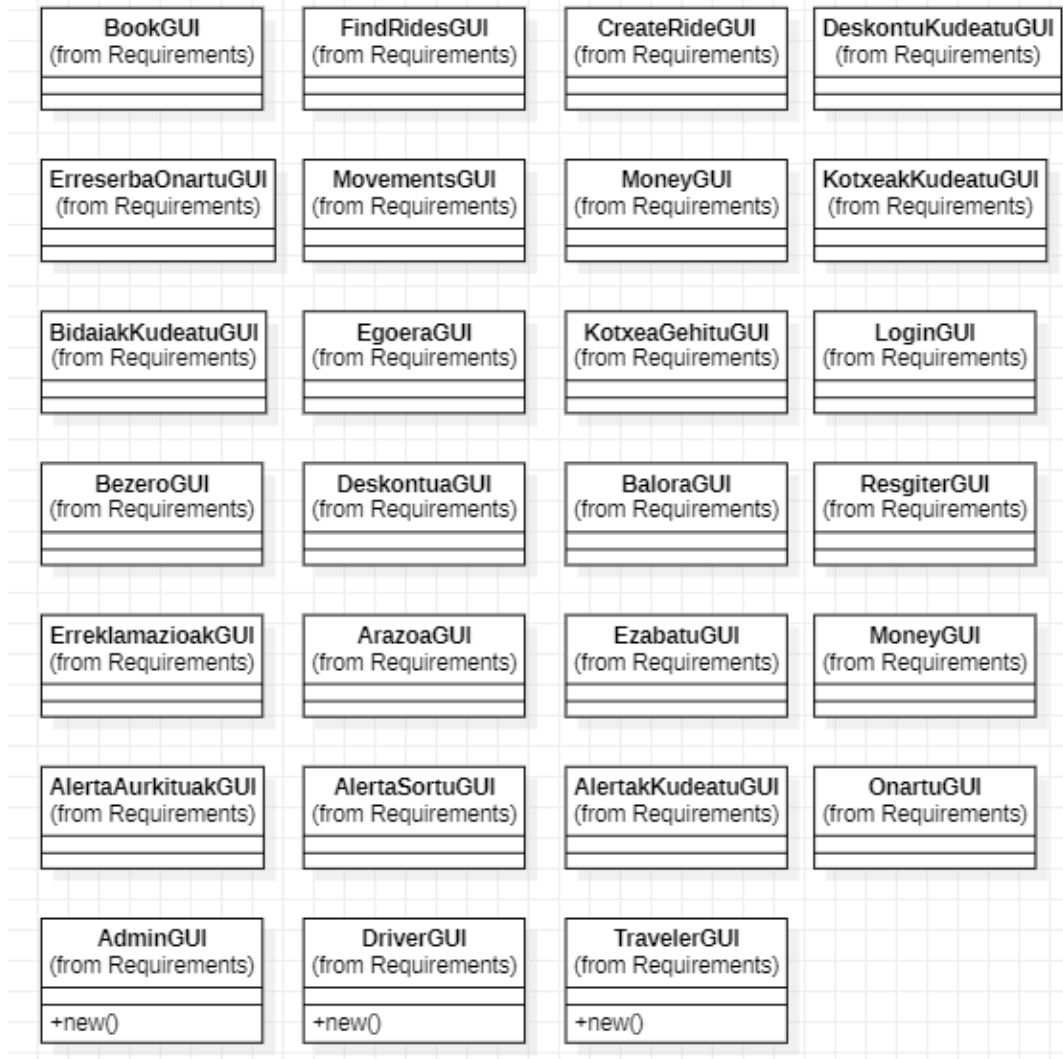


## SOFTWARE INGENIARITZA

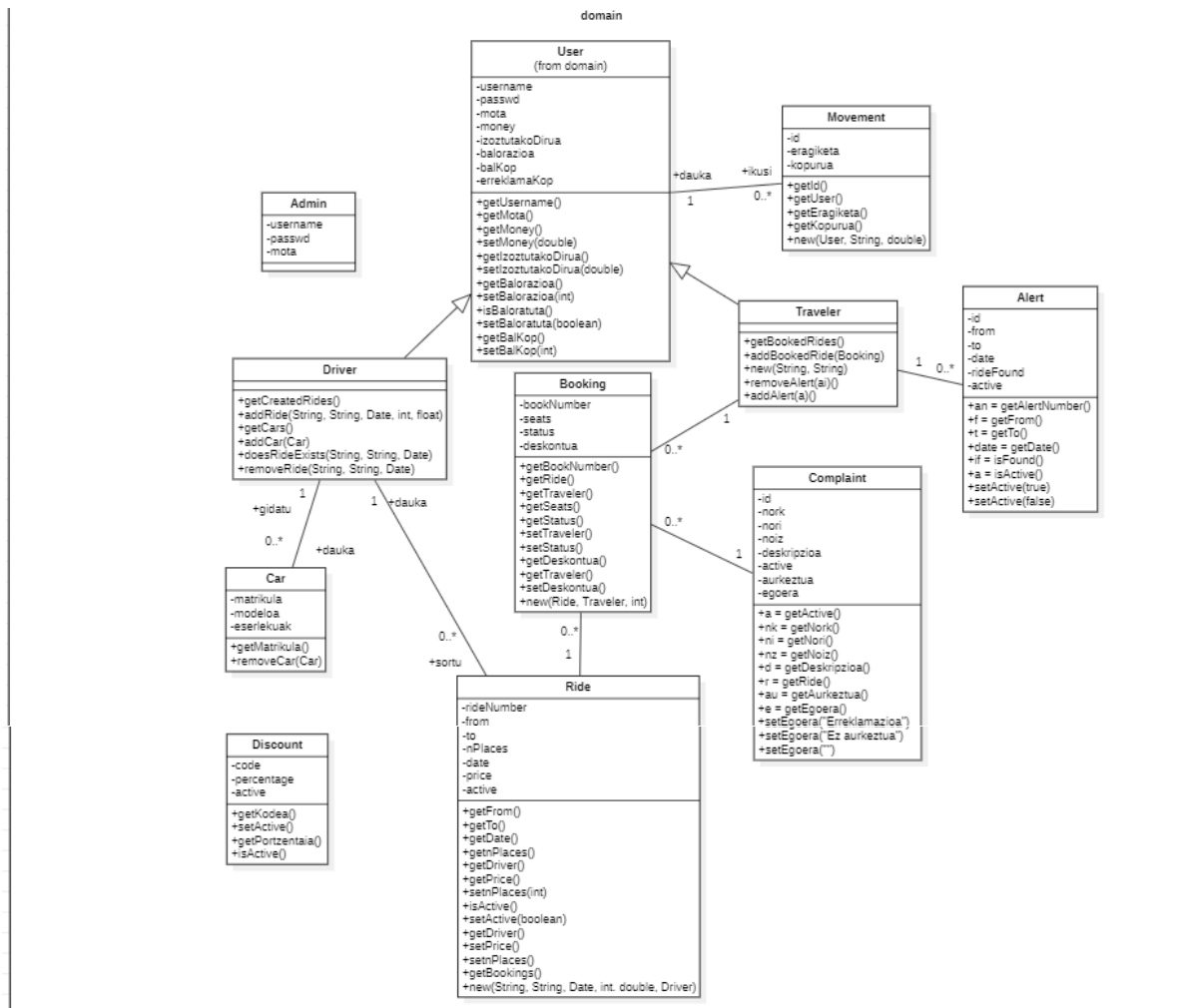
### ○ Klase diagrama osoa

| BLFacade                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | DataAccess                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> +&lt;d1..dN&gt; = getDepartCities() +&lt;a1..aN&gt; = getArrivalCities(di)() +&lt;m1..mN&gt; = getThisMonthDatesWithRides(di, ai, today)() +&lt;r1..rN&gt; = getRides(di, ai, mi)() +m = getActualMoney(izena)() +gauzatuEragiketa(izena, kop, auk) +addCar(matrikula, modelo, eserlekuak) +&lt;m1..mN&gt; = getAllMovements(izena)() +cancelRide(r) +&lt;r1..rN&gt; = getRidesByDriver(izena)() +&lt;b1..bN&gt; = getBookedRides(izena)() +desk = getDesk(deskontua)() +d = getDriver(izena)() +&lt;b1..bN&gt; = getBookingFromDriver(izena)() +erreserbaKudeatu(bi, "Onartu") +erreserbaKudeatu(bi, "Ezetzatu") +erreserbaZehaztu(bi, "Egina") +erreserbaZehaztu(bi, "Bertan Behera") +b = bookRide(izena, ri, eser, deskontua)() +gauzatuEragiketa(name, p, true)() +m1 = addMovement(t, "UnfreezeCompleteT", 0)() +m3 = addMovement(t, "UnfreezeNotComplete", p)() +c = getKotxeByMatrikula(matr)() +&lt;m1..mN&gt; = getThisMonthDatesWithRides(departCity, ArrivalCity, today)() +r = createRide(departCity, arrivalCity, mi, e, prezioa, izena)() +updateBooking(bi)() +m1 = addMovement("BookDeny", p)() +deleteCar(c)() +&lt;c1..cN&gt; = getAllComplaints() +u = getUser(nk)() +&lt;m1..mN&gt; = getAllMovements(u)() +u = getUser(izena)() +gauzatuEragiketa(izena, kop, true)() +m1 = addMovement(u, "Deposit", kop)() +gauzatuEragiketa(izena, kop, false)() +m2 = addMovement(u, "Withdrawal", kop)() +t = getTraveler(izena)() +erreklamazioaBidali(nork, nori, data, erreserba, textua)() +&lt;u1..uN&gt; = getUserList() +userez(ui)() +deskberria(d)() +&lt;d1..dN&gt; = getDiscount() +ezades(di)() +setdis(di)() +addDriver(izena, pass)() +addTraveler(izena, pass)() +u = isRegistered(izena, pass)() +mota = getMotaByUsername(izena)() +getComplaintsByBook(bi)() +updateBooking(bi)() +m3 = addMovement(t, "UnfreezeCompleteT", 0)() +c = getComplaintsByBook(bi)() +erreklamazioaBidali(nori, nork, gaur, booking, "Ez da agertu", false)() +&lt;a1..aN&gt; = getAlertsByUsername(izena)() +updateAlert(ai) +deleteAlert(an) +b = createAlert(a)() +m1 = addMovement(t, "BookFreeze", (p-deskontua)*eser)() +m1 = addMovement(t, "BookDeny", (p-desk)*se)() +gauzatuEragiketa(name, (p-desk)*e, true)() +m2 = addMovement(d, "UnfreezeCompleteD", (p-desk)*e)() +m4 = addMovement(d, "UnfreezeCompleteD", (p-desk)*e)() </pre> | <pre> +&lt;d1..dN&gt; = getDepartCities() +&lt;a1..aN&gt; = getArrivalCities(di)() +&lt;m1..mN&gt; = getThisMonthDatesWithRides(di, ai, today)() +&lt;r1..rN&gt; = getRides(di, ai, mi)() +m = getActualMoney(izena)() +gauzatuEragiketa(izena, kop, auk) +addCar(matrikula, modelo, eserlekuak) +&lt;m1..mN&gt; = getAllMovements(izena)() +cancelRide(r) +&lt;r1..rN&gt; = getRidesByDriver(izena)() +&lt;b1..bN&gt; = getBookedRides(izena)() +desk = getDesk(deskontua)() +d = getDriver(izena)() +&lt;b1..bN&gt; = getBookingFromDriver(izena)() +erreserbaKudeatu(bi, "Onartu") +erreserbaKudeatu(bi, "Ezetzatu") +erreserbaZehaztu(bi, "Egina") +erreserbaZehaztu(bi, "Bertan Behera") +b = bookRide(izena, ri, eser, deskontua)() +gauzatuEragiketa(name, p, true)() +m1 = addMovement(t, "UnfreezeCompleteT", 0)() +m3 = addMovement(t, "UnfreezeNotComplete", p)() +c = getKotxeByMatrikula(matr)() +&lt;m1..mN&gt; = getThisMonthDatesWithRides(departCity, ArrivalCity, today)() +r = createRide(departCity, arrivalCity, mi, e, prezioa, izena)() +updateBooking(bi)() +m1 = addMovement("BookDeny", p)() +deleteCar(c)() +&lt;c1..cN&gt; = getAllComplaints() +u = getUser(nk)() +&lt;m1..mN&gt; = getAllMovements(u)() +u = getUser(izena)() +gauzatuEragiketa(izena, kop, true)() +m1 = addMovement(u, "Deposit", kop)() +gauzatuEragiketa(izena, kop, false)() +m2 = addMovement(u, "Withdrawal", kop)() +t = getTraveler(izena)() +erreklamazioaBidali(nork, nori, data, erreserba, textua)() +&lt;u1..uN&gt; = getUserList() +userez(ui)() +deskberria(d)() +&lt;d1..dN&gt; = getDiscount() +ezades(di)() +setdis(di)() +addDriver(izena, pass)() +addTraveler(izena, pass)() +u = isRegistered(izena, pass)() +mota = getMotaByUsername(izena)() +getComplaintsByBook(bi)() +updateBooking(bi)() +m3 = addMovement(t, "UnfreezeCompleteT", 0)() +c = getComplaintsByBook(bi)() +erreklamazioaBidali(nori, nork, gaur, booking, "Ez da agertu", false)() +&lt;a1..aN&gt; = getAlertsByUsername(izena)() +deleteAlert(an) +updateAlert(ai) +b = createAlert(a)() +m1 = addMovement(t, "BookFreeze", (p-deskontua)*eser)() +m1 = addMovement(t, "BookDeny", (p-desk)*se)() +gauzatuEragiketa(name, (p-desk)*e, true)() +m2 = addMovement(d, "UnfreezeCompleteD", (p-desk)*e)() +m4 = addMovement(d, "UnfreezeCompleteD", (p-desk)*e)() </pre> |

## SOFTWARE INGENIARITZA



## SOFTWARE INGENIARITZA



---

## SOFTWARE INGENIARITZA

- **Implementazioa**

**public void** initializeDB(): Programa hasterakoan objektu desberdinak inzializatzeko erabiliko dugu hau. Modu honetan ez da izango beharrezkoa objektu batzuk sortzea aldiro, adibidez gidariak, bidaiariak, kotxeak...

**public** User getUser(String erab): Erabiltzaile baten izena parametro bezela pasaz, honek User-a itzuliko du.

**public double** getActualMoney(String erab): Erabiltzaile baten izena pasaz, honek erabiltzaile horrek duen dirua itzuliko du.

**public boolean** isRegistered(String erab, String passwd): Erabiltzaile izen bat eta pasahitza pasaz, ea hau erregistratua dagoen bueltatuko du boolean baten bitartez.

**public** Driver getDriver(String erab): Erabiltzaile baten username pasaz, honek Driver-a itzuliko du.

**public** Traveler getTraveler(String erab): Lehengo metodoaren berdina egingo du, baina honek Driver-a beharrean Traveler-a itzuliko du.

**public** Admin getAdmin(String erab): Berriz ere berdina egingo du, baina honek Admin-a itzuliko du.

**public** String getMotaByUsername(String erab): Erabiltzaile baten username-a pasaz, honek ze erabiltzaile mota den itzuliko du. Gidari, bidaiari edo admin.

**public boolean** addDriver(String username, String password): Metodo hau erabiliko dugu erabiltzaile berri bat erregistratzerakoan bidaiari bezela. Bidaiari berri bat sortuko du pasa dituen parametroen bidez.

---

## SOFTWARE INGENIARITZA

**public boolean** addTraveler(String username, String password): Lehenagoko metodoaren berdina egiten du, baina kasu honetan Traveler berri bat sortuko du.

**public boolean** gauzatuEragiketa(String username, **double** amount, **boolean** b): Dirua dakarten eragiketeentzako erabiltzen da hau. Zein erabiltzaileari eragin, diru kantitatea eta ea dirua sartu edo atera behar den sartuko dugu metodoa gauzatu ahal izateko.

**public boolean** bookRide(String username, Ride ride, **int** seats, **double** esk): Metodo honen bitartez Traveler-ak bidaiak erreserbatu ahal izango du.

**public** List<Movement> getAllMovements(User user): User bat emanez, honek izan dituen mugimendu guztiak itzuliko ditu lista batean.

**public void** addMovement(User user, String eragiketa, **double** amount): Emandako user-ari mugimendu bat gehitu.

**public** List<Booking> getBookedRides(String username): User batek dituen erreserba guztiak itzuliko ditu lista batean.

**public void** updateTraveler(Traveler traveler): Traveler batean aldaketa egin ondoren, hau datu basean gorde dadin egiten dugu metodo hau.

**public void** updateDriver(Driver driver): Lehengo metodo berdina da, baina driver-arekin.

**public void** updateUser(User user): Lehengo berdina da, baina User batekin.

**public** List<Booking> getPastBookedRides(String username): Egunerako egindako erreserbak itzuliko ditu lista batean.

---

SOFTWARE INGENIARITZA

**public void** updateBooking(Booking booking) : Bookin bat eguneratzen du aldaketa baten bat egin ondoren.

**public** List<Booking> getBookingFromDriver(String username) : Driver baten bidai desberdinetan dituen erreserbak itzuliko ditu lista batean.

**public** List<Ride> getRidesByDriver(String username) : Gidari batek dituen bidaiak itzuliko ditu lista batean.

**public void** cancelRide(Ride ride) : Bidai bat pasaz, hau ezabatuko du.

**public boolean** addCar(String username, Car kotxe) : Kotxe berri bat gehituko du driver-ari.

**public boolean** isAdded(String username, String matr) : Lehengo metodoarekin zerikusia dauka. Lehengoa erabiltzerakoan honi deitzen zaio ikusteko ea dagoeneko gehitua daukan kotxea itzuliko du.

**public** Car getKotxeByMatrikula(String matr) : Parametro bezela pasa den matrikularen kotxea bueltatuko da.

**public void** deleteCar(Car car) : Kotxe bat ezabatuko du Driver-aren kotxeen listatik.

**public boolean** erreklamazioaBidali(String username1, String username2, Date gaur, Booking book, String textua, **boolean** aurk) : Erreklamazio berri bat egiteko erabiltzen da metodo hau.

**public void** updateComplaint(Complaint erreklamazioa) : Erreklamazio bat berrituko du aldaketa bat egiterakoan.



---

SOFTWARE INGENIARITZA

**public void** createDiscount(Discount di) : Deskontu bat sortzen duen metodoa da.

**public** List<Discount> getAllDiscounts() : Lista batean itzuliko ditu dauden diskontu guztiak.

**public void** deleteDiscount(Discount dis) : Deskontu bat pasaz, honek deskontuen listatik ezabatuko du.

**public void** updateDiscount(Discount dis) : Deskontu bat aldatu bada, metodo hau aplikatuz aldaketan datu basean gordeko dira.

**public** Discount getDiscount(String desk) : Deskontuaren izenaren bitartez, deskontua itzuliko du.

**public** List<User> getUserList() : User guztiak itzuliko ditu lista batean.

**public void** deleteUser(User us) : Parametro bidez pasa den user-a ezabatuko da.

**public** List<Alert> getAlertsByUsername(String username) : User batek dituen alertak itzuliko ditu lista batean.

**public** Alert getAlert(int alertNumber) : Alerta zenbakiaren bitartez, alerta eskuratu eta bueltatuko da.

**public void** updateAlert(Alert alert) : Alertar batean aldaketa bat egin ondoren, metodo hau erabiliko da alertaren eguneraketa datu basean gordetzeko.

**public boolean** updateAlertaAurkituak(String username) : Alerta aurkitzerakoan metodo hau erabiliko da.

**public boolean** createAlert(Alert newAlert) : Alerta bat sortuko duen metodoa da.

## SOFTWARE INGENIARITZA

**public boolean** deleteAlert(**int** alertNumber) : Alerta zenbaki bat parametro bezela pasata, honi dagokion alerta ezabatuko du.

**public** Complaint getComplaintsByBook(Booking **bo**) : Erreserba batek dituen erreklamaioak itzuliko ditu.

### • Ondorioak

Proiektu honetan lan egin ondoren ondorio hauek ateratu ditugu.

- Denon artean proiektu hau aurrera eramatea lortu dugu, talde bezala ondo moldatu garelako eta beestengan laguntza jaso dugulako.
- Ikasgai honetan erabilitako iterazioen metodoa interesgarria iruditu zaigu proiektua gutxinaka osatzen eta hobetzen joateko.
- Hurrengo urteko ikasleei gomendatuko genieke aldez aurretik proiektuaren diseinu osoa zehazki nola izango den planifikatzea, programatzerako garaian denbora ez galtzeko.
- Proiektuaren garapena nahiko konplexua izan da, hasieran ez genekielako ezta botoi bat nola ipintzen zen eta azkenean aplikazio handi eta funtzional bat garatu dugulako.
- Gure iritziz, aplikazioaren garapena zaila izan da, batez ere, kode aldetik gure kabuz ikasi izan dugulako metodoen implementazioa eta ez genekien nola funtzionatzen zuten. Aipatzekoa da, datu basearen kudeaketa nahiko konplexua dela eta gure eskura informazio gehiago edukitzea gustatuko litzaigukeen.
- Baita ere, diseinu aldetik, sekuentzia diagramen azalpena ikasgaiaren hasieratik jasotzea hobeagoa izango zatekeen.
- Bukatzeko, Irakasgai bezala erakargarria da eta hilabete hauetan egindako lanarekin pozik gaude.

### • Bideoaren URL-a

<https://youtu.be/JyPL2yxF3aI>

### • Kodearen zip fitxategia

(Entregaren atazan)