

# PROIEKTUAREN DOKUMENTAZIOA

Egileak:

Aitor Seises

Eneko Santos

Malen Bilbatua

Luken Aizpiolea

# AURKIBIDEA

• Web Zerbitzua eta Internalizazioa.....	3
• Sarrera.....	3
• Eskakizun bilketa eta diseinua.....	3
◦ Entitateak.....	3
◦ Domeinuaren eredua eta bere erlazioak.....	4
◦ Erabilpen kasuen eredua.....	5
◦ Erabilpen kasu bakoitzeko bere gertaera fluxua eta interfaze grafikoa.....	5
• Inplementazioa.....	26
• Ondorioak.....	30
• Bideoaren URL-a.....	30
• Kodearen zip fitxategia.....	30

- **Web Zerbitzua eta Internalizazioa**

Programazio lengoai ezberdinak erabili ditugu. Java erabili dugu proiektuaren zatirik handiena egiteko, eta SQL erabili dugu datu-basetik atzipenak egiteko.

Beste alde batetik, ez ditugu web zerbitzuak erabili.

- **Sarrera**

Proiektu honetan Eclipse ingurunean garatutako bidaiak partekatzeko aplikazio bat diseinatu eta inplementatu dugu.

Aplikazioak hainbat funtzionalitate ditu (geroago zehaztuko direnak), baina nagusienak “Bidaiak sortzea”, “Bidaiak bilatu” eta “Bidaiak erreserbatu” dira; gainera, bi rol desberdin daude erabiltzailearentzat: gidaria (bidaiak eskaintzen dituen) eta bidaiaria (bidaiak erreserbatzen dituen). Admin rola ere badago, baina erabiltzaile batek ezin du admin konturik sortu.

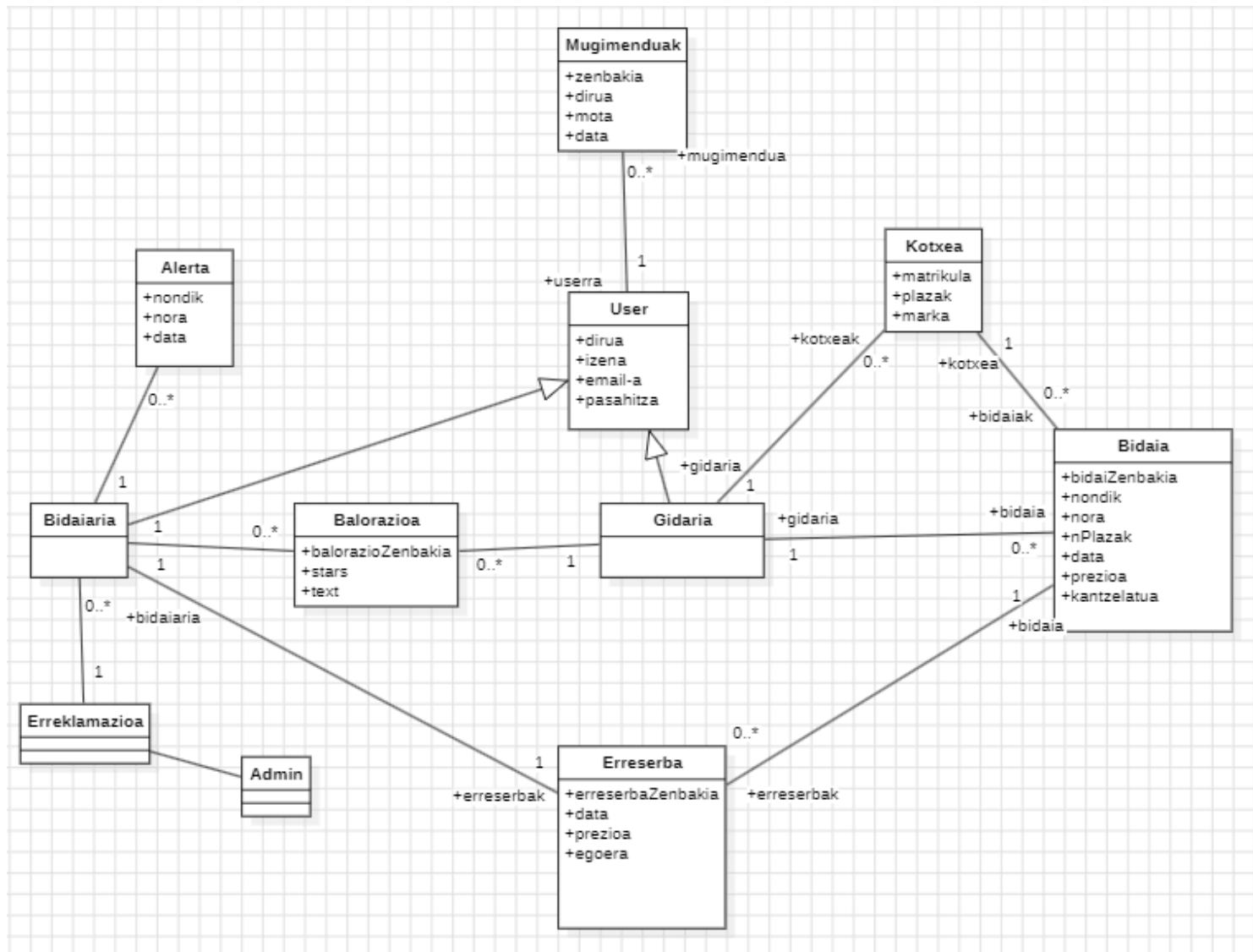
- **Eskakizun bilketa eta diseinua**

- **Entitateak**

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

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, lau entitate nagusiak *User*, *Traveler*, *Driver* eta *Admin* dira. *Traveler*, *Driver* eta *Admin* klaseek atributu berdin batzuk dituztenez, *User* klasearen umeak dira eta hainbat funtzionalitate konpartitzen dituzte.

- Domeinuaren ereduak eta bere erlazioak



Diseinatu dugun domeinuaren ereduari, 11 klase definitu ditugu: *Mugimenduak*, *User*, *Kotxea*, *Bidaia*, *Gidaria*, *Erreserba*, *Balorazioa*, *Bidaiaria*, *Alerta*, *Erreklamazioa* eta *Admin* .

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

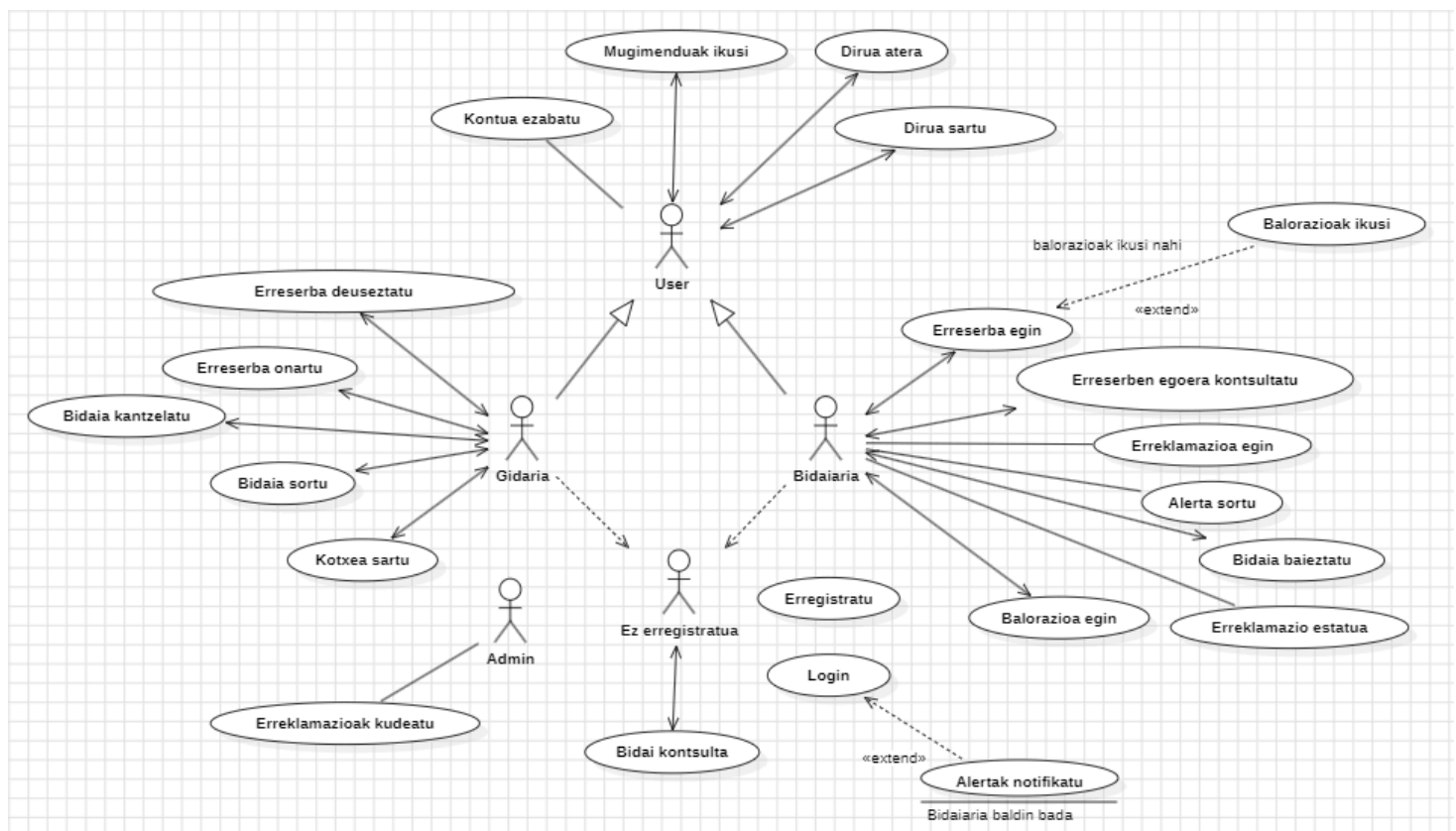
*Gidaria* klaseak *Kotxea* motako lista bat dauka eta *Kotxea* bakoitzak *Gidaria* bakarra gidatzeko aukera izango du. Horretaz gain, *Gidariak* sortutako bidaien lista bat dauka, *Bidaia*

izenarekin definitutakoa. *Bidaia* eta *Kotxea* ere lotuta daude, *Kotxe* bakoitzak hainbat bidai egin ditzake baina bidai bakoitza *kotxe* batek soilik egin dezake.

*Bidaiaria* klaseak hainbat *Alerta* kudeatzeko lista bat dauka eta *Alerta* bakoitza *Bidaiaria* bakar bati lotuta dago. *Bidaiaria* klaseak baita *Erreserba* klaseari lotuta dago, erreserbatutako bidaia guztiak gordetzeko. *Erreserba* klase hau *Bidaia* klasearekin lotuta dago baita, *Bidaia* bakoitzak hainbat *Erreserba* ditu eta *Erreserba* bakoitza *Bidaia* bati lotuta dago. Gainera, *Bidaiaria* eta *Gidaria*, *Balorazio*arekin lotuta daude.

Azkenik, *Bidaiaria* klasea *Erreklamazioa* klasearekin lotuta dago eta hau *Admin* klasearekin, erreklamazioak kudeatzeko.

○ Erabilpen kasuen eredu



○ Erabilpen kasu bakoitzeko bere gertaera fluxua eta interfaze grafikoa

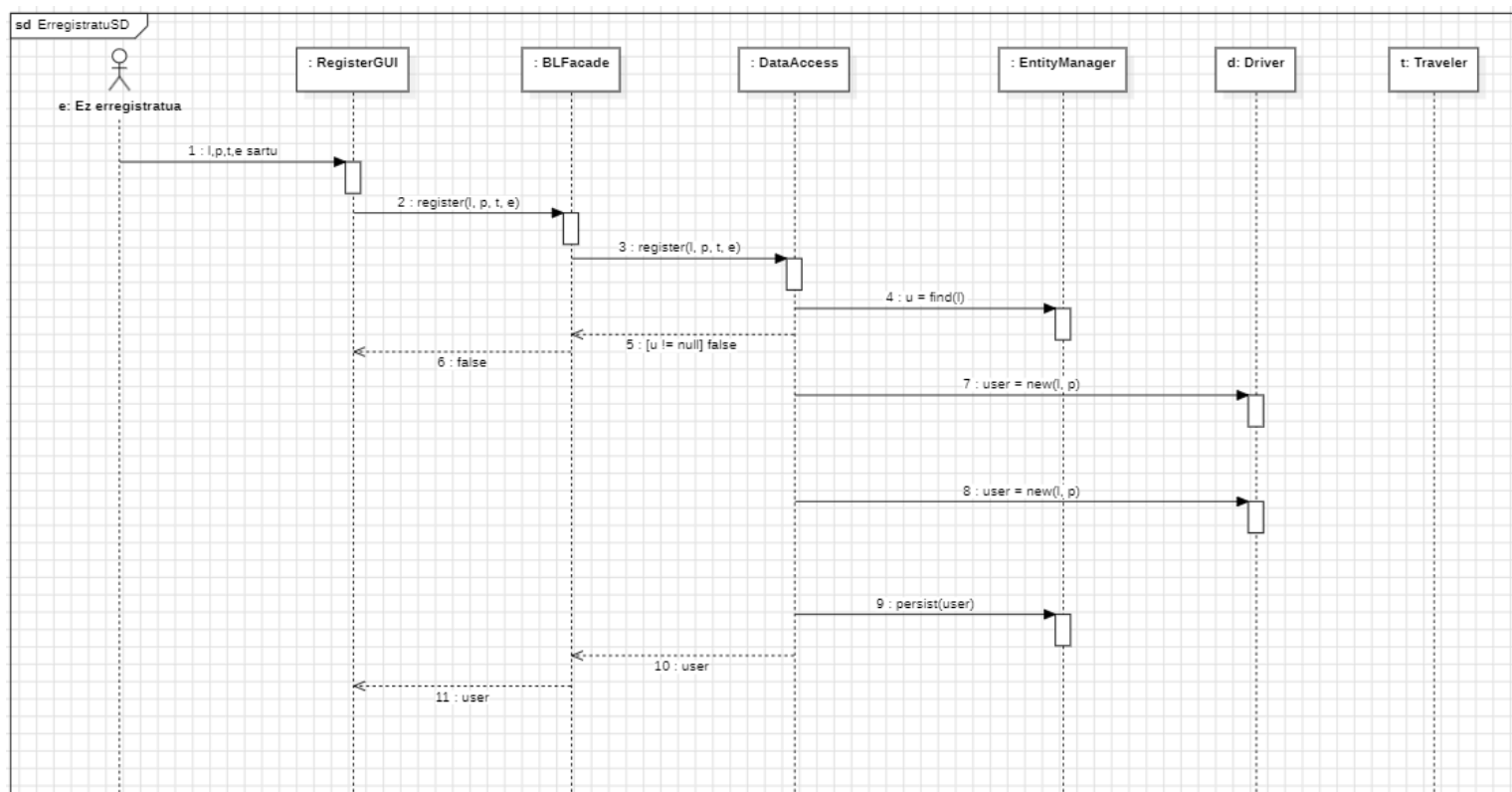
**ERREGISTRATU**

```

# Erregistroaren Gertaera Fluxuak
## Oinarrizko Gertaerak
1. *Ez-erregistratuak* **email**a, **izena** eta **pahahitza** sartzen ditu.
2. *Ez-erregistratuak* **erregistratzeko botoia** sakatzen du.
3. *Sistemak* kontu berri bat sortzen du sartutako datuekin eta **ez erregistratua** **user** bihurtzen da.

## Gertaera Alternatiboak
1. Kontua dagoeneko existitzen zen datu-basean. Errore mezu bat inprimatzen da.

```



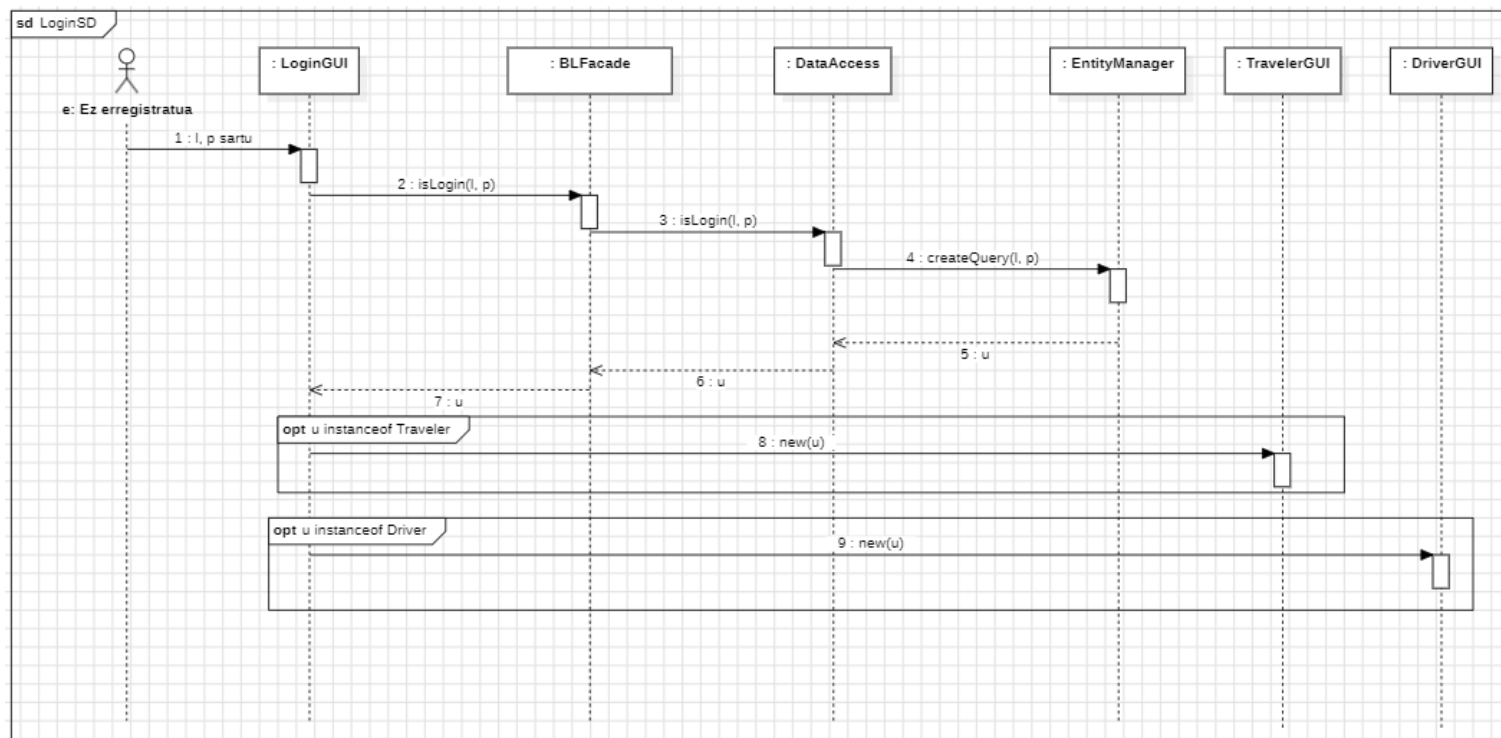
## LOGIN

```

# Login-aren Gertaera Fluxuak
## Oinarrizko Gertaerak
1. *Ez-erregistratuak* **izena** eta **pahahitza** sartzen ditu.
2. *Ez-erregistratuak* **login botoia** sakatzen du.
3. *Sistemak* datu horrek dituen kontua bilatzen du, eta **ez erregistratua** kontu horretako **user** bihurtzen da.

## Gertaera Alternatiboak
1. Kontua ez da existitzen datu-basean. Errore mezu bat inprimatzen da.
2. Pasahitza ez da zuzena. Berriro saiatzeko errore mezu bat inprimatzen da.

```



## BIDAI KONTSULTA

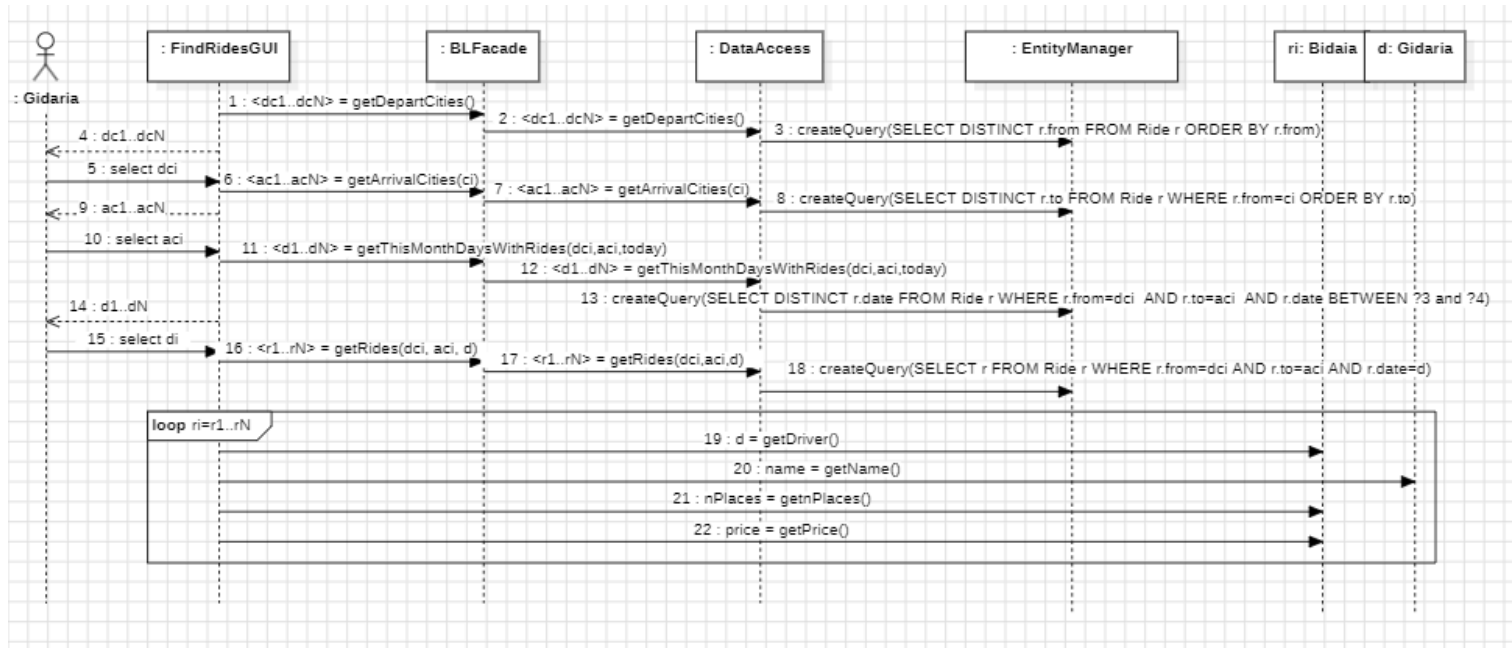
# Ibilaldien Kontsulta Gertarea Fluxuak

## Oinarrizko Gertaerak

1. \*Sistemak\* \*\*ibilaldiak\*\* irteten diren \*\*hiri\*\* guztiak erakusten ditu.
2. \*Gidariak\* irteerako \*\*hiria\*\* hautatzen du.
3. \*Sistemak\* hautatutako \*\*hiritik\*\* irteten diren \*\*ibilaldien\*\* helmuga \*\*hiri\*\* guztiak erakusten ditu.
4. \*Gidariak\* helmuga \*\*hiria\*\* hautatzen du.
5. \*Sistemak\* egutegian nabarmentzen ditu hautatutako hilabeteko \*\*irteera\*\* eta \*\*helmuga\*\* hirien arteko \*\*ibilaldiak\*\* dauden egunak.
6. \*Gidariak\* egutegian \*\*data\*\* bat hautatzen du.
7. \*Sistemak\* hautatutako \*\*irteera hiritik\*\* hautatutako \*\*helmuga hira\*\* doazen \*\*ibilaldiak\*\* erakusten ditu hautatutako \*\*daten\*\*.

## Gertaera Alternatiboak

1. Hautatutako \*\*daten\*\* ez dago \*\*ibilaldirik\*\*. \*\*Ibilaldien\*\* erakuslea hutsik dago.



## ALERTAK NOTIFIKATU

# Alertak Notifikatu Gertaera Fluxuak

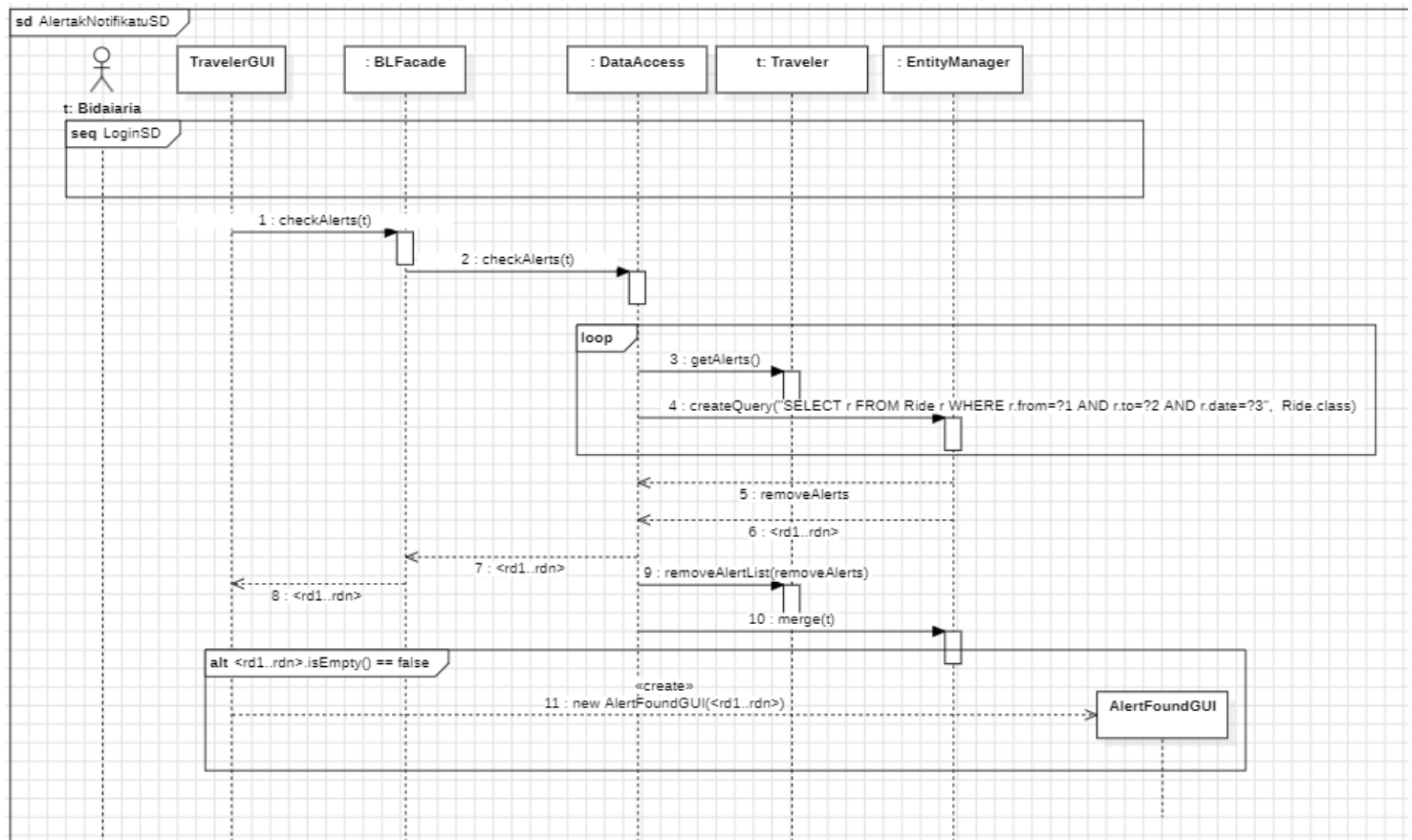
## Oinarrizko Gertaerak

1. **\*Sistemak\*** ea **\*\*bidaiari\*\*** horren **\*\*alertetako\*\*** **\*\*bidai\*\*** bat existitzen den begiratzeko du.

## Gertaera Alternatiboak

1. Alertetako bidaia existitzen da. GUI berri baten bidez notifikatzen da.



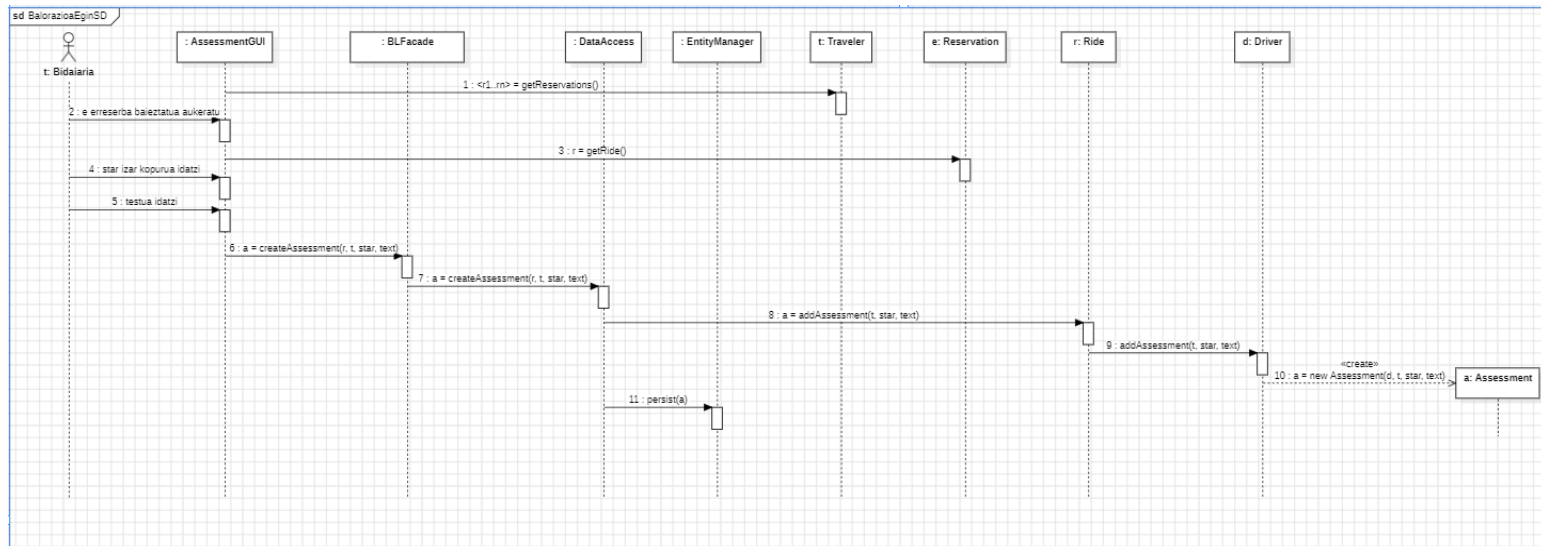


## BALORAZIOA EGIN

# Balorazioa egin Gertarea Fluxuak

## Oinarrizko Gertaerak

1. **\*Sistemak\*** **\*\*erreserba\*\*** **\*\*baieztatu\*\*** guztiak erakusten ditu.
2. **\*Bidaiariak\*** izar kopuruak aukeratzen ditu.
3. **\*Bidaiariak\*** balorazioa idazten du.
4. **\*Sistemak\*** **\*\*balorazio\*\*** bat sortzen du informazio horrekin eta **\*\*gidariari\*\*** esleitzen dio.



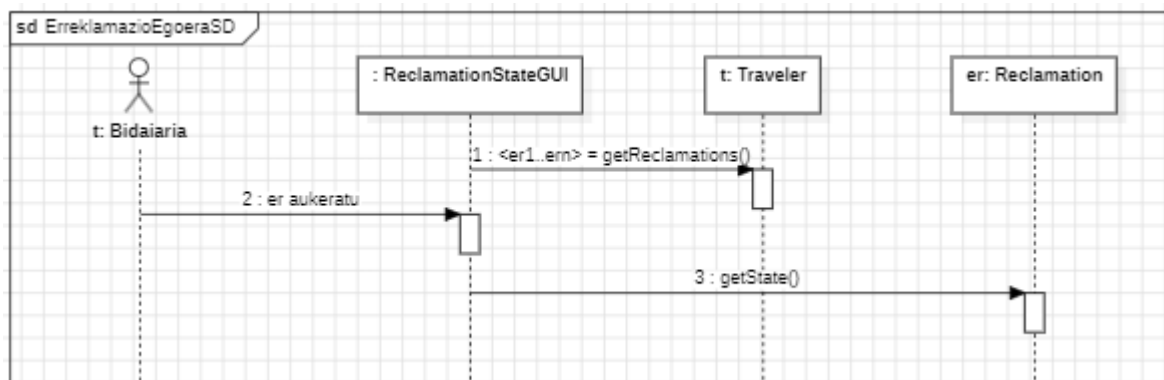
## ERREKLAMAZIOAREN EGOERA

```

# Erreklamazio egoera Gertarea Fluxuak
## Oinarrizko Gertaerak

1. *Sistemak* **bidaiari** horren **erreklamazio** guztiak erakusten ditu.
2. *Bidaiariak* **erreklamazio** bat aukeratzen du.
3. *Sistemak* erreklamazio horren egoera erakusten du.

```

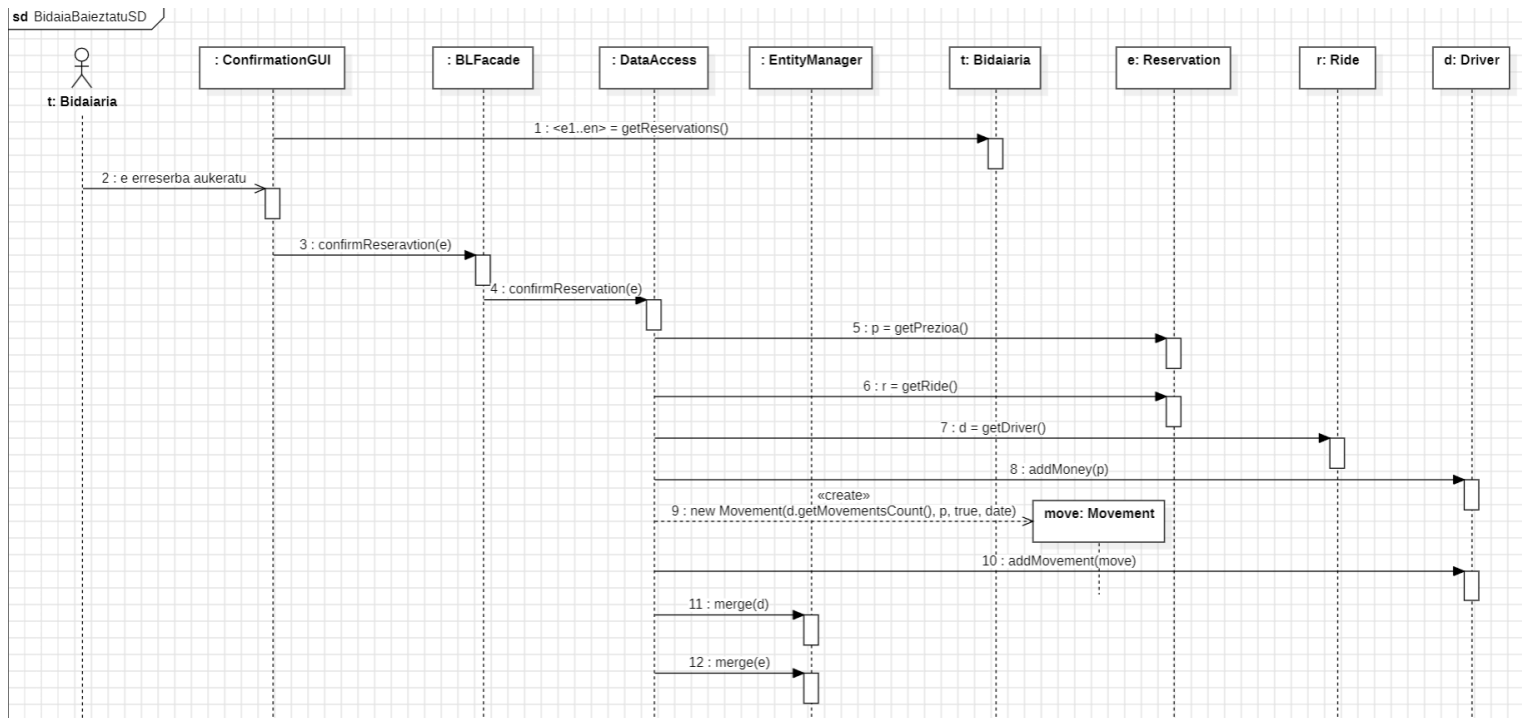


## BIDAIA BAIEZTATU

# Bidaia baieztatzearen Gertarea Fluxuak

## Oinarrizko Gertaerak

1. *\*Bidaiaia\**-k *\*\*erreserba\*\** aukeratzen du.
2. *\*Bidaiaia\**-k botoia sakatzen du
3. *\*Sistema\**-k erreserbaren *\*\*egorea\*\** baieztatua-ra aldatzen du.



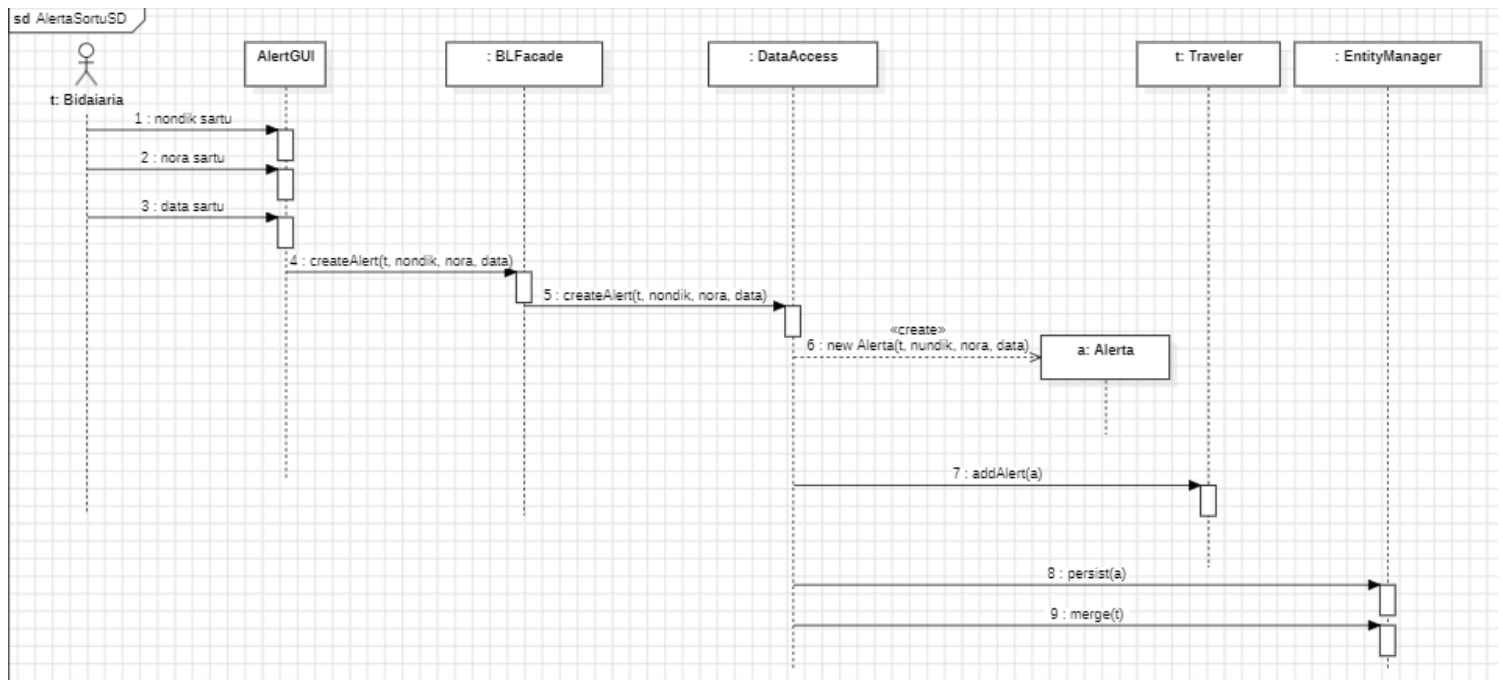
## ALERTA SORTU

```

# Alerta sortu Gertarea Fluxuak
## Oinarrizko Gertaerak

1. *Bidaiaariak* **bidai** baten nondik, nora eta data sartzen ditu.
2. *Sistemak* **alerta** bat sortzen du **bidai** horren ezaugarriekin.
3. *Sistemak* **alerta** hori **bidaiaari** horretara esleitzen du.

```

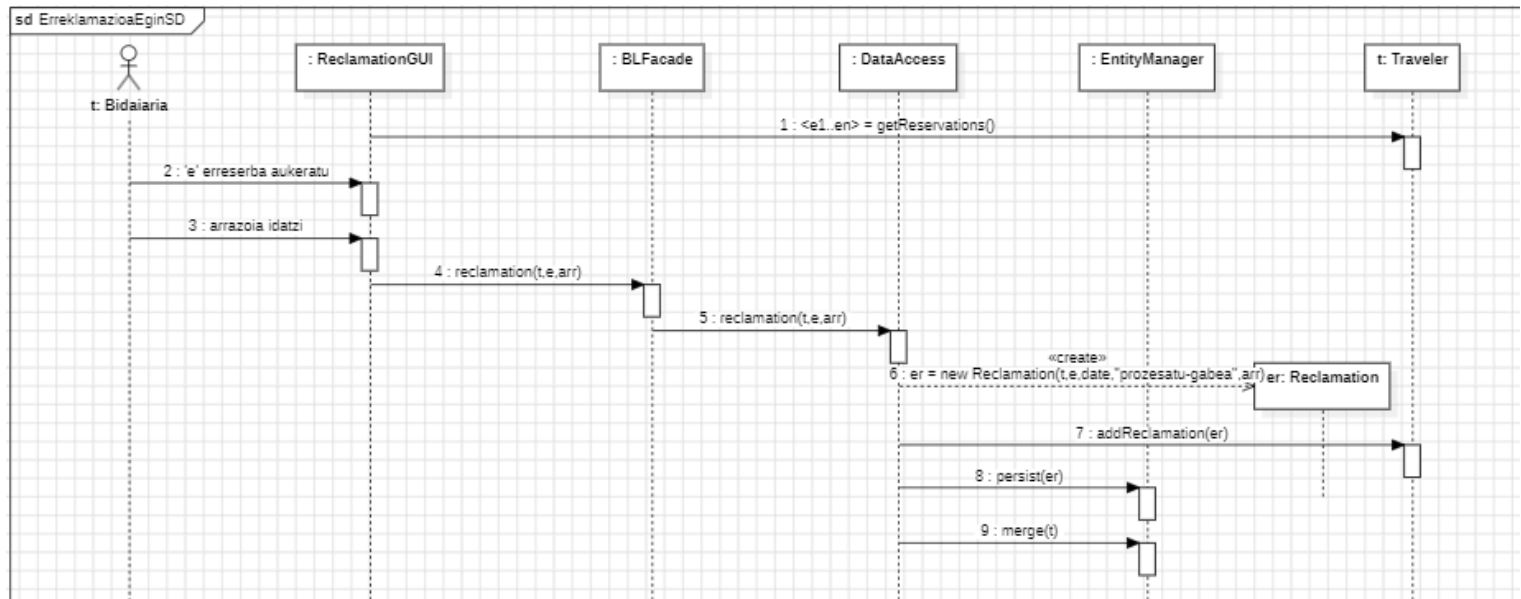


## ERREKLAMAZIOA EGIN

```

# Erreklamazioa egin Gertarea Fluxuak
## Oinarrizko Gertaerak

1. *Sistemak* **bidaiari** horren data hori baino lehen dituen **erreserba** guztiak erakusten ditu.
2. *Bidaiariak* **erreserba** bat aukeratzen du.
3. *Bidaiariak* zergatik nahi duen erreklamatu idazten du.
4. *Sistemak* **erreserbarekin** eta **zergatiarekin erreklamazio** bat sortzen du.
3. *Sistemak* **erreklamazio** hori **bidaiari** horri esleitzen dio.
  
```

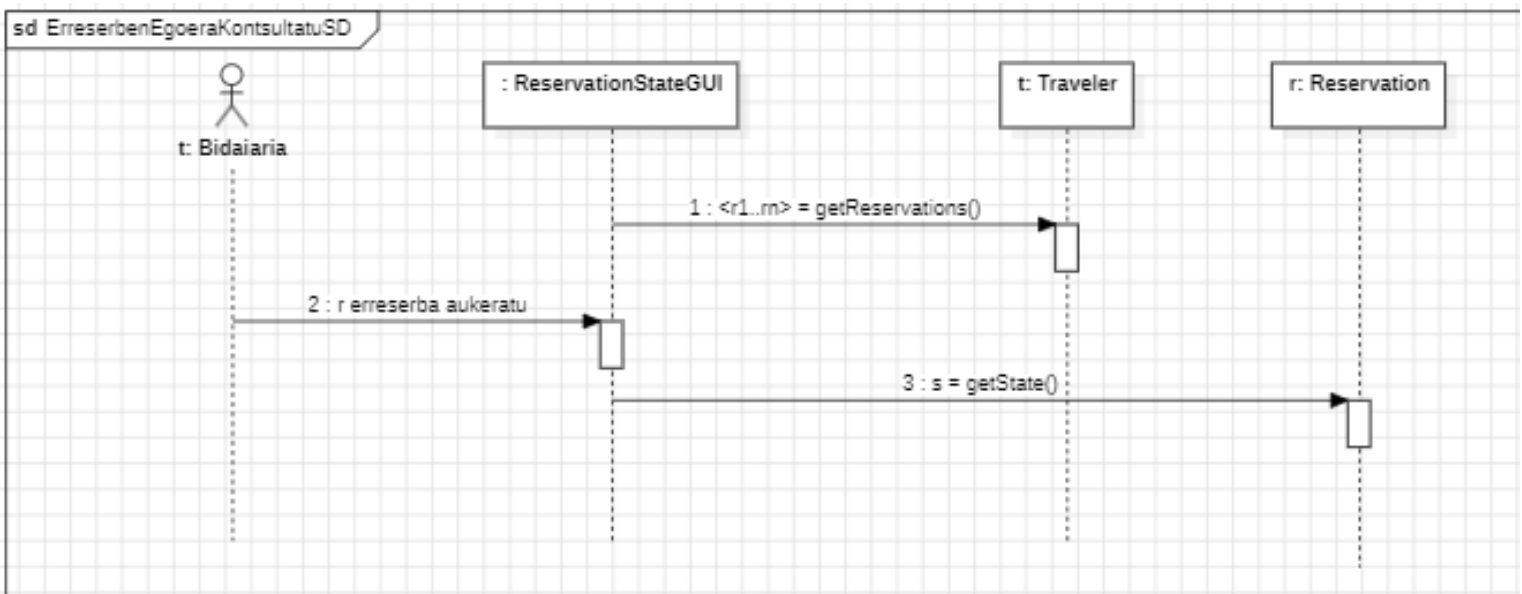


## ERRESERBEN EGOERA KONTSULTATU

# Egoeren kontultaren Gertarea Fluxuak

## Oinarrizko Gertaerak

1. *\*Bidaiaria\**-k *\*\*erreserba\*\** aukeratzen du.
2. *\*Sistema\**-k erreserbaren *\*\*egoreea\*\** pantailarazten du.

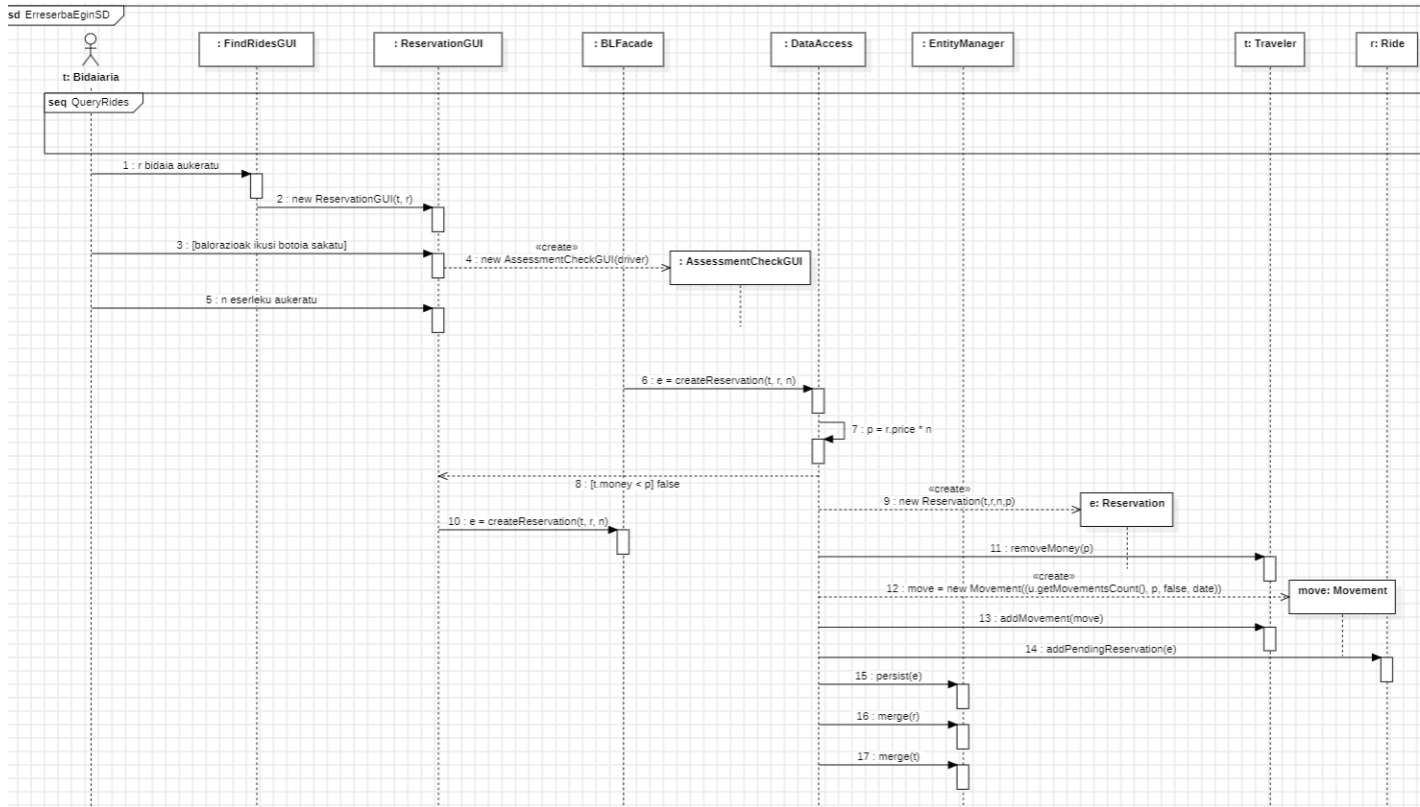


## ERRESERBA EGIN

```

# Ibilaldien Kontsulta Gertarea Fluxuak
## Oinarrizko Gertaerak
1. *Bidaiariak* nahi duen **bidaia** aukeratzen du.
2. *Sistemak* **erreserbatu** botoia gaitzen du.
3. *Bidaiariak* **erreserbatu** botoia sakatzen du.
4. *Sistemak* erreserba hori kostatzen duen dirua izozten du.
5. *Sistemak* **bidaiari** hori lista batean sartzen du. (gidaiaria aukeratzeko ea onartu edo deuseztatu)

## Gertaera Alternatiboak
1. *Bidaiariak* ez du dirurik. Ez da erreserba egiten.
2. Hautatutako **daten** ez dago **ibilaldirik**. **Ibilaldien** erakuslea hutsik dago.
3. *Bidaiariak* erreserbako gidaariaren balorazioak ikusi nahi ditu. Balorazioen pantaila agertzen da.
  
```



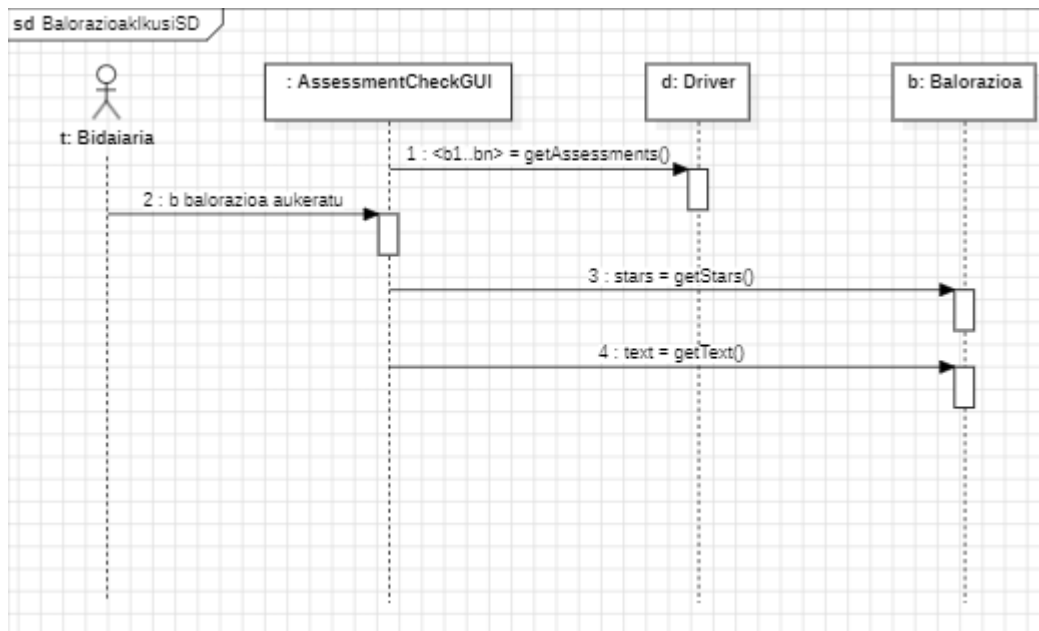
## BALORAZIOAK IKUSI

# Balorazioak ikusi Gertarea Fluxuak

## Oinarrizko Gertaerak

1. **\*Sistemak\*** **\*\*gidari\*\*** horren **\*\*balorazio\*\*** guztiak erakusten ditu.
2. **\*Bidaiaariak\*** **\*\*balorazio\*\*** bat aukeratzen du.
3. **\*Sistemak\*** balorazio horren izarrak eta testua erakusten du.



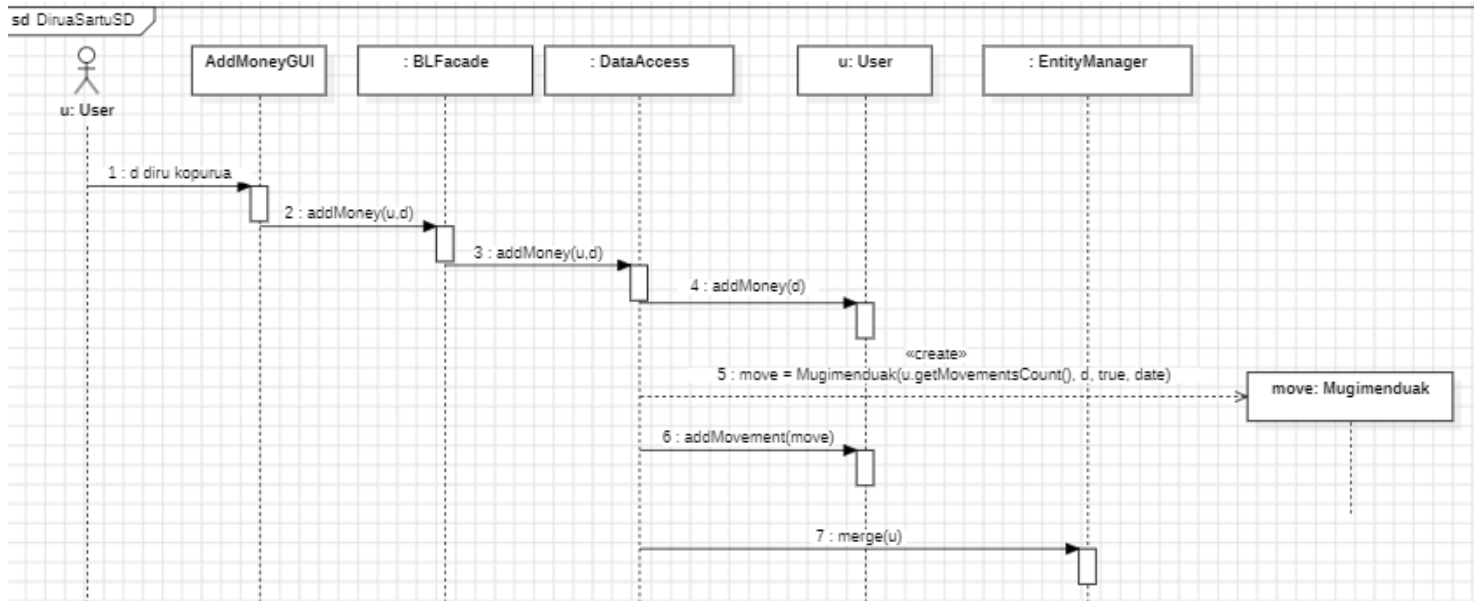


## DIRUA SARTU

```

# Dirua sartzeko Gertaera Fluxuak
## Oinarrizko Gertaerak

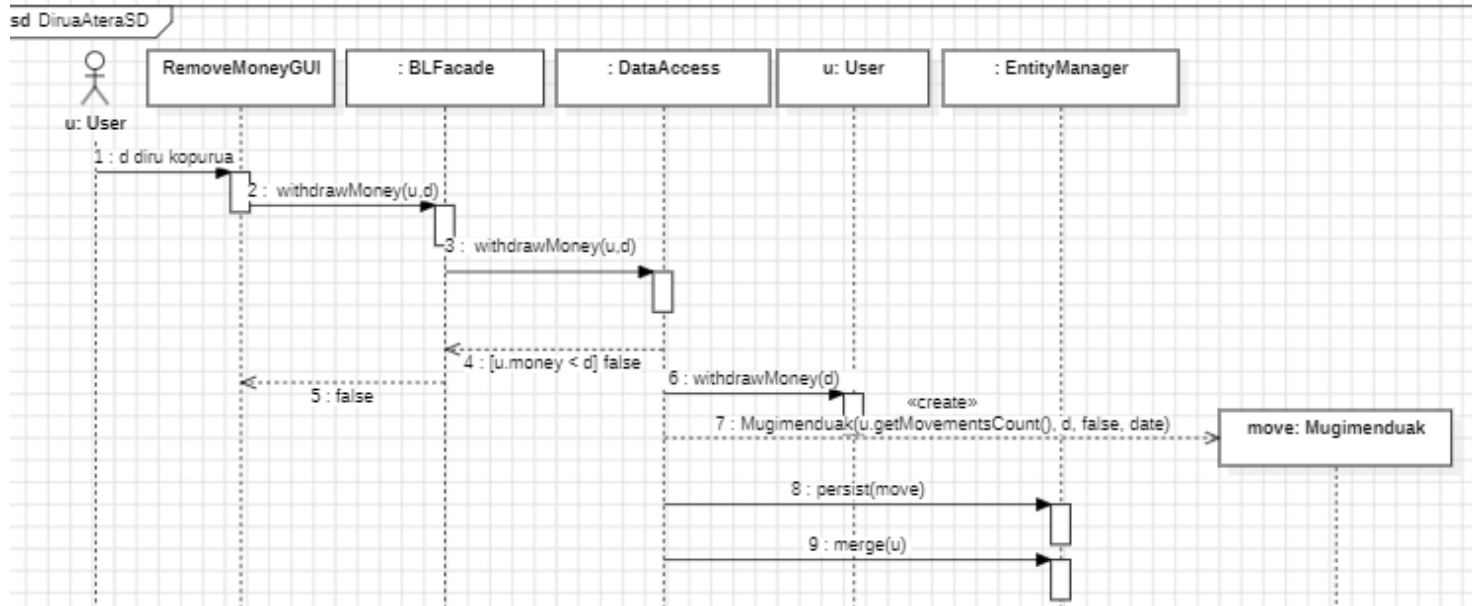
1. *Uzerrak* sartu nahi duen **dirua** sartzen du.
2. *Uzerrak* **dirua sartu** botoia sakatzen du.
3. *Sistemak* diruaren mugimendua gordetzen du.
4. *Sistemak* **user**-ari dirua gehitzen dio.
  
```



## DIRUA ATERA

```

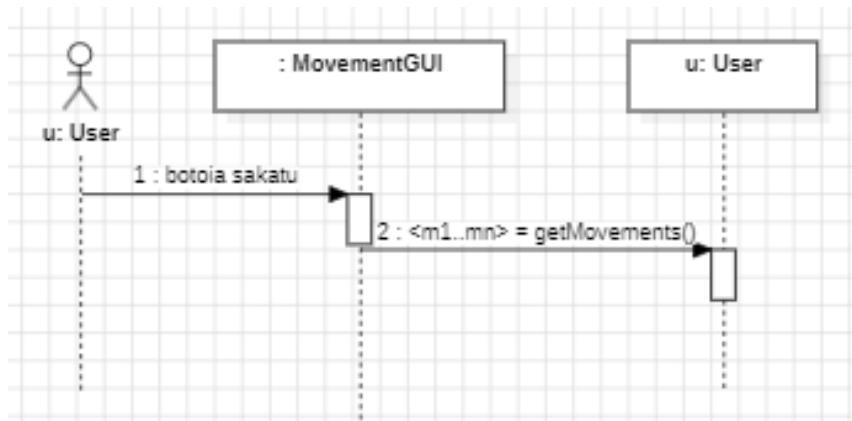
# Dirua ateratzeko Gertaera Fluxuak
## Oinarrizko Gertaerak
1. *User-ak* atera nahi duen **dirua** sartzen du.
2. *User-ak* **dirua atera** botoia sakatzen du.
3. *Sistemak* diruaren mugimendua gordetzen du.
4. *Sistemak* **user** horrek ea dirua baduen konprobatzen du, eta dirua kentzen dio.
## Gertaera Alternatiboak
1. User-ak ez du diru hori. Errore mezu bat agertzen da.
  
```



## MUGIMENDUAK IKUSI

```

# Mugimenduen kontultaren Gertarea Fluxuak
## Oinarrizko Gertaerak
1. *Sistemak* **mugimendu** guztiak eta beraien **mota** pantailarazten du.
  
```

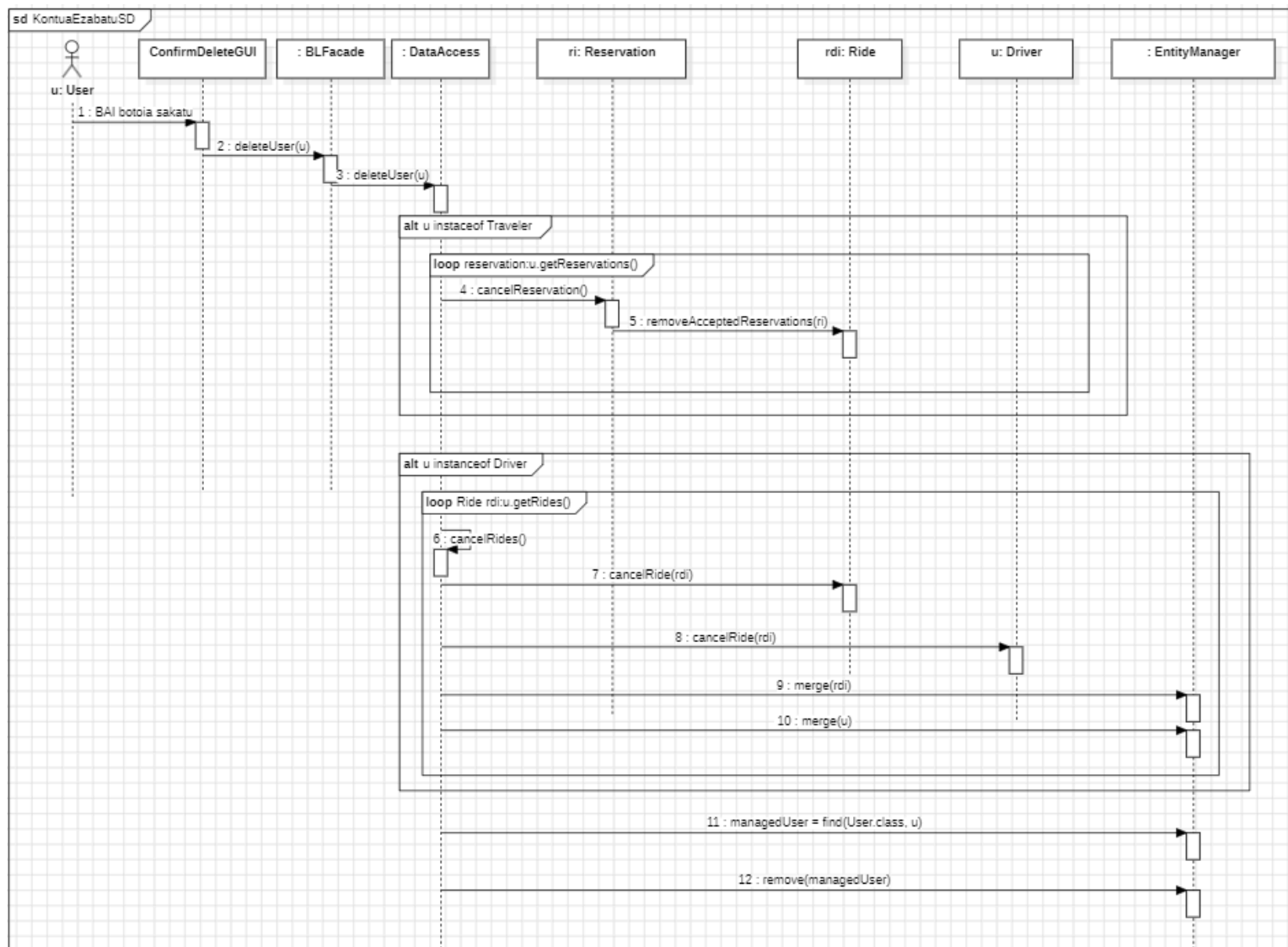


## KONTUA EZABATU

```

# Kontua ezabatu Gertarea Fluxuak
## Oinarrizko Gertaerak
1. *Userak* **bai** botoia sakatzen du.
2. *Sistemak* ez du ezer egiten.

## Gertaera Alternatiboak
1. User-a **bidaiaria** bada, erreserba guztiak ezabatuko ditu eta kontua datu basetik ezabatuko du.
2. User-a **agidaria** bada, bidai guztiak kantzulatuko ditu eta kontua datu basetik ezabatuko du.
3. User-a **admin-a** bada, ez da ezer egingo.
  
```

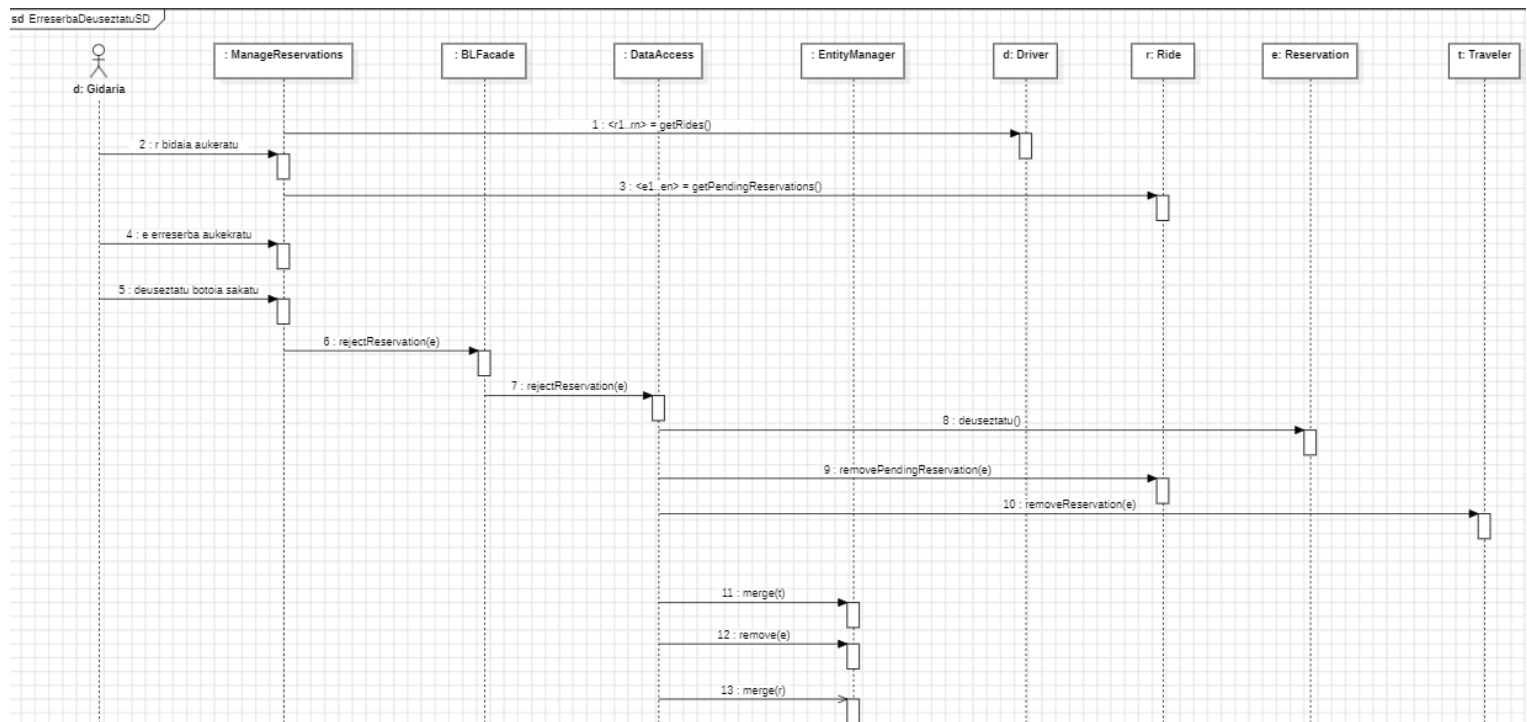


## ERRESERBA DEUSEZTATU

# Erreserba Deuseztatu Gertaera Fluxuak

## Oinarrizko Gertaerak

1. *\*Gidariak\** deuseztatu nahi duen *\*\*bidaiaria\*\** aukeratzen du.
2. *\*Gidariak\** *\*\*deuseztatu\*\** botoia sakatzen du.
3. *\*Sistemak\** *\*\*bidaiari\*\** hori listatik ezabatzen du.

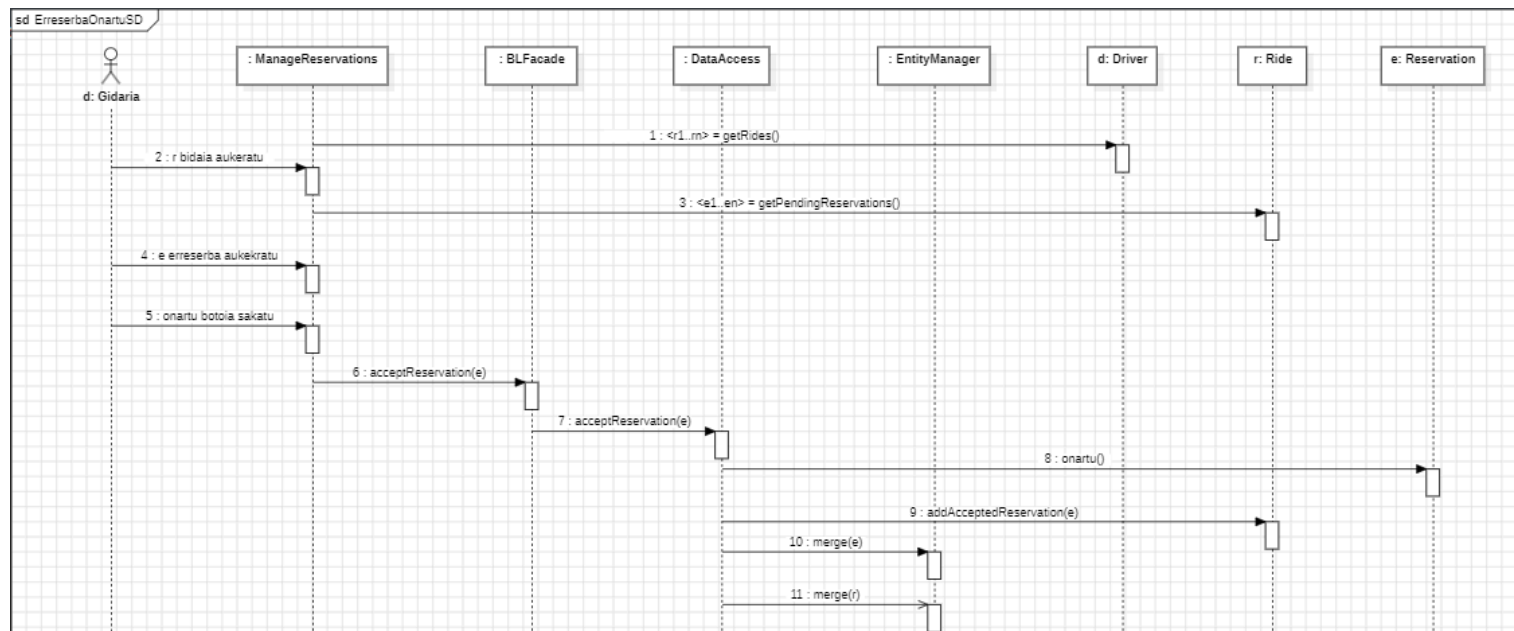


## ERRESERBA ONARTU

# Erreserba Onartu Gertaera Fluxuak

## Oinarrizko Gertaerak

1. **\*Gidariak\*** onartu nahi duen **\*\*bidaiaria\*\*** aukeratzen du.
2. **\*Gidariak\*** **\*\*onartu\*\*** botoia sakatzen du.
3. **\*Sistemak\*** **\*\*bidaiari\*\*** hori bidaian sartzen du.

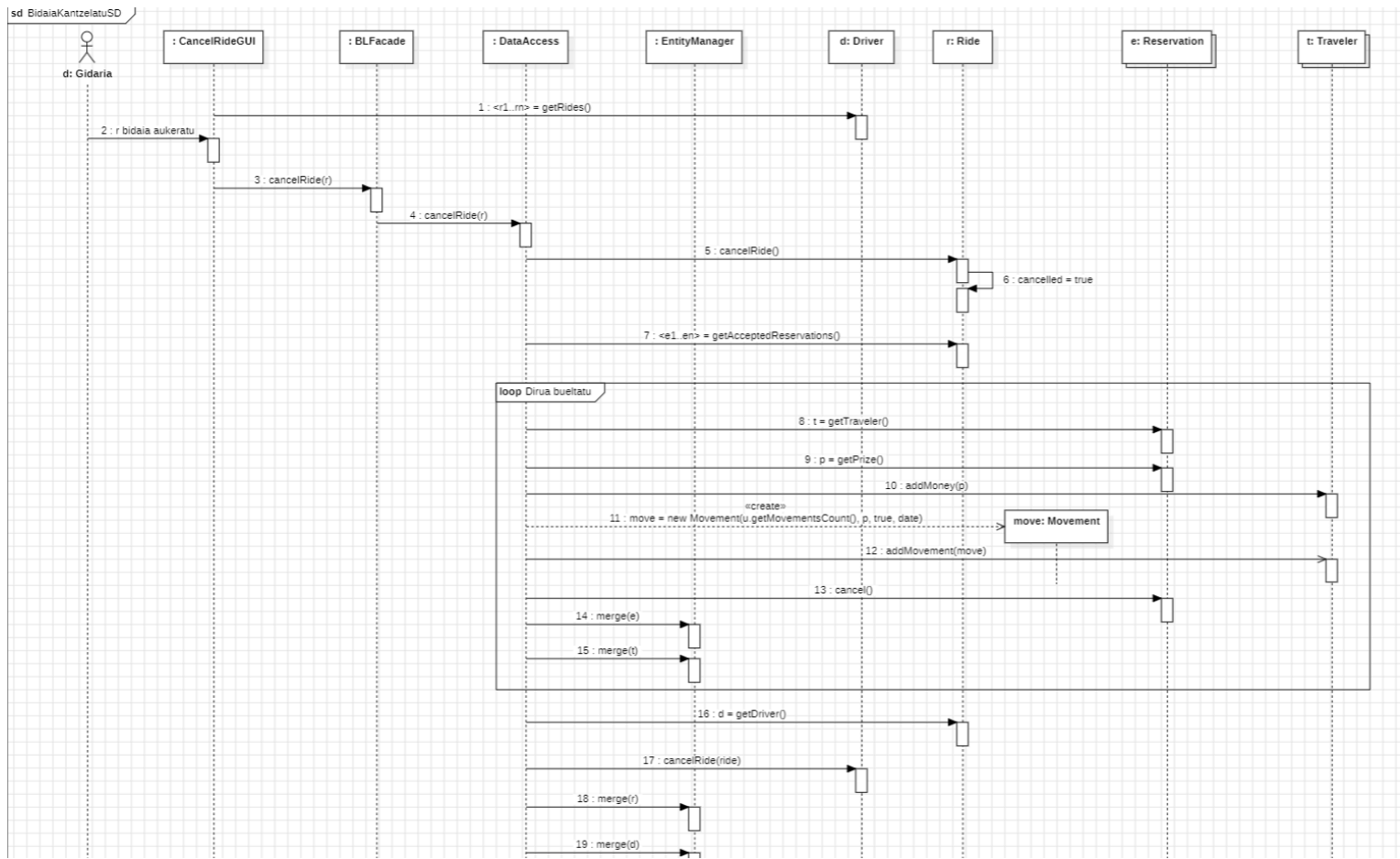


## BIDAIA KANTZELATU

```

# Bidaia kanzelatze Gertarea Fluxuak
## Oinarrizko Gertaerak

1. *Sistemak* bukatuta ez dauden **bidai** guztiak erakusten ditu.
1. *Gidariak* **bidai** bat aukeratzen du.
3. *Sistemak* **bidai** horren **kantzelatua** atributoa **true**-ra aldatzen du.
4. *Sistemak* **bidai** horren **erreserben** egoerak kanzelatzen ditu.
5. *Sistemak* **dirua** **bidaiari** bueltatzen dio.
6. *Sistemak* **bidai** hori **gidariaren** bidai listatik kentzen du.
  
```

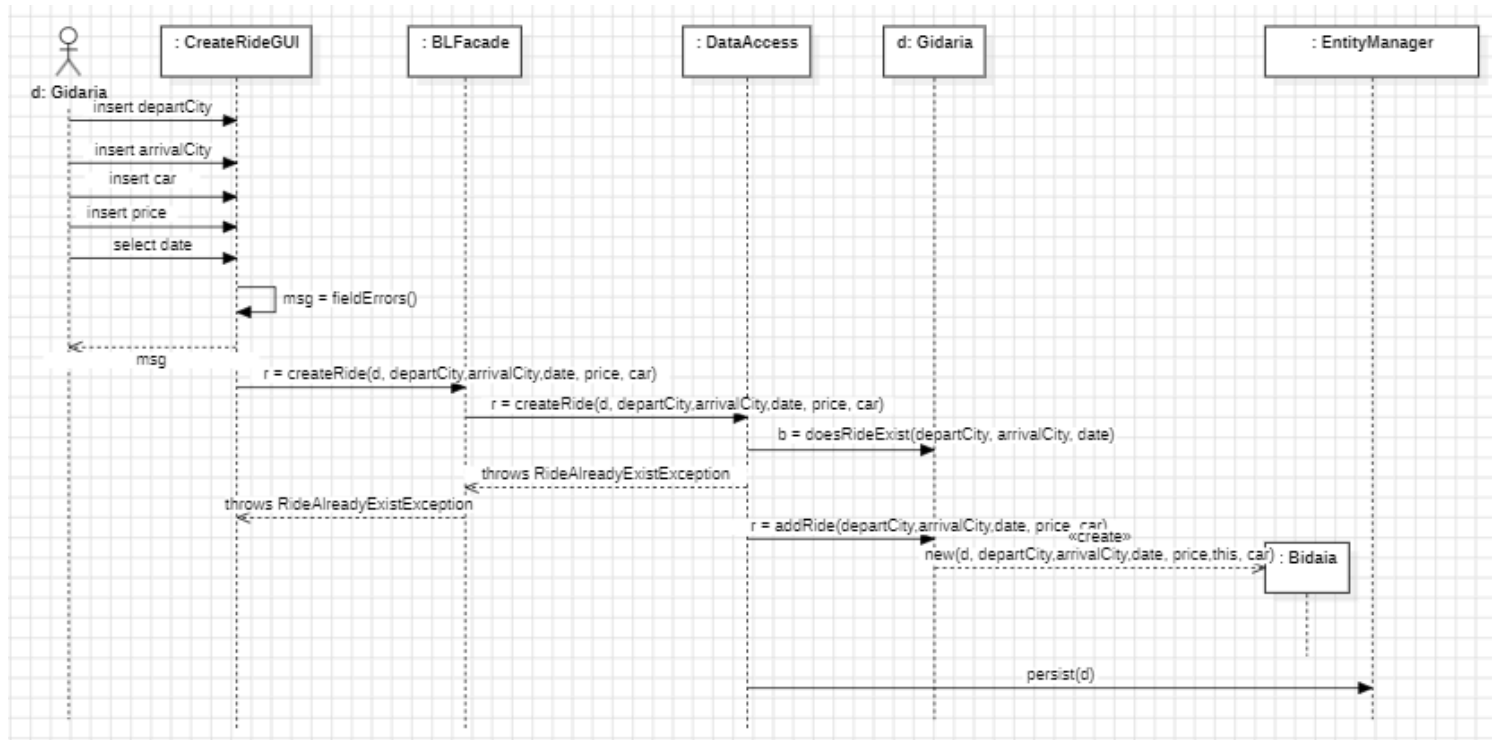


## BIDAIA SORTU

```

# Ibilaldiaren Gertaera Fluxuak
## Oinarritzko Gertaerak
1. *Gidariak* **irteera hiria**, **helmuga hiria**, **eserleku kopurua** eta **prezioa** sartzen ditu.
2. *Gidariak* **ibilaldiaren data** hautatzen du.
3. *Sistemak* **ibilaldia** sortzen du sartutako datuekin eta **gidariari** esleitzen dio.

## Gertaera Alternatiboak
1. Eremu batzuk hutsik daude. Amsiera.
2. **Eserleku kopurua** edo **prezioa** ez dira zenbakiak. Sistemak erabiltzaileari jakinarazten dio.
3. **Ibilaldiaren data** gaur baino lehenagokoa da. Sistemak erabiltzaileari jakinarazten dio.
4. **Gidari horrek** dagoeneko **ibilaldi hori** du sortua. Ibilaldia ez da sortzen. Sistemak erabiltzaileari jakinarazten dio.
  
```



## KOTXEA SARTU

```

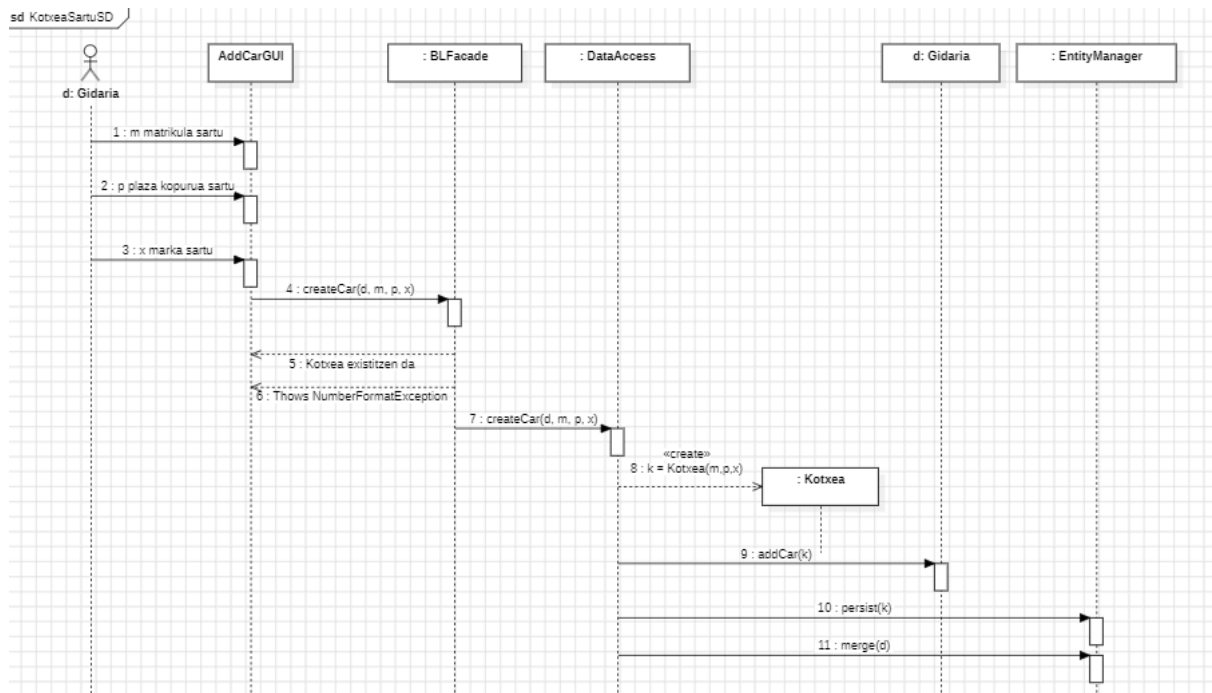
# Kotxea sartzeko Gertarea Fluxuak
## Oinarrizko Gertaerak

1. *Gidariak* kotxearen **matrikula**, **marka** eta **eserlekuak** sartzen ditu.
2. *Gidariak* **kotxea sartu** botoia sakatzen du.
3. *Sistemak* **kotxea** datu basean gordetzen du.

## Gertaera Alternatiboak

1. Kotxea dagoeneko sartuta dago. Ez da kotxe berria sortzen.
  
```





## ERREKLAMAZIOAK KUDEATU

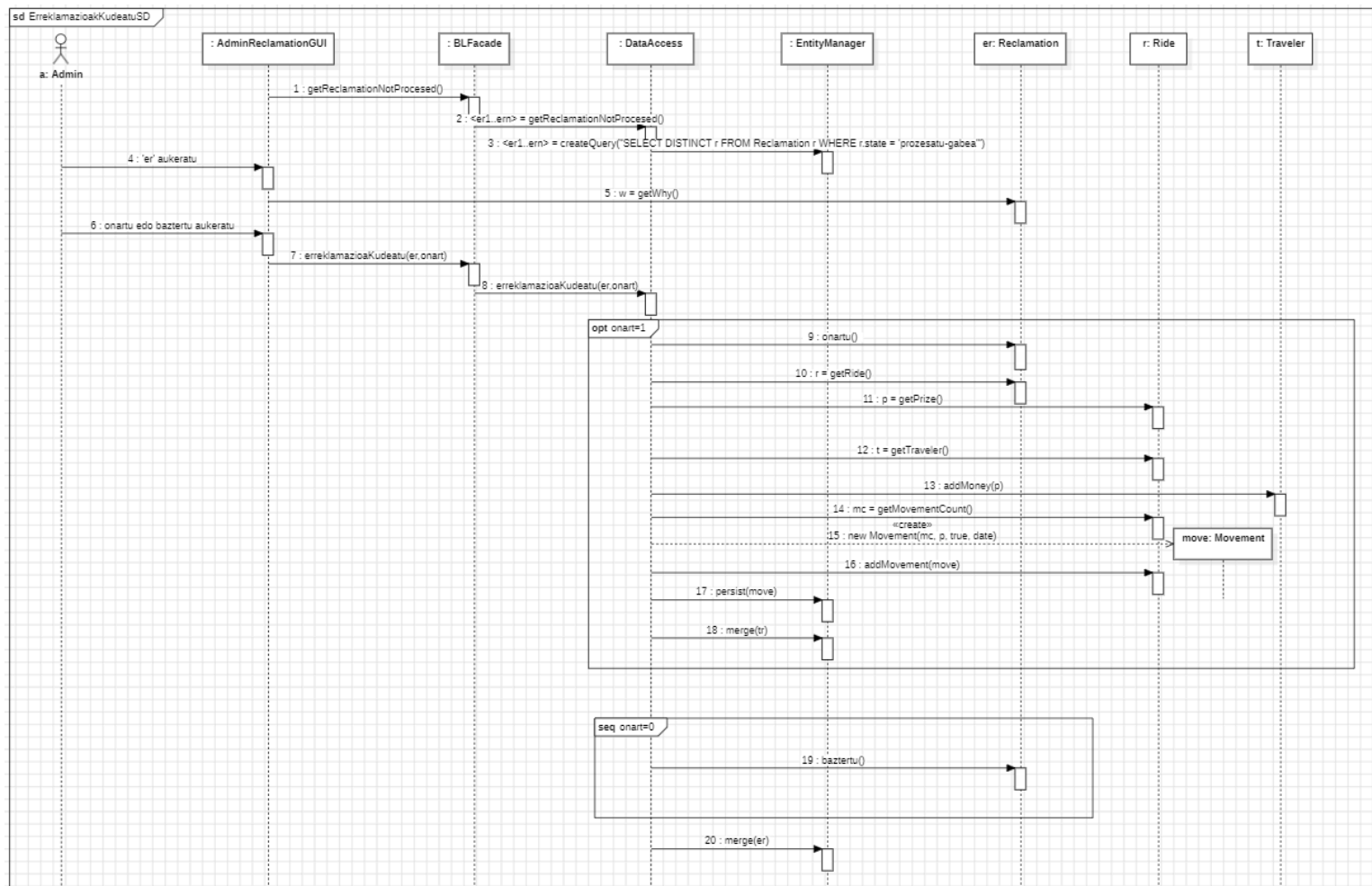
# Erreklamazioak kudeatu Gertarea Fluxuak

## Oinarrizko Gertaerak

1. \*Sistemak\* prozesatu gabe dauden \*\*erreklamazio\*\* guztiak erakusten ditu.
2. \*Administratzaileak\* \*\*erreklamazio\*\* bat aukeratzen du.
3. \*Sistemak\* erreklamazio horren zergatia erakusten du.
4. \*Administratzaileak\* erreklamazio hori baztertzen du.
5. \*Sistemak\* \*\*erreklamazio\*\* horren egoera aldatzen du "Baztertua" egoerara.

## Gertaera Alternatiboak

1. \*Administratzaileak\* \*\*erreklamazio\*\* hori onartzen du. Dirua \*\*bidaiariari\*\* ematen zaio eta \*\*erreklamazioaren\*\* egoera "Onartua" egoerara aldatzen da.



- **Implementazioa**

**public void initializeDB():** Programa lehen aldiz exekutatzean datu basea sortzeko erabiliko da, gidariak, bidaiariak, bidaiak... sortuarazteko.

**public List<String> getDepartCities():** Bidaia guztien irteera hiriak buelatzen ditu metodo honek.

**public** List<String> getDestinationCities(String from) : Irteera hiri bat emanda, hemendik ateratzen diren bidaia guztien helmuga hiriak bueltatzen ditu metodo honek.

**public** Ride createRide(String from, String to, Date date, **float** price, String driverEmail, Car pCar) **throws** RideMustBeLaterThanTodayException, RideAlreadyExistException: Nondik, norakoa, data, prezio, gidari email eta kotxe bat emanda, bidaia bat sortzen du.

**public** Car createCar(Driver driver, String matrikula, **int** nPlazak, String marka) : Kotxe bat sortzen du, gidari, matrikula, plaza kopurua eta kotxe marka bat emanda.

**public** List<Ride> getRides(String from, String to, Date date) : Nondik norako bat emanda eta data bat pasata, dauden bidaiak jasotzen dituen metodo bat da.

**public** List<Date> getThisMonthDatesWithRides(String from, String to, Date date) : Metodo honek datu-basetik berreskuratzen ditu gertaerak dauden hilabeteko datak

**public** Assessment createAssessment(Ride ride, Traveler traveler, **int** stars, String text) : balorazio bat sortzen da, bidaia, bidaiari, izar kopuru eta komentario bat pasatuz.

**public boolean** confirmReservation(Reservation erreserba) : erreserba bat konfirmatzeko erabiltzen den metodoa da.

**public boolean** acceptReservation(Reservation erreserba) : bidaiari batek erreserba bat egitean, gidariak erreserba onartzeko metodoa da.

**public boolean** rejectReservation(Reservation erreserba) : bidaiari batek erreserba bat egitean, gidariak erreserba onartzeko metodoa da.

**public boolean** createReservation(Traveler traveler, Ride ride, **int** [nPlaces](#)) : bidaiari batek bidaia bat erreserbatzeko erabiliko da metodo hau, bidaiaria, bidaia eta eserleku kantitatea pasaz.

**public boolean** cancelRide(Ride ride) : bidaia bat pasaz, bidaia kantzelatuko du.

**public boolean** updateRide(Ride ride) : bidaia bat pasaz, bidaia eguneratuko du.

**public boolean** updateReservation(Reservation reservation) : erreserba bat emanda, erreserba eguneratuko du.

**public boolean** removeReservation(Reservation reservation) : erreserba bat emanda, erreserba ezabatuko du.

**public boolean** updateTraveler(Traveler traveler) : traveler bat jasota, travelerra eguneratuko du.

**public boolean** addMoney(User user, float money) : erabiltzaile bat jasota eta diru kantitate bat emanda, erabiltzaileari diru kantitate hau gehituko zaio kontu korrontean.

**public boolean** withdrawMoney(User user, float money) : aurrekoaren antzera, erabiltzaile bat jasota eta diru kantitate bat emanda, erabiltzaileari diru kantitate hau kenduko zaio kontu korrontetik.

**public** Driver getDriverByName(String driver) : driver bat lortuko da driver baten izena jasotzean.

**public** Ride getRideByParams(Driver driver, int nPlaces, float price, String from, String to, Date date) : bidaia bat lortzen da gidari, eserleku kantitate, prezio, nondik norakoa eta data jasota.

**public** Ride getRideByID(int id) : bidaia bat lortzen da bidaia id bat jasotzean.

**public** User checkLogin(String user, String password) : erabiltzaile eta pasahitz bat sartzean, datu basean zihurtatzen du erabiltzaile eta pasahitzak existitzen diren eta ondo dauden.

**public** User register(String izena, String email, String pasahitza, String mota) : erabiltzaile bat erregistratzen du izena, emaila, pasahitza eta erabiltzaile mota jasoz.

**public** List<Reservation> travelersReservations (Traveler traveler) : lista bat itzultzen du travelerrak dituen bidai erreserba guztiekin.

**public** List<Ride> ridesByDriver (Driver driver) : lista bat itzultzen du driverrak dituen bidaia guztiekin.

**public boolean** deleteUser (User user) : erabiltzaile bat jasota, erabiltzailea ezabatzen du.

**public** List<Reclamation> getReclamation(Traveler [trav](#)) : lista bat itzultzen du travelerrak dituen erreklamazio guztiekin.

**public boolean** reclamation(Traveler traveler, Reservation res, String why) : erreklamazio bat sortzen du bidaiari, erreserba eta arrazoi bat jasota.

**public** List<Reclamation> getReclamationNotProcesed() : oraindik prozesatuak izan ez diren erreklamazioak itzuliko dituen lista bat itzultzen du.

**public boolean** erreklamazioaKudeatu(Reclamation r, **int** [a](#)) : administratzaileak erreklamazioak kudeatzeko, erreklamazio eta int bat emanda (baieztatua edo ezeztatua esan nahi duena) kudeatuko du.

**public** List<Reservation> getErreserbaBefore(Traveler traveler) : erreserba lista bat itzuliko du travelerrak egindako erreserbena, gaurko data baino lehenago izandakoak.

**public** Reclamation getReclamationByID(**int** id) : erreklamazio bat itzuliko du id bat jasotzean.

**public** Alert createAlert(Traveler u, String nondik, String nora, Date [d](#)) : traveler, nondik norakoa eta data bat pasata, alerta bat sortuko du.

**public** List<Ride> checkAlerts(Traveler t) : traveler bat jasotzean, alertarik duen txekeatuko du eta bidai lista bat itzuliko du.

- **Ondorioak**

Proiektu hau egin ondoren, atera ditugun ondorioak hauek izan dira:

- Taldeko dinamika ona izan da, denok aportatu dugu.
- Ikasgaian entrega asko egotea gutxien gustatu zaiguna izan da.
- Iterazioekin lan egitea interesgarria izan da, pixkanaka proiektu bera osatzen joatea.
- Hurrengo urteko ikasleei gomendatuko genieke denborarekin pentsatzea nola egingo duten programaren atal ezberdinak eta ahalik eta modu txukunean egitea hasieratik, beranduago denbora ez galtzeko.

- **Bideoaren URL-a**

<https://youtu.be/-09P7hPd4II>

- **Kodearen zip fitxategia**

(Entregaren atazan)