

SOFTWARE INGENIARITZA  
2025KO MAIATZA

# BIDAIAK KONPARTITZEKO APLIKAZIOA

---

HUIYING JIN, QIAN LIZHANG,  
ANGELA NORIEGA ETA AITZOL WALIÑO

2.1 TALDEA

eman ta zabal zazu



UPV EHU

# AURKIBIDEA

1. Egileak	3
2. Iterazioak	3
3. Web Zerbitzuak eta Internalizazioa	3
4. Sarrera	3
5. Eskakizun Bilketa	4
5.1. Domeinuaren eredua	4
5.2. Erabilpen kasuen eredua	5
6. Diseinua	29
6.1. Sekuentzia diagramak	29
6.2. Klase diagrama	37
7. Inplementazioa	38
8. Ondorioak	42
8.1. Pentsatutako hobekuntzak	42
8.2. Lan taldearen dinamika	42
8.3. Proiektuaren konplexutasuna	42
8.4. Lortutako emaitzen poztasuna	42
8.5. Gomendioak eta iradokizunak hurrengo kurtsoko ikasleentzat	42
8.6. Proiektua garatzeko erabili den metodologia	42
8.7. Proiektuetan oinarritutako irakaskuntzari eta irakasgaiari buruzko iritzia edo kritika	42
9. Bideoaren URL-a	43
10. Kodea	43

# 1. Egileak

Huiying Jin, Qian Lizhang, Angela Noriega eta Aitzol Waliño.

## 2. Iterazioak

	Scrum Masterra	Erabilitako orduak (h)
1. Iterazioa	Huiying Jin	42:30
2. Iterazioa	Aitzol Waliño	25
3. Iterazioa	Qian Lizhang	55:30

1. taula: iterazio bakoitzeko Scrum Masterra eta erabilitako ordu kopurua

## 3. Web Zerbitzuak eta Internalizazioa

- Web zerbitzuak garatu? BAI
- Internalizazioa garatu? BAI

## 4. Sarrera

Proiektu honen helburua bidaiak konpartitzeko aplikazio bat garatzea da, non edozeinek bidaiak kontsultatu dezaken, baina bakarrik erregistratuek beste funtzionalitateak eskura izango dituzte. Aplikazioa hiru hizkuntza desberdinetan erabili daiteke: euskara, gaztelera eta ingelesa. Erregistratzerako orduan, zein rola hartu nahi den hautatzen da: gidaria edo bidaiaria.

Gidariak bidaiak sortuko dituzte, baina aurretik bere kotxea sartuta izan behar dute. Bidaiariak erreserbak egin ahal dituzte, eta egindako erreserben egoera ikusi. Erreserbak aplikazioaren saldoaren bitartez gauzatuko dira, non bidaiariak saldoa sartu eta gidariak saldoa atera dezaketen.

Baliteke bidaiariak nahi duen bidai konkretua ez egotea, beraz, bidaiariak alerta bat sortu dezake nahi duen jatorri, helmuga eta datarekin. Gidari batek bidai hori sortzen badu, alertaren egoera aldatuko da, eta bidaiariak bere alertak kontsultatu ditzazke.

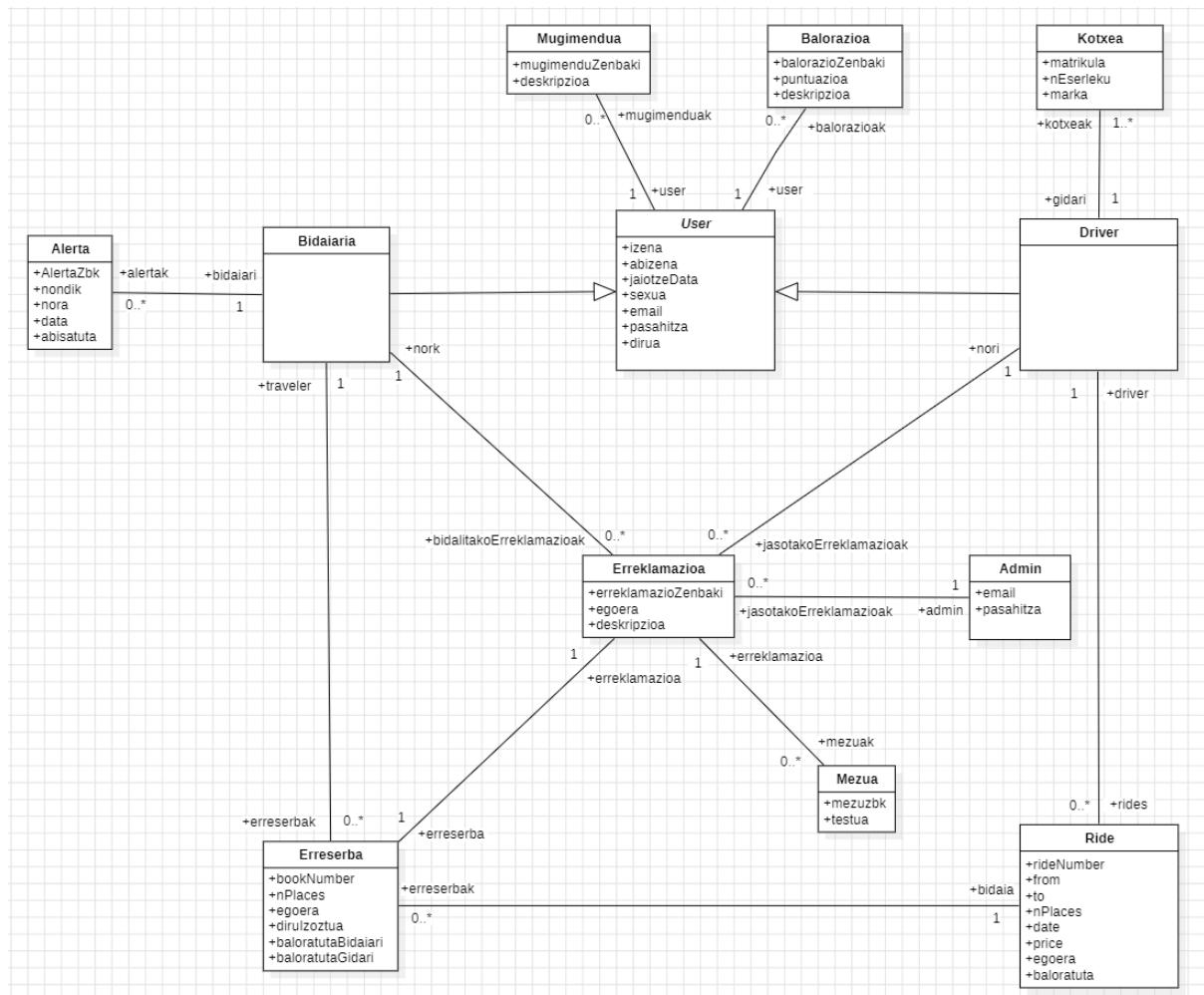
Gidariak bidaia kantzeltatu dezakete eta erreserbak onartu edo deuseztatu ditzakete. Behin bidaia amaituta bidaiariak bidaia baieztatu dezakete. Bidaiari guztiek bidaia baieztatu ondoren, bidaiak ahalik eta seguruena izateko asmoz, bai gidariak bai bidaiariak bidaia baloratu ahal izango dute. Gainera, bidaiariak erreklamazioak sortu ditzazke, honek gidariaren eta bidaiariaren arteko mezu trukea ahalbidetuko luke. Gidariak erreklamazioa

onartuz gero, bidaiariari dirua itzuliko zaio, eta deuseztatuz gero, erreklamazioa administratzaileari pasako lioke.

Erabiltzaile bakoitzak izandako diru mugimenduak kontsultatu ditzake. Bestalde, erabiltzaileek beraien kontua ezabatu dezakete.

## 5. Eskakizun Bilketa

### 5.1. Domeinuaren eredua



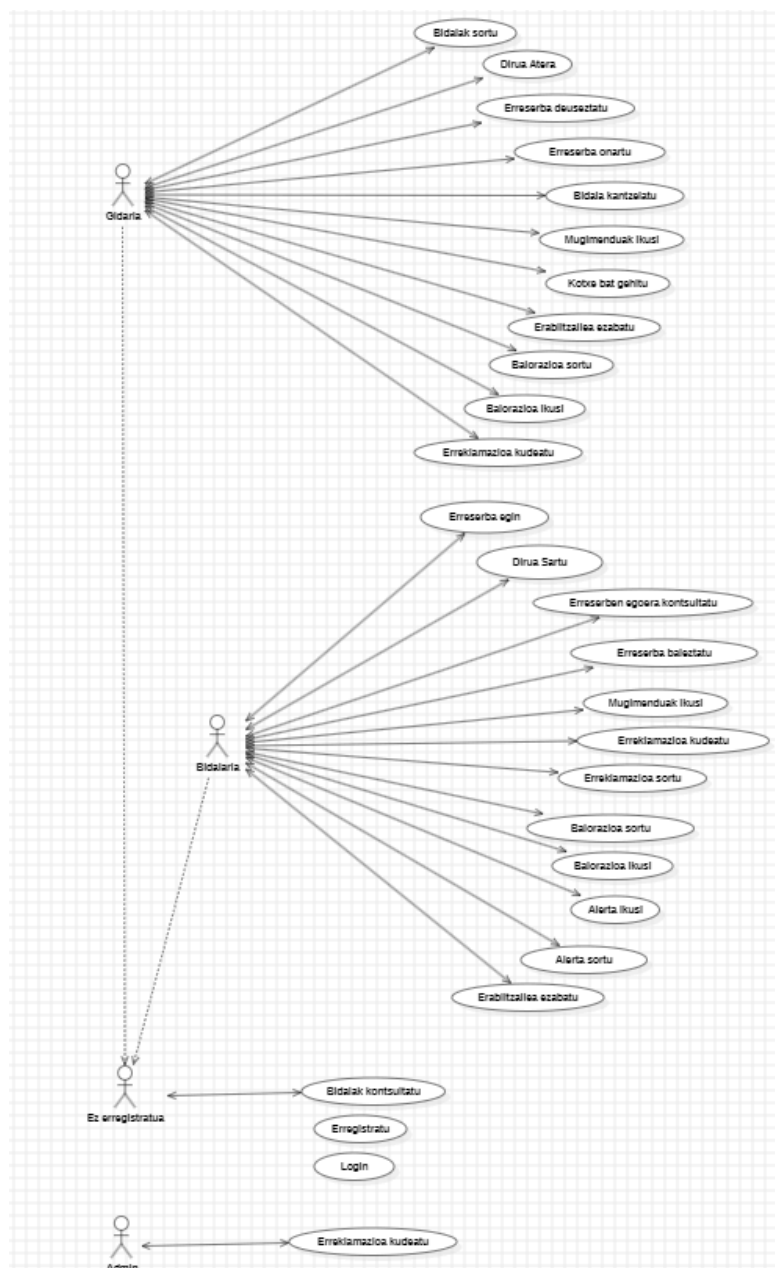
1. irudia: aplikazioaren domeinuzko eredua

Sortutako entitateak eta haien arteko erlazioak:

1. User: Bidaiaria eta gidariaren klase gurasoa, User batek hainbat mugimendu eta balorazio izan ditzake (0 edo gehiago).
2. Bidaiaria: Bidaiari batek hainbat alerta, erreserba eta erreklamazio izan ditzake (0 edo gehiago). User klasearen klase umea da.
3. Driver: Gidari batek gutxienez kotxe bat izan behar du eta hainbat bidai eta erreklamazio izan ditzake (0 edo gehiago). User klasearen klase umea da.

4. Admin: Administrazioaile batek hainbat erreklamazio izan ditzake (0 edo gehiago).
5. Kotxea: bakoitzak gidari bakar bat izan behar du.
6. Ride: Bidai bakoitzak gidari bakar bat izan behar du. Bidai batek hainbat erreseba izan daiteke.
7. Erreserba: Erreserba bakoitzak bidai, bidaiari, erreklamazio eta gidari bakar bat ditu.
8. Erreklamazio: bakoitzak bidaiari, gidari, erreserba eta admin bat du. Hainbat mezu izan ditzake (0 edo gehiago).
9. Mezua: bakoitzak erreklamazio bakar bat izan behar du.
10. Alerta: bakoitzak bidaiari bakar bat izan behar du.
11. Mugimendua: bakoitzak User bakar bat du.
12. Balorazioa: bakoitzak User bakar bat du.

## 5.2. Erabilpen kasuen eredua



2. irudia: aplikazioaren erabilpen kasuen eredua

### 1. Bidaia sortu (gidari)

### 3. irudia: bidaia sorturen interfaze grafikoa

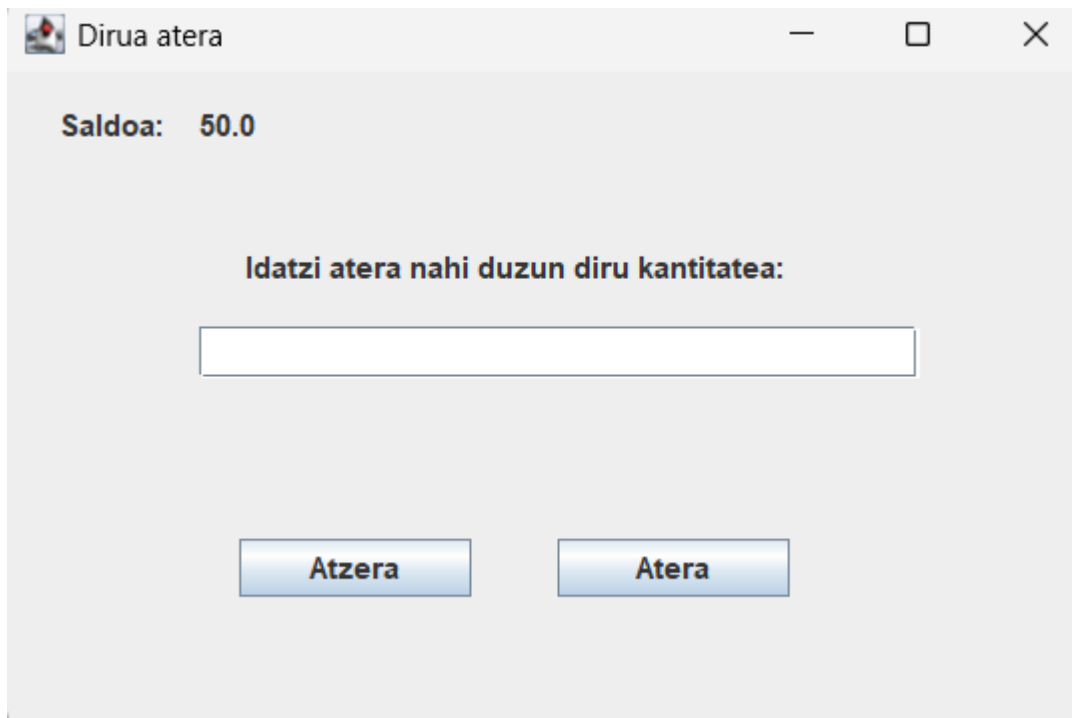
#### # OINARRIZKO FLUXUA

1. Gidariak bidaia jatorri hiria, iritsiera hiria eta prezioa zehazten ditu.
2. Gidariak kotxe bat aukeratzen du.
3. Gidariak bidaia data ezartzen du.
4. System-ek bidaia bat sortzen du sartutako jatorri hiria, iritsiera hiria, prezioa, kotxea eta bidaia datarekin gidaria-erentzat.
5. System-ek alerta guztien lista eskuratzen du eta bidaia eta alerta baten data berdina baldin bada alerta horren "abisatuta"-ri true ezartzen dio.

#### # FLUXU ALTERNATIBOA

1. Eremu batzuk hutsik daude. Bukatu.
2. Prezioa ez da zenbaki bat. Sistemak erabiltzaileari jakinarazten dio.
3. Bidaia data gaurko eguna baino lehenagoko data bati dagokio. Sistemak erabiltzaileari jakinarazten dio.
4. Bidaia hau sortuta dago gidari honentzat. Bidaia ez da sortzen. Sistemak erabiltzaileari jakinarazten dio.
5. Ez dago kotxerik erregistratuta beraz, ezin da bidaia sortu. Sistemak erabiltzaileari jakinarazten dio.
6. Itxi botoiari ematen dio eta System-ek gidaria-ren hasiera orrira itzultzen du.

## 2. Dirua atera (gidari)



Dirua atera

Saldoa: 50.0

Idatzi atera nahi duzun diru kantitatea:

Atzera Atera

4. irudia: dirua ateraren interfaze grafikoa

### # OINARRIZKO FLUXUA

1. Gidariak atera nahi duen diru kopurua ezarri.
2. System-ek dirua eguneratu eta mugimendu berri bat sortzen du.

### # FLUXU ALTERNATIBOA

1. Atera nahi duen diru kopurua saldoan daukana baino handiagoa izatea, zenbakia ez izatea edo balio negatiboa izatea. System-ek erabiltzailea jakinarazi.
2. Atzera botoiari ematen dio eta System-ek gidariaren hasiera orrira itzultzen du.

3. Erreserba deuseztatu eta erreserba onartu (gidari)

Nondik	Nora	Data	Prezioa	Bidaiaia	Eserleku ...	Erreserb...
Bilbo	Donostia	Wed May...	10.0	Magdalen...	2	32

Onartu Deuseztatu Atzera

5. irudia: erreserba deuseztatu eta onarturen interfaze grafikoa

(Deuseztatu)

# OINARRIZKO FLUXUA

1. Gidariak, bidaiariak eskatutako erreserba deuseztatzen du.
2. System-ek bidaiariari bidaiaren dirua itzultzen dio eta mugimendu berri bat sortzen du.
3. System-ek erreserbaren egoera eguneratzen du, eserleku kopuruak eguneratuz.

# FLUXU ALTERNATIBOA

1. Atzera botoiari ematen dio eta System-ek gidariaren hasiera orrira itzultzen du.
2. Ez dago erreserbarik eginda. Amaitu.

(Onartu)

# OINARRIZKO FLUXUA

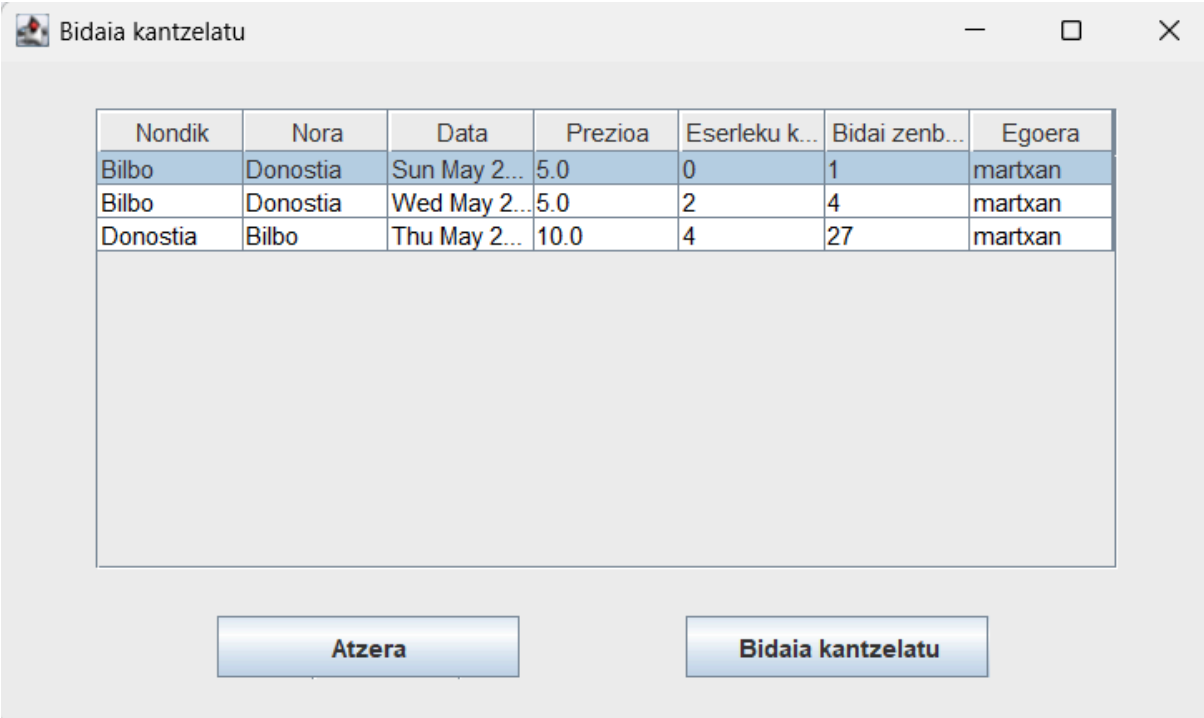
1. Gidariak, bidaiariak eskatutako erreserba onartzen du.
2. Systemek bidaiaren egoera eguneratzen du.

#FLUXU ALTERNATIBOA

1. Atzera botoiari ematen dio eta Systemek gidariaren hasiera orrira itzultzen du.
2. Ez dago erreserbarik eginda. Amaitu.



#### 4. Bidaia kantzelatu (gidari)



Nondik	Nora	Data	Prezioa	Eserleku k...	Bidai zenb...	Egoera
Bilbo	Donostia	Sun May 2...	5.0	0	1	martxan
Bilbo	Donostia	Wed May 2...	5.0	2	4	martxan
Donostia	Bilbo	Thu May 2...	10.0	4	27	martxan

Atzera      Bidaia kantzelatu

6. irudia: bidaia kantzelaturen interfaze grafikoa

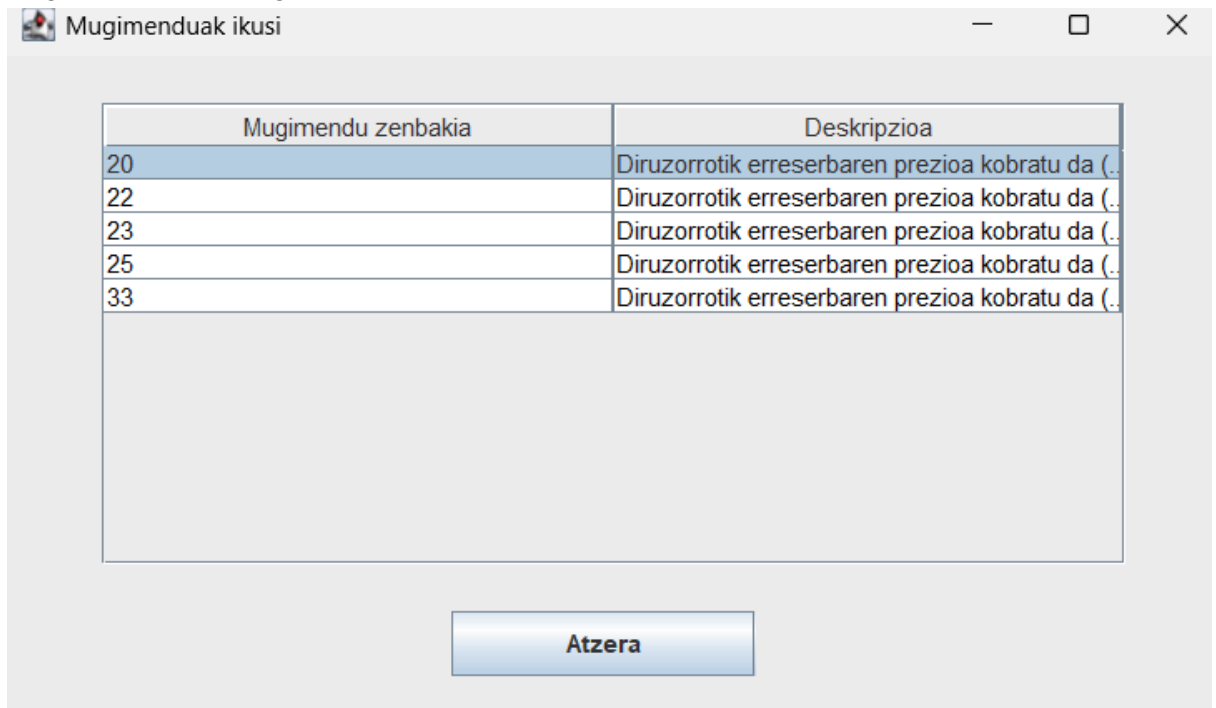
#### # OINARRIZKO FLUXUA:

1. System-ek erreserben lista pantailan erakutsi.
2. Gidariak bidaia bat aukeratu eta kantzelatu.
3. System-ek bidaiari guztei dirua bueltatzen die eta mugimendu berri bat sortzen du eta erreserben eta bidaiaren egoera eguneratzen du.

#### # FLUXU ALTERNATIBOA:

1. Ez dago bidaiarik sortuta. Amaitu.
2. Gidariak atzera botoiari eman. System-ek hasiera orrira eraman.

5. Mugimenduak ikusi (gidari eta bidaiari)



7. irudia: mugimenduak ikusiaren interfaze grafikoa

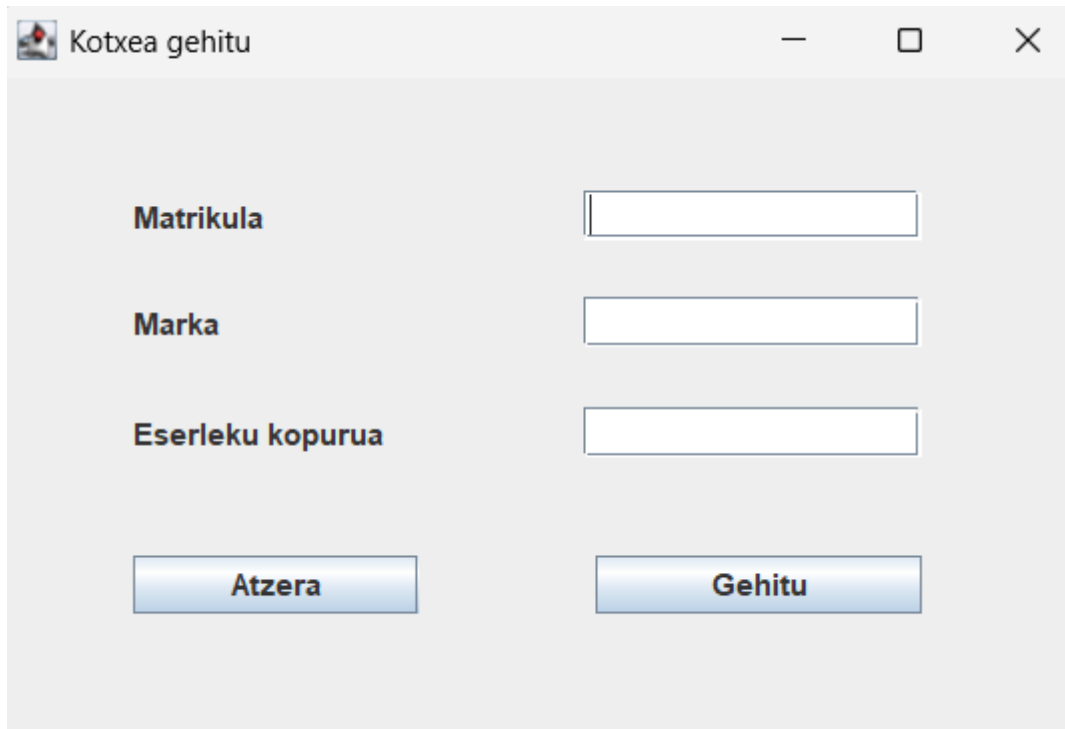
# OINARRIZKO FLUXUA:

1. System-ek egindako mugimenduak pantailaratu.

# FLUXU ALTERNATIBOA:

1. Ez da mugimendurik egin. Amaitu.
2. Bidaiariak atzera botoiari ematen dio. System-ek hasiera orrira eraman.

6. Kotxe bat gehitu (gidari)



Kotxea gehitu

**Matrikula**

**Marka**

**Eserleku kopurua**

**Atzera** **Gehitu**

8. irudia: kotxe bat gehituren interfaze grafikoa

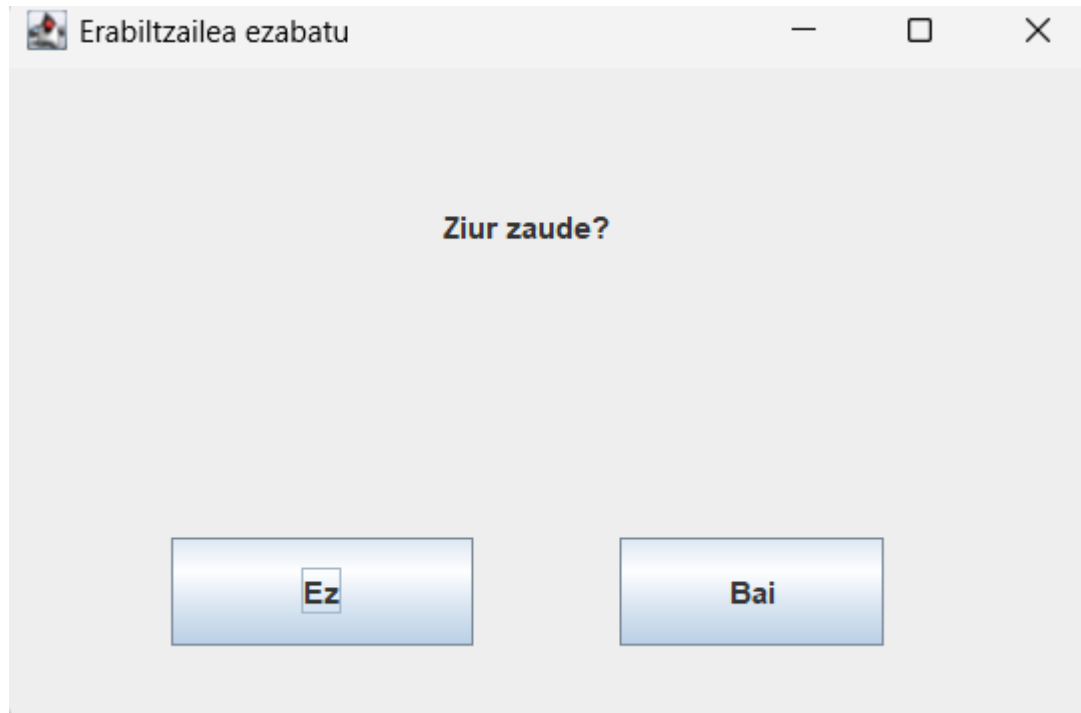
# OINARRIZKO FLUXUA:

1. Gidariak kotxearen marka, matrikula eta eserleku kopurua sisteman gehitu.
2. System-ek kotxea sortzen du eta datu-basean gordetzen du.

# FLUXU ALTERNATIBOA:

1. Matrikula berdineko beste kotxe bat existitzen da sisteman. Gidariari jakinarazi.
2. Eserleku kopurua negatiboa edo testu bat da. Gidariari jakinarazi.

7. Erabiltzailea ezabatu (gidari eta bidaiari)



9. irudia: erabiltzailea ezabaturen interfaze grafikoa

(Gidari)

# OINARRIZKO FLUXUA:

1. Gidariak, erabiltzailea ezabatu botoia sakatzen du.
2. System-ek gidari hori datu-basetik ezabatuko du.
3. System-ek, aplikazioan, gidari horrek zituen eragiketa guztiak ezabatuko ditu (egindako balorazioak utziz).
4. Lehio guztiak ixten dira.

# FLUXU ALTERNATIBOA:

1. Gidariak ez botoia sakatzen du. Hasiera orrira itzuli.

(Bidaiari)

# OINARRIZKO FLUXUA:

1. Bidaiariak, erabiltzailea ezabatu botoia sakatzen du.
2. System-ek bidaiari hori datu-basetik ezabatuko du.
3. System-ek, aplikazioan, bidaiari horrek zituen eragiketa guztiak ezabatuko ditu (egindako balorazioak utziz).
4. Lehio guztiak ixten dira.

# FLUXU ALTERNATIBOA:

1. Bidaiariak ez botoia sakatzen du. Hasiera orrira itzuli.

## 8. Balorazioa sortu (gidari)



**Balorazioa sortu**

**Bidaia** 1 ▼ **Bidaia** bidaia@gmail.com ▼

**Puntuazioa** 1 ▼

**Deskripzioa**

Atzera Bidali

10. irudia: balorazioa sorturen interfaze grafikoa

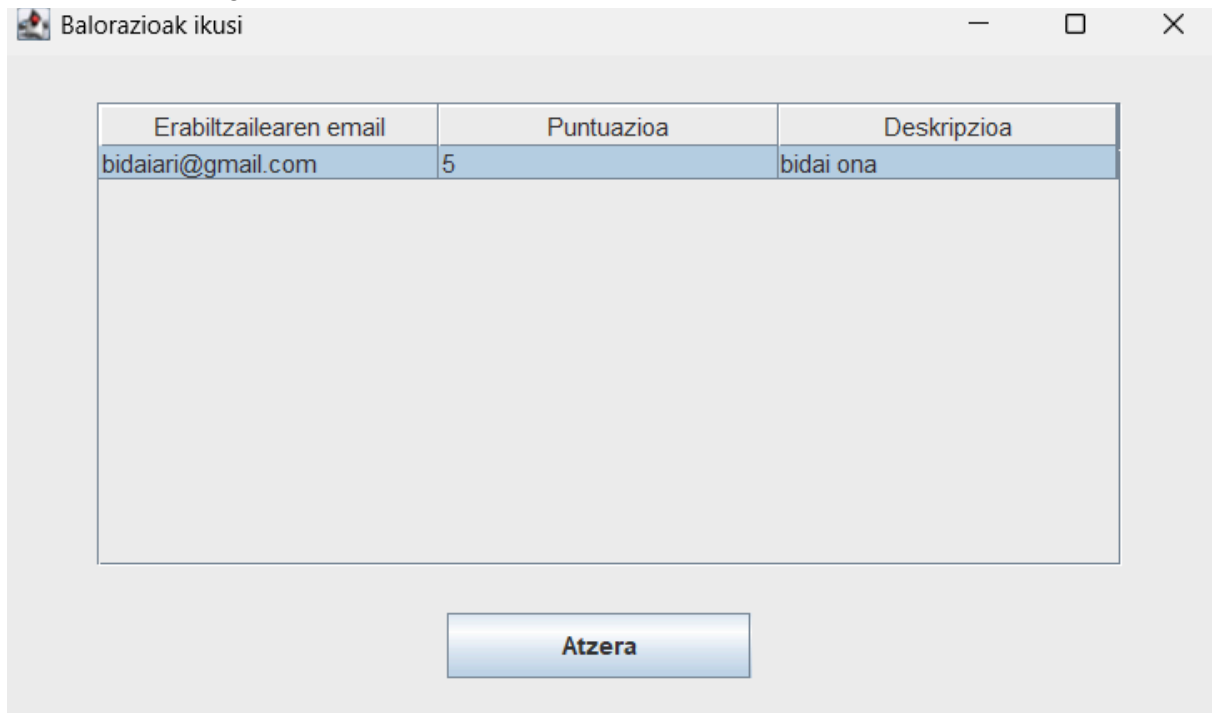
### # OINARRIZKO FLUXUA:

1. Gidariak bidaia bat aukeratzen du.
2. System-ek gidariak aukeratutako bidaiaren bidaiarien lista pantailaratzen du.
3. Gidariak aukeratutako bidaiako bidaiari bat aukeratzen du
4. Gidariak 1 eta 5 arteko puntuazio bat ipini egindako bidaiaren eraman dion bidaiariari.
5. Gidariak mezua bat idazten du bidaiariaren inguruan.
6. Gidariak bidali botoiari ematen dio.
7. System-ek balorazioa sortzen du eta datu-basean gordetzen du.

### # FLUXU ALTERNATIBOA:

1. Atzera botoiari sakatu. System-ek hasiera orrira eraman.
2. Ez du bidaiarik egin gidariak. Amaitu.

9. Balorazioa ikusi (gidari eta bidaiari)



11. irudia: gidariaren balorazioa ikusiren interfaze grafikoa

(Gidari)

# OINARRIZKO FLUXUA:

1. System-ek bidaiariak gidariari egindako balorazioen zerrenda pantailaratu.

# FLUXU ALTERNATIBOA:

1. Ez egotea oraindik baloraziorik gidariarentzat. Amaitu.
2. Atzera botoiari eman eta System-ek hasiera orrira eraman.

(Bidaiari)

# OINARRIZKO FLUXUA:

1. System-ek gidariak bidaiariari egindako balorazioen zerrenda pantailaratu.

# FLUXU ALTERNATIBOA:

1. Ez egotea oraindik baloraziorik bidaiariarentzat. Amaitu.
2. Atzera botoiari eman eta System-ek hasiera orrira eraman.

## 10. Erreklamazioa kudeatu (gidari)

Erreklamazio z...	Email	Deskripzioa	Egoera
30	driver@gmail.c...	esperientzia tx...	itxaron

Mezua
Bidaia: ez zait gustatu

**Gehitu mezu bat**

Atzera Onartu Deuseztatu Bidali

12. irudia: gidariaren erreklamazioa kudeaturen interfaze grafikoa

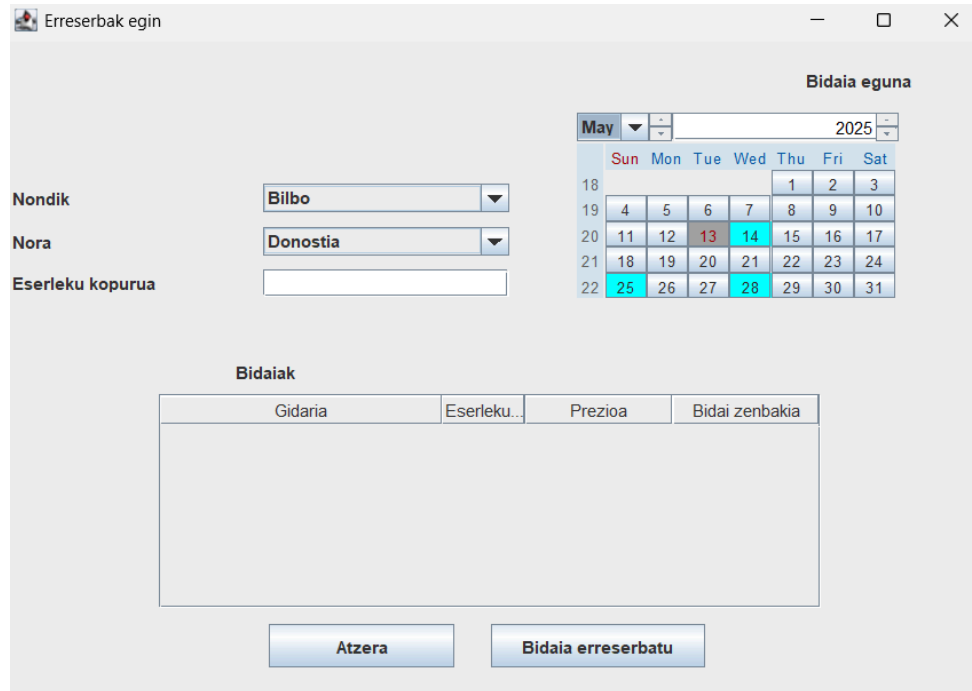
### # OINARRIZKO FLUXUA:

1. System-ek erreklamazioen zerrenda pantailaratu.
2. Gidariak erreklamazio bat aukeratzen du.
3. Gidariak mezu bat idatzi eta bidali botoiari eman, System-ek mezu berria sortzen du eta datubasean gorde.
4. Gidariak onartu edo deuseztatu botoiari ematen dio. System-ek erreklamazioaren egoera eguneratzen du. Egoera onartu bada, bidaia irudiak dirua itzultzen dio, bestela, System-ek erreklamazioa administratzailearen listara gehitzen du.

### # FLUXU ALTERNATIBOA:

1. Gidariak atzera botoiari ematen dio eta System-ek hasiera orrira itzultzen dio.
2. Ez dago erreklamaziorik. Amaitu.
3. Mezua hutsik dago. Erabiltzaileari jakinarazi.
4. Erreklamazioa deuseztatu edo onartu egoeran dago. Erabiltzaileari jakinarazi.

## 11. Erreserba egin (bidaiari)



13. irudia: erreserba eginaren interfaze grafikoa

### # OINARRIZKO FLUXUA:

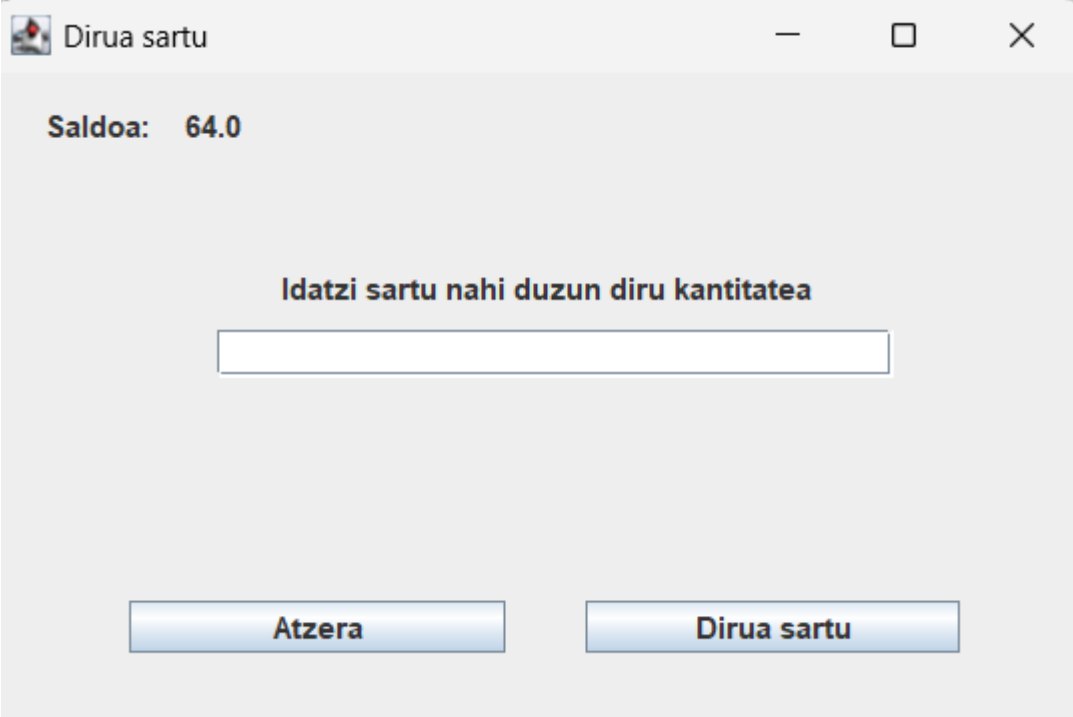
1. System-ek ibilaldiak abiatzen diren hiri guztiak bistaratzen ditu.
2. Bidaiariak irteerako hiria hautatzen du.
3. System-ek hautatutako hiritik abiatzen diren bidaien helmugako hiri guztiak bistaratzen ditu.
4. Bidaiariak helmuga hiria hautatzen du.
5. System-ek egutegi batean irteera hiri-tik helmugarako hirietara aukeratutako hilabete batean ibilaldiak dauden egunak nabarmentzen ditu.
6. Bidaiariak nabarmendutako egunen artean bat aukeratuko du.
7. System-ek aukeratutako egunean dauden bidaia guztiak bistaratzen ditu.
8. Bidaiariak bidaia bat aukeratzen du.
9. Bidaiariak eserleku kopurua zehazten du.
10. System-ek datu basean gordetzen du eskaria eta gidariari gehitzen dio eskaera.
11. System-ek bidaiariari ezarritako dirua kentzen dio eta mugimendu berri bat sortzen du eta erreserbari egoera ezartzen dio.

### # FLUXU ALTERNATIBOA:

1. Bidaiaria-k Eserleku kopurua aukeratutako bidaiari ezarritako eserleku kopuru maximoa baino handiagoa eskatu du, ez da zenbakia edo balioa negatiboa da. System-ek erabiltzaileari jakinarazi.
2. Bidaiaria-k ezarritako prezioa baino diru gutxiago dauka. System-ek erabiltzaileari jakinarazi.
3. Atzera botoiari ematen dio eta System-ek Bidaiaria-ren hasiera orrira itzultzen du.



## 12. Dirua sartu (bidaiari)



Dirua sartu

Saldoa: 64.0

Idatzi sartu nahi duzun diru kantitatea

Atzera Dirua sartu

14. irudia: dirua sarturen interfaze grafikoa


### # OINARRIZKO FLUXUA:

1. Bidaiaria-k kontuan sartu nahi duen diru kantitatea ezarri.
2. System-ek kontuko dirua eguneratu eta mugimendu berri bat sortzen du.

### # FLUXU ALTERNATIBOA:

1. Sartutako diru kantitatea negatiboa izatea edo zenbakia ez izatea. System-ek erabiltzailea jakinarazi.
2. Atzera botoiari ematen dio eta System-ek bidaiaria-ren hasiera orrira itzultzen du.

### 13. Erreserben egoera kontsultatu (bidaia)



The screenshot shows a window titled "Erreserba egoera" with a table containing reservation data. The table has 8 columns: Nondik, Nora, Data, Prezioa, Bidaiaia, Eserleku ..., Erreserb..., and Egoera. There are three rows of data. Below the table is a large empty rectangular area, and at the bottom center is a button labeled "Atzera".

Nondik	Nora	Data	Prezioa	Bidaiaia	Eserleku ...	Erreserb...	Egoera
Bilbo	Donostia	Sun May ...	4.0	Magdalen...	4	19	itxaron
Bilbo	Donostia	Sun May ...	5.0	Magdalen...	2	21	itxaron
Bilbo	Donostia	Sun May ...	5.0	Magdalen...	2	24	onartu

Atzera

16. irudia: erreserben egoera kontsultaturen interfaze grafikoa

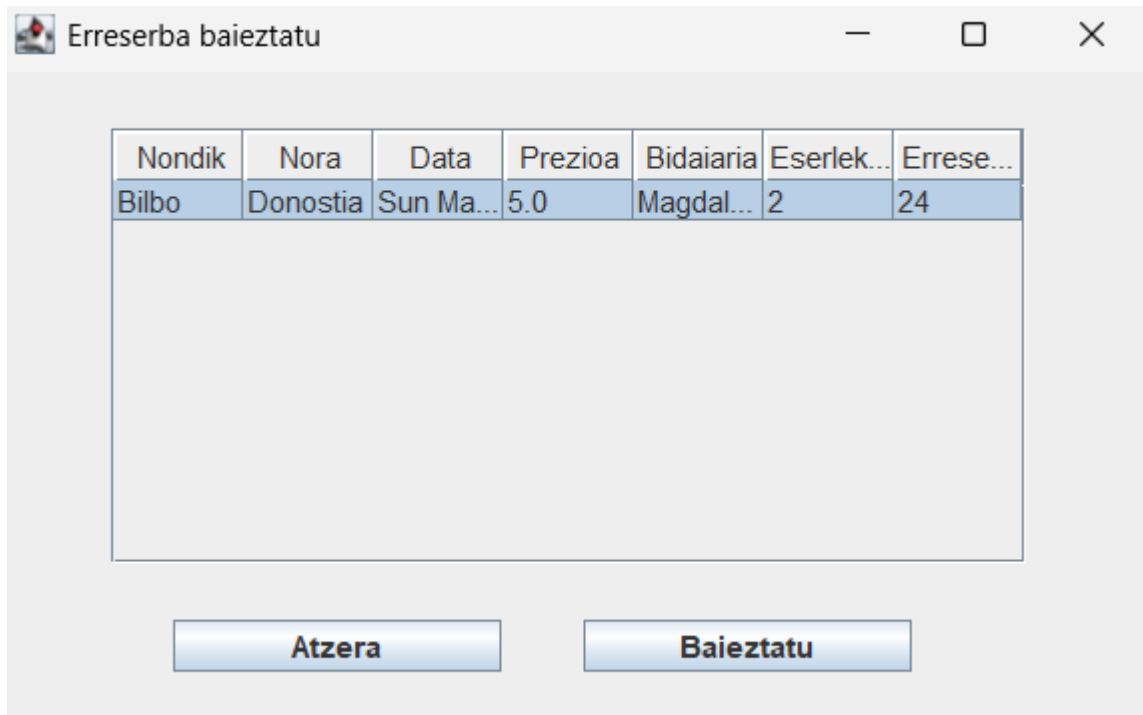
#### # OINARRIZKO FLUXUA:

1. System-ek, bidaiaiak egindako erreserba lista pantailaratu.

#### # FLUXU ALTERNATIBOA:

1. Ez dago bidaiairik erreserbatuta. Amaitu.
2. Bidaiaiak atzera botoiari ematen dio. System-ek hasiera orrira eraman.

#### 14. Erreserba baieztatu (bidaiari)



Nondik	Nora	Data	Prezioa	Bidaiaria	Eserlek...	Errese...
Bilbo	Donostia	Sun Ma...	5.0	Magdal...	2	24

AtzeraBaieztatu

17. irudia: erreserba baieztaturen interfaze grafikoa

#### # OINARRIZKO FLUXUA:

1. System-ek bidaiariaren erreserba onartuen zerrenda pantailaratu.
2. Bidaiariak bidaia bat aukeratzen du.
3. Bidaiariak baieztatu botoia sakatu eta bidaiaren egoera eguneratzen da.
4. System-ek gidariari bidaiaren prezioaren balioa gehitzen dio, mugimendu berri bat sortzen du eta erreserbaren egoera eguneratzen da.

#### # FLUXU ALTERNATIBOA:

1. Bidaiarik ez egotea. Amaitu.
2. Bidaiariak atzera botoiari eman. System-ek hasiera orrira eraman.

### 15. Erreklamazioa kudeatu (bidaiari)

Erreklamazio z...	Email	Deskripzioa	Egoera
30	driver@gmail.c...	esperientzia tx...	itxaron

Mezua
Bidaiaria: ez zait gustatu

**Gehitu mezu bat**

**Atzera** **Bidali**

18. irudia: bidaiariaren erreklamazio kudeaturen interfaze grafikoa

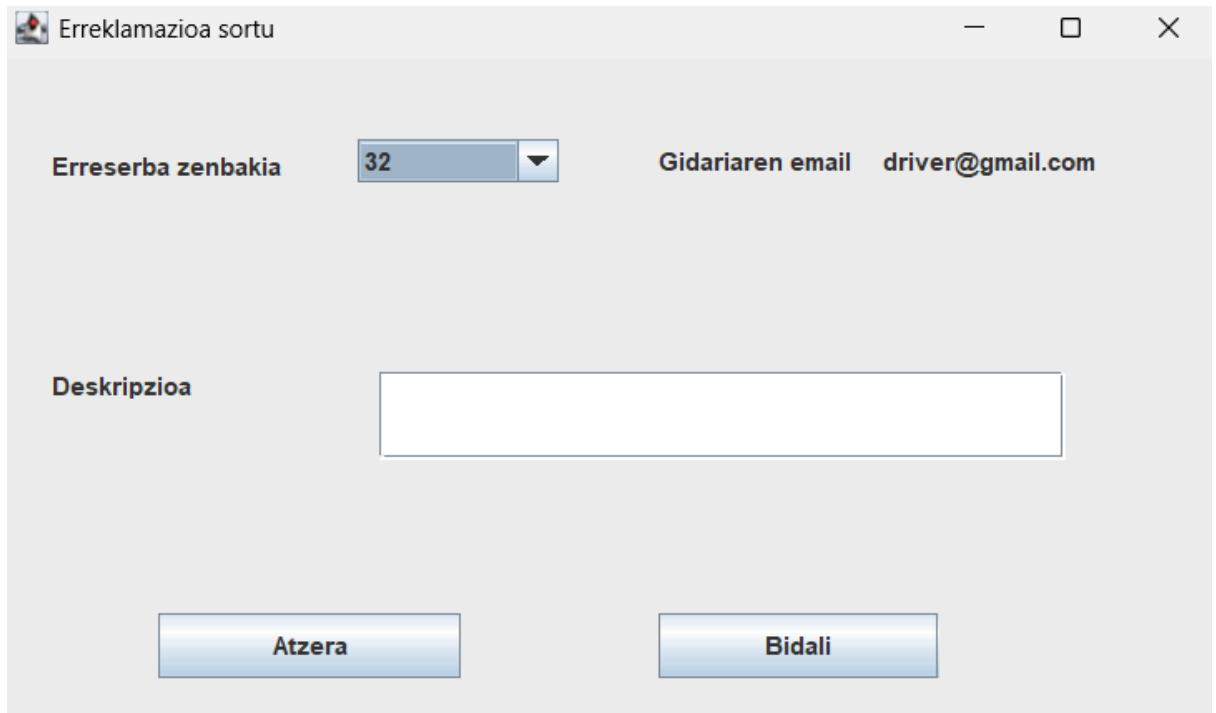
#### # OINARRIZKO FLUXUA:

1. System-ek bidaiariak egindako erreklamazioen lista pantailaratu.
2. Bidaiariak erreklamazio bat aukeratu.
3. Bidaiariak mezu bat gehitu aukeratutako erreklamazioari. System-ek mezu berri bat sortu eta datu-basean gordetzen du.
4. Bidaiariak bidali botoiari eman eta System-ek erreklamazioa eguneratu.

#### # FLUXU ALTERNATIBOA:

1. Bidaiariak oraindik ez eginda edukitzea erreklamaziorik. Amaitu.
2. Mezu bat bidaltzerakoan deskripzioak ezer ez izatea. Bidaiariari jakinarazi.
3. Atzera botoiari eman eta System-ek hasiera orrira eraman.
4. Erreklamazioa deuseztatu egoeran dago. Erabiltzaileari jakinarazi.

#### 16. Erreklamazioa sortu (bidaia)



Erreklamazioa sortu

Erreserba zenbakia 32

Gidariaren email driver@gmail.com

Deskripzioa

Atzera Bidali

19. irudia: bidaia-irtearen erreklamazioa sortuaren interfaze grafikoa

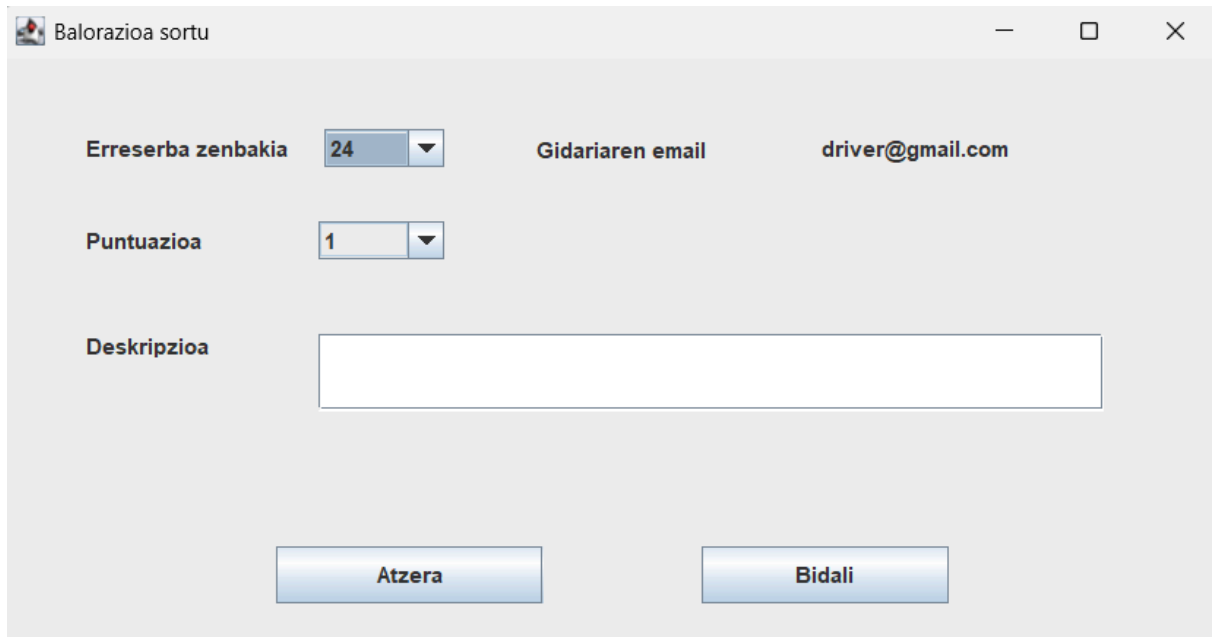
#### # OINARRIZKO FLUXUA:

1. System-ek, bidaia-irtearen amaitutako bidaien zerrenda pantailaratu.
2. Bidaia-irteak amaitutako bidaien listatik bidaia bat aukeratu.
3. Bidaia-irteak, deskripzio bat idazten du bere azalpenekin.
4. Bidaia-irteak bidali botoia sakatzen du.
5. System-ek erreklamazioa sortu, datu-basean gorde eta gidariari erreklamazioa bidali.

#### # FLUXU ALTERNATIBOA:

1. Ez dago amaitutako bidaia-irte. Amaitu.
2. Bidaia-irteak atzera botoia eman. System-ek hasiera orrira eraman.
3. Deskripzioa hutsik. Bidaia-irteari jakinarazi.

## 17. Balorazioa sortu (Bidaiari)



Balorazioa sortu

Erreserba zenbakia 24

Gidariaren email driver@gmail.com

Puntuazioa 1

Deskripzioa

Atzera Bidali

20. irudia: bidaiariaren balorazioa sorturen interfaze grafikoa

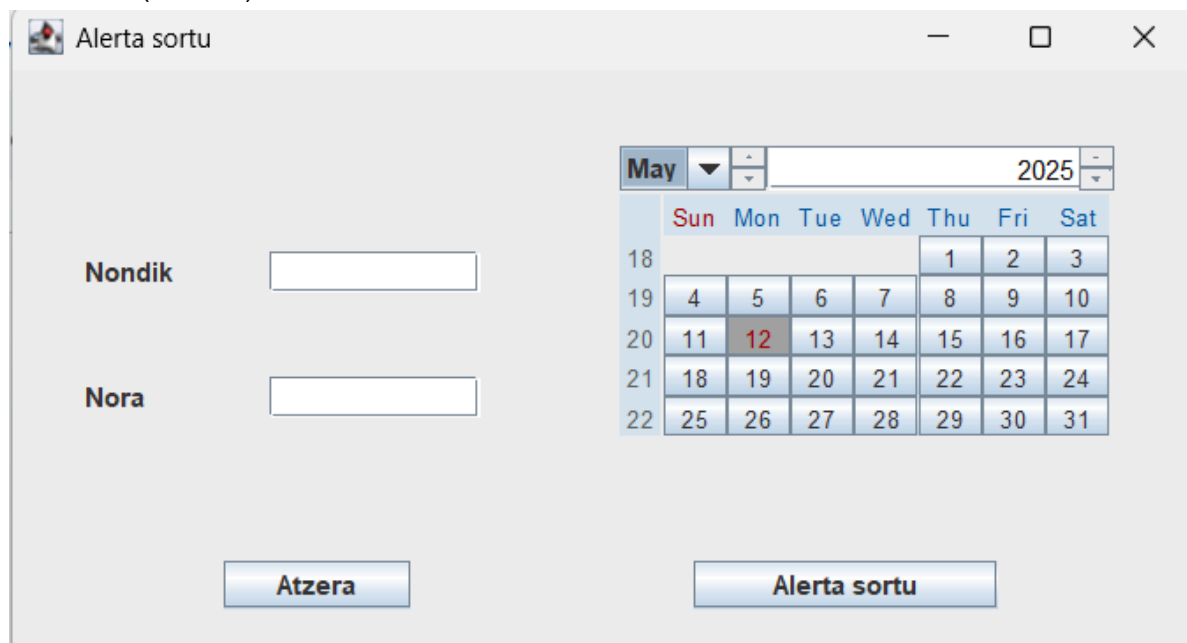
### # OINARRIZKO FLUXUA:

1. Bidaiariak bidaia bat aukeratzen du.
2. Bidaiariak 1 eta 5 arteko puntutazio bat ipini aukeratu duen bidaiako gidariari.
3. Bidaiariak mezu bat idazten du gidariaren inguruan.
4. Bidaiariak bidali botoiari ematen dio.
5. System-ek balorazioa sortu eta datubasean gordetzen du.

### # FLUXU ALTERNATIBOA:

1. Atzera botoiari sakatu. System-ek hasiera orrira eraman.
2. Ez du bidaiarik egin gidariak. Amaitu.

## 18. Alerta sortu (bidaiari)



**Alerta sortu**

Nondik

Nora

May 2025

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18					1	2	3
19	4	5	6	7	8	9	10
20	11	12	13	14	15	16	17
21	18	19	20	21	22	23	24
22	25	26	27	28	29	30	31

Atzera

Alerta sortu

21. irudia: alerta sorturen interfaze grafikoa

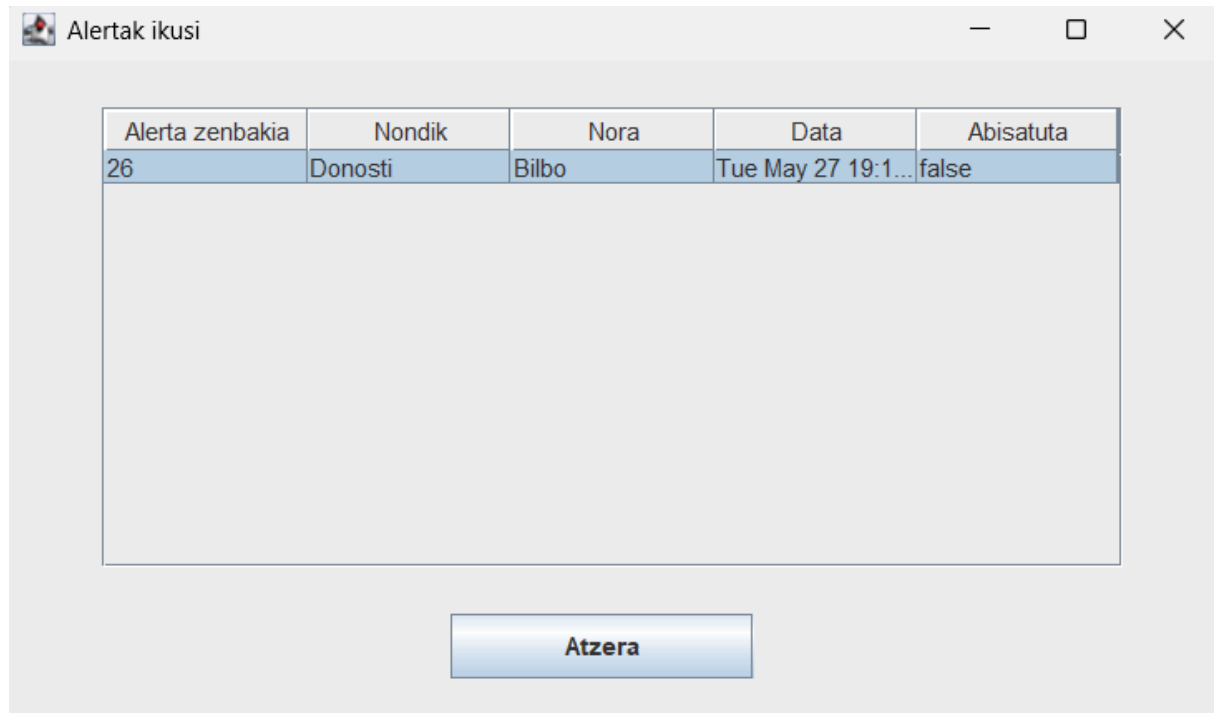
### # OINARRIZKO FLUXUA:

1. Bidaiariak irteera hiria eta helmuga hiria eta sartzen ditu.
2. Bidaiariak egun bat aukeratzen du.
3. System-ek alerta sortu eta datu-basean gordetzen du.

### # FLUXU ALTERNATIBOA:

1. Irteera hiria edo helmuga hiria ez dira testu motakoak. System-ek bidaiaria jakinarazi.
2. Irteera hiria, helmuga hiria eta data ezin dira hutsik egon. System-ek bidaiariari jakinarazi.
3. Bidaiariak atzera botoiari ematen dio. System-ek hasiera orrira eraman.

## 19. Alerta ikusi (bidaiari)



## 22. alerta ikusiren interfaze grafikoa

### # OINARRIZKO FLUXUA:

1. System-ek, bidaiariak programatuta dituen alertak pantailaratu.

### # FLUXU ALTERNATIBOA:

1. Ez dago alertarik. Amaitu.
2. Atzera botoiari eman eta System-ek Bidaiaria hasiera orrira eraman.



## 20. Bidaia kontsultatu (ez erregistratua)

**Bidaia eguna**

Nondik:

Nora:

May 2025

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18					1	2	3
19	4	5	6	7	8	9	10
20	11	12	13	14	15	16	17
21	18	19	20	21	22	23	24
22	25	26	27	28	29	30	31

**Bidaiak: 2025 May 14**

Gidaria	Eserlek...	Prezioa
Alejandra Redondo	1	3.0

**Itxi**

## 23. irudia: bidaia kontsultaturen interfaze grafikoa

### # OINARRIZKO FLUXUA:

1. System-ek ibilaldiak abiatzen diren hiri guztiak bistaratzen ditu.
2. Gidaria-k irteerako hiria hautatzen du.
3. System-ek hautatutako hiritik abiatzen diren bidaien helmugako hiri guztiak bistaratzen ditu.
4. Gidaria-k helmuga hiria hautatzen du.
5. System-ek egutegi batean irteera-tik helmugarako hirietara aukeratutako hilabete batean ibilaldiak dauden egunak nabarmentzen ditu.
6. Gidaria-k egun bat aukeratzen du egutegian.
7. System-ek aukeratutako irteera hiritik aukeratutako helmuga hirira, egun horretan, dauden bidaia bistaratzen ditu.

### # FLUXU ALTERNATIBOA:

1. Ez dira existitzen bidaia aukeratutako egunean. Bidaia pantaila hutsik dago.
2. Itxi botoiari eman. System-ek hasiera orrira eraman.

## 21. Erregistratu

**Izena**

**Abizena**

**Jaiotze data** May 2025

	S...	M...	T...	...	T...	Fri	Sat
18					1	2	3
19	4	5	6	7	8	9	10
20	11	12	13	14	15	16	17
21	18	19	20	21	22	23	24
22	25	26	27	28	29	30	31

**Sexua** ☐ **Gizona** ☐ **Emakumea**

**Erabiltzaile mota** ☐ **Gidaria** ☐ **Bidaia**

**Email**

**Pasahitza**

**Atzera** **Erregistratu**

24. irudia: erregistraturen interfaze grafikoa

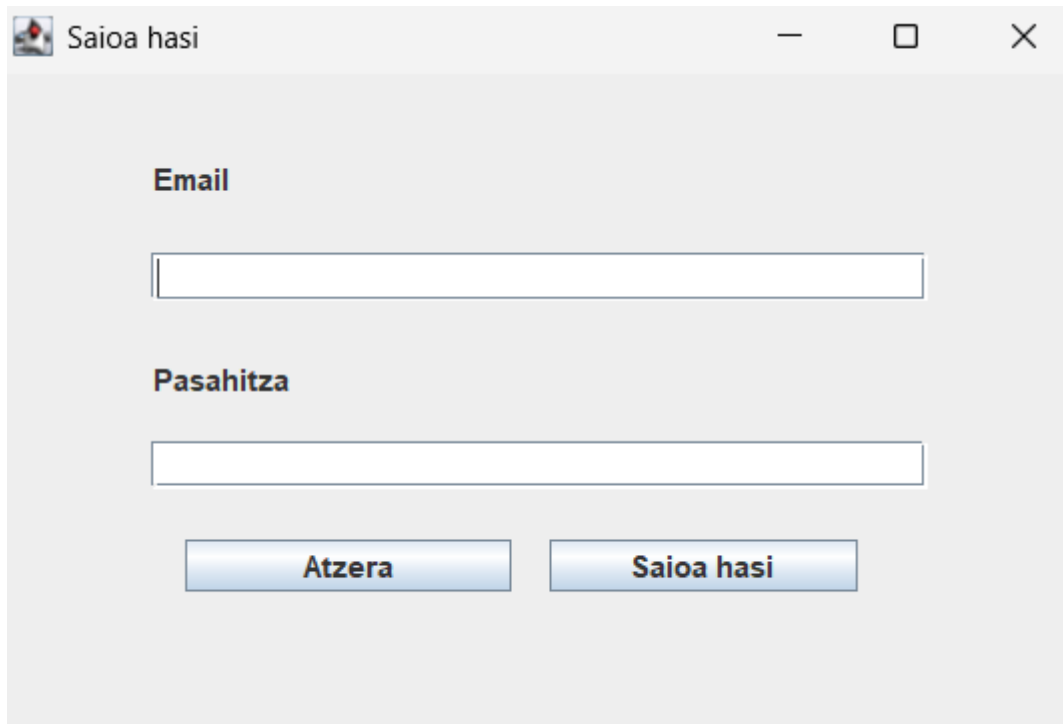
### # OINARRIZKO FLUXUA:

1. Erabiltzaile ez erregistratua-k izena, abizena, jaiotze data, sexua, erabiltzaile mota, email kontua eta pasahitza sartzen ditu.
2. System-ek gidaria edo bidaiari bat sortu eta datu basean gordeko du, erabiltzaile ez erregistratuak aukeratzen duenaren arabera.

### # FLUXU ALTERNATIBOA:

1. Atzera botoia sakatu eta System-ek hasiera orrira eraman.
2. Eremu batzuk hutsik daude. System-ek erabiltzaileari jakinarazi.
3. Jaiotze data adin txikikoa edo zentzurik gabekoa da. System-ek erabiltzaileari jakinarazi.
4. Izena edo abizena ez dira testu motakoak. System-ek erabiltzaileari jakinarazi.
5. Erabiltzailea iadanik existitzen da. Erabiltzailea ez da sortzen. System-ek erabiltzaileari jakinarazi.

## 22. Login



The image shows a graphical user interface for a login system. The window has a title bar with the text 'Saioa hasi' and standard window control buttons (minimize, maximize, close). The main area contains two labels: 'Email' and 'Pasahitza' (Password). Below each label is a text input field. At the bottom of the window, there are two buttons: 'Atzera' (Back) and 'Saioa hasi' (Login).

25. irudia: loginaren interfaze grafikoa

### # OINARRIZKO FLUXUA:

1. Email kontua eta pasahitza sartzen ditu erabiltzaileak.
2. System-ek hasiera orrira bidaltzen dio. Gidaria bada, azken honen hasiera orrira, bidaiaria bada, azken honen hasiera orrira eta administratzailea bada, azken honen hasiera orrira.

### # FLUXU ALTERNATIBOA:

1. Email kontua edo pasahitza ez dira testu motakoak edo zuzenak. System-ek erabiltzailea jakinarazi.
2. Atzera botoia sakatu eta System-ek hasiera orrira eraman.
3. Erabiltzailea ez da oraindik datu basean existitzen. System-ek erabiltzailea jakinarazi.

### 23. Erreklamazioa kudeatu (administratzailea)

The screenshot shows a window titled 'Administratzailea' with a table of complaints and a summary section.

Erreklamazio z...	Email	Deskripzioa	Egoera
6	driver@gmail.c...	esperientzia tx...	deuseztatu
3	driver@gmail.c...	berandu iritsi ga...	deuseztatu

Mezua
Bidaiaari: nire dirua bueltan .
Gidari: ez

Buttons: **Onartu** (Accept) and **Deuseztatu** (Cancel/Dismiss).

26. irudia: administratzailearen erreklamazio kudeaturen interfaze grafikoa

#### # OINARRIZKO FLUXUA:

1. System-ek erreklamazioen zerrenda pantailaratu.
2. Adminak erreklamazio bat aukeratzen du.
3. Adminak erreklamazioa onartzen du eta System-ek erreklamazioaren egoera eguneratzen du, bidaiaari dirua itzuli eta gidariari dirua kentzen dio.
4. Adminak deuseztatu botoiari ematen dio eta erreklamazio listatik kentzen da.

#### # FLUXU ALTERNATIBOA:

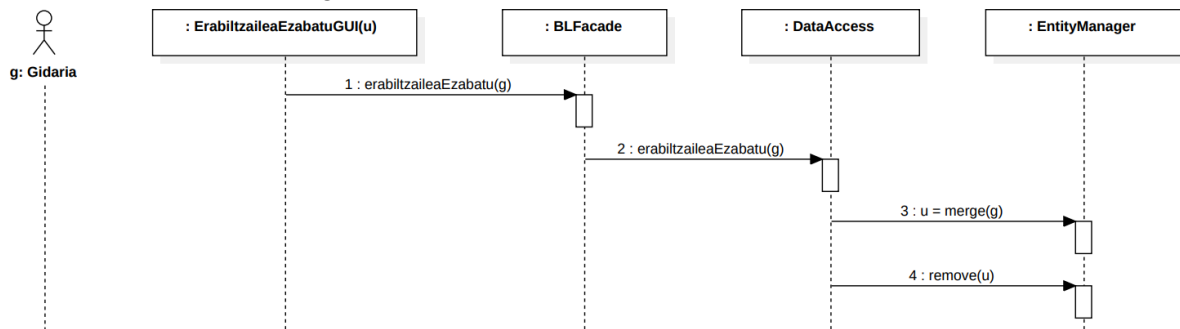
1. Ez dago erreklamaziorik. Amaitu.

## 6. Diseinua

Proiektu osoaren StarUML-a implementazioaren zip-aren barruan dago (Rides24 karpeta).

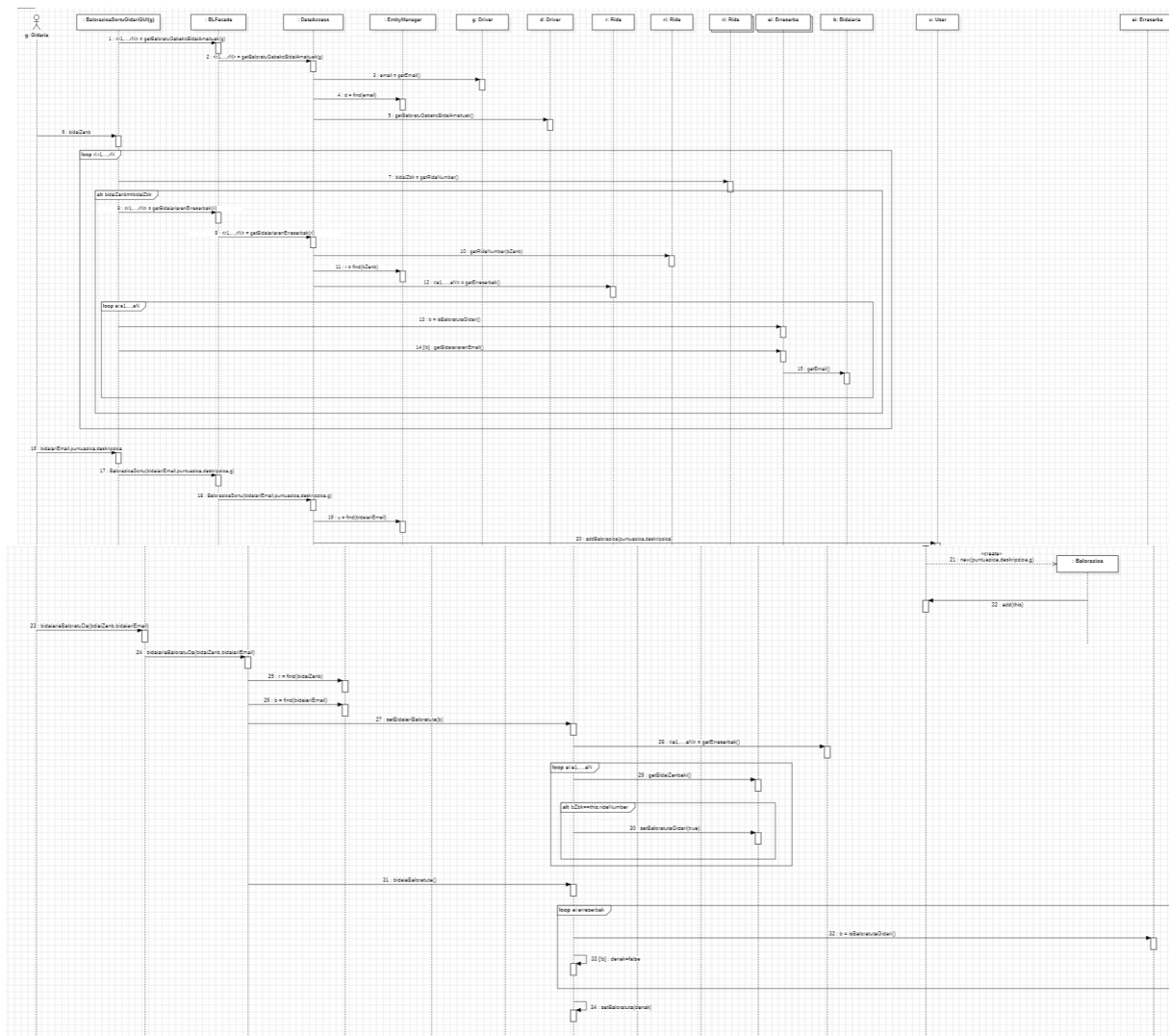
### 6.1. Sekuentzia diagramak

#### 1. Erabiltzailea ezabatu (gidari)



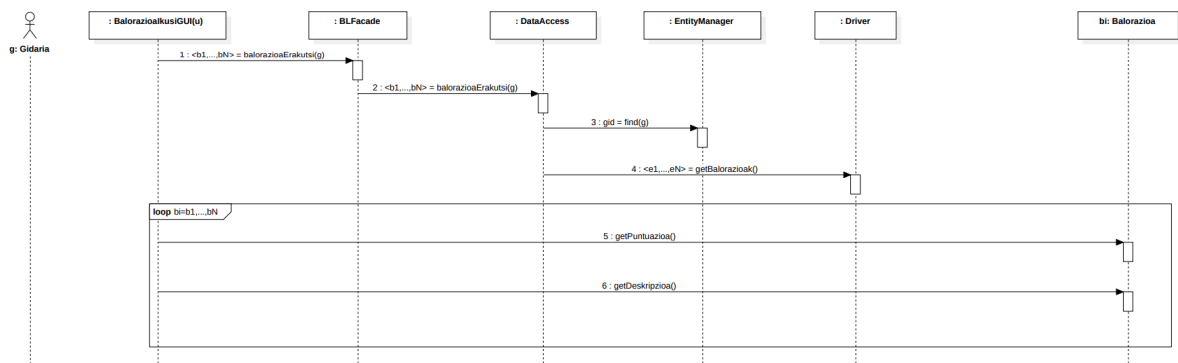
27. irudia: Erabiltzailea ezabatu gidari-ren sekuentzia diagrama

## 2. Balorazioak sortu (gidari)



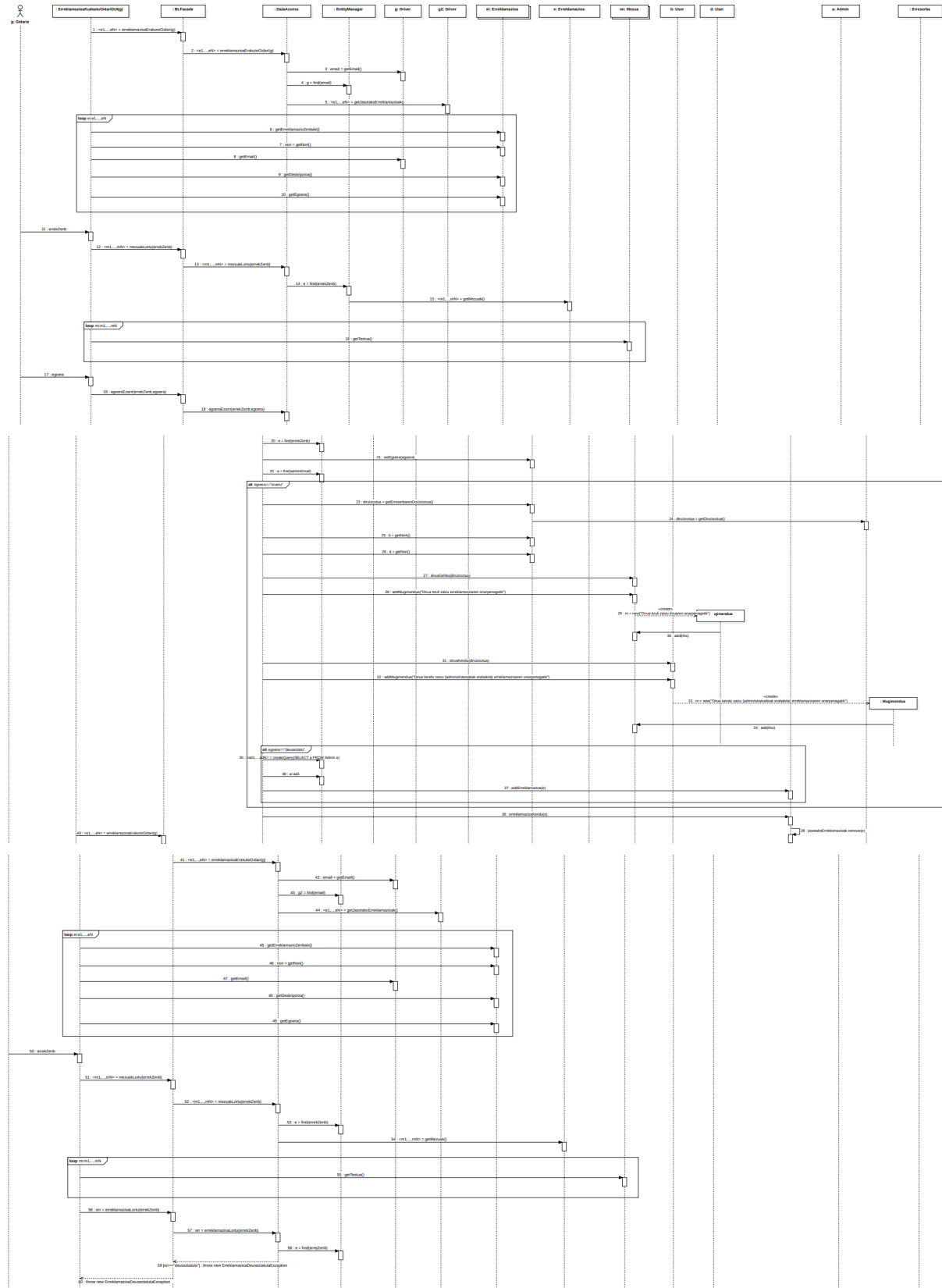
28. irudia: Balorazioa sortu gidari-ren sekuentzia diagrama

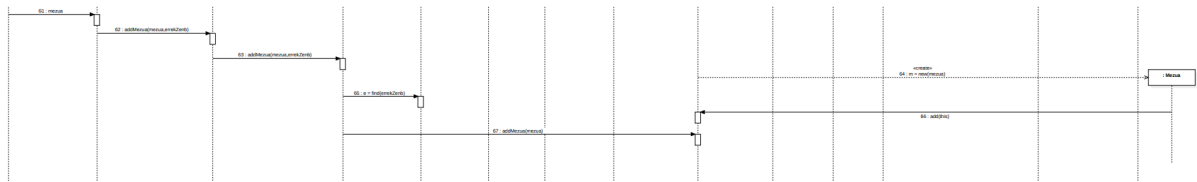
## 3. Balorazioa ikusi (gidari)



29. irudia: Balorazioa ikusi gidari-ren sekuentzia diagrama

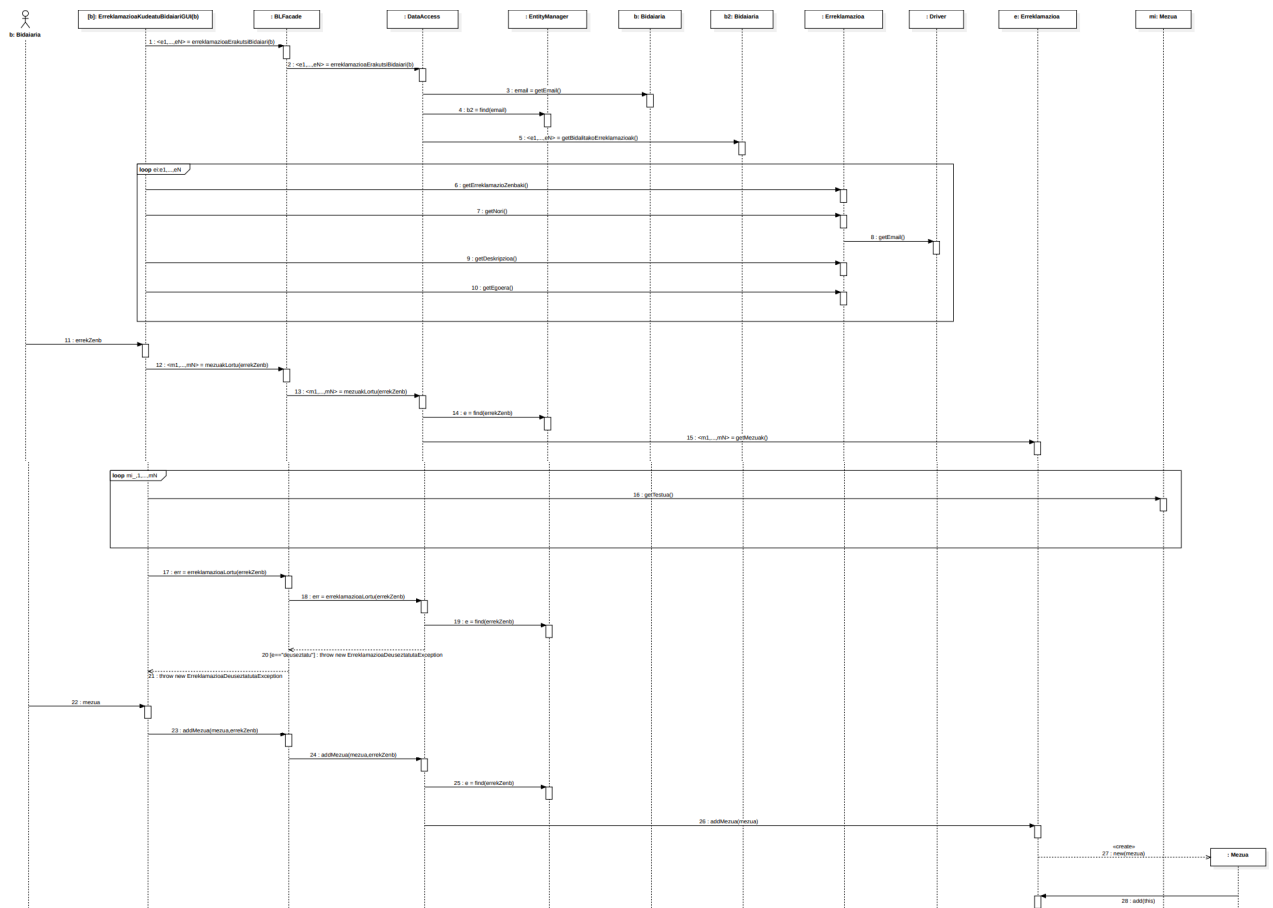
#### 4. Erreklamazioa kudeatu (gidari)





30. irudia: Erreklamazioa kudeatu gidari-ren sekuentzia diagrama

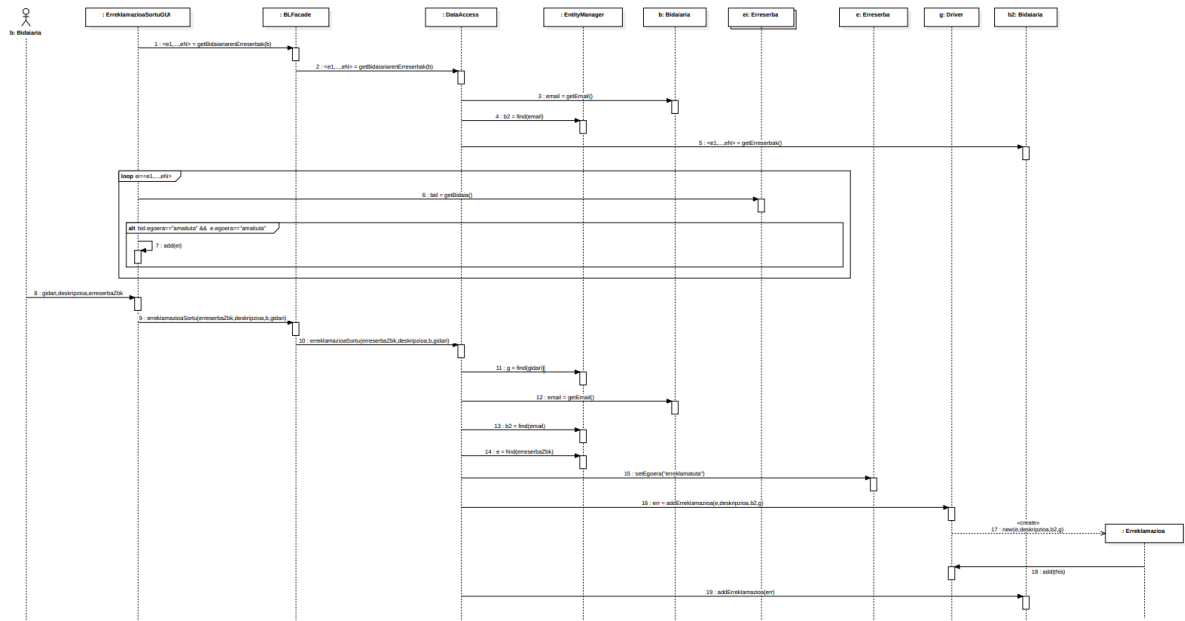
## 5. Erreklamazioa kudeatu (bidaiari)



31. irudia: Erreklamazioa kudeatu bidaiari-ren sekuentzia diagrama

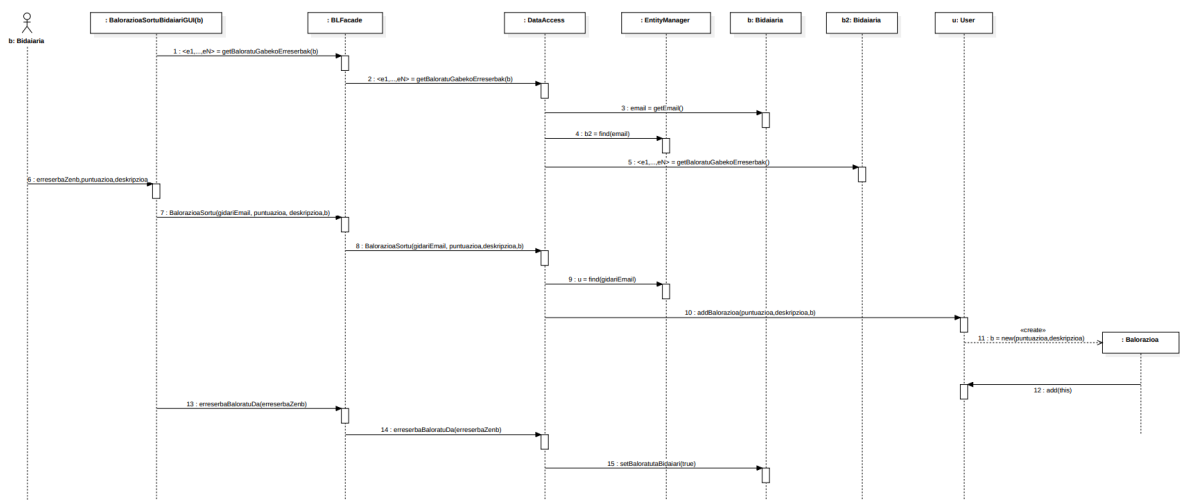


## 6. Erreklamazioa sortu (bidaiari)



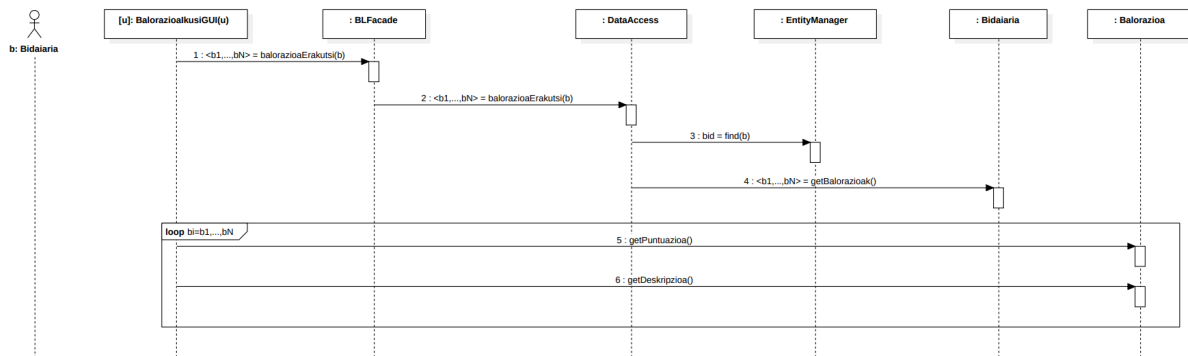
32. irudia: Erreklamazioa sortu bidaiari-ren sekuentzia diagrama

## 7. Balorazioa sortu (bidaiari)



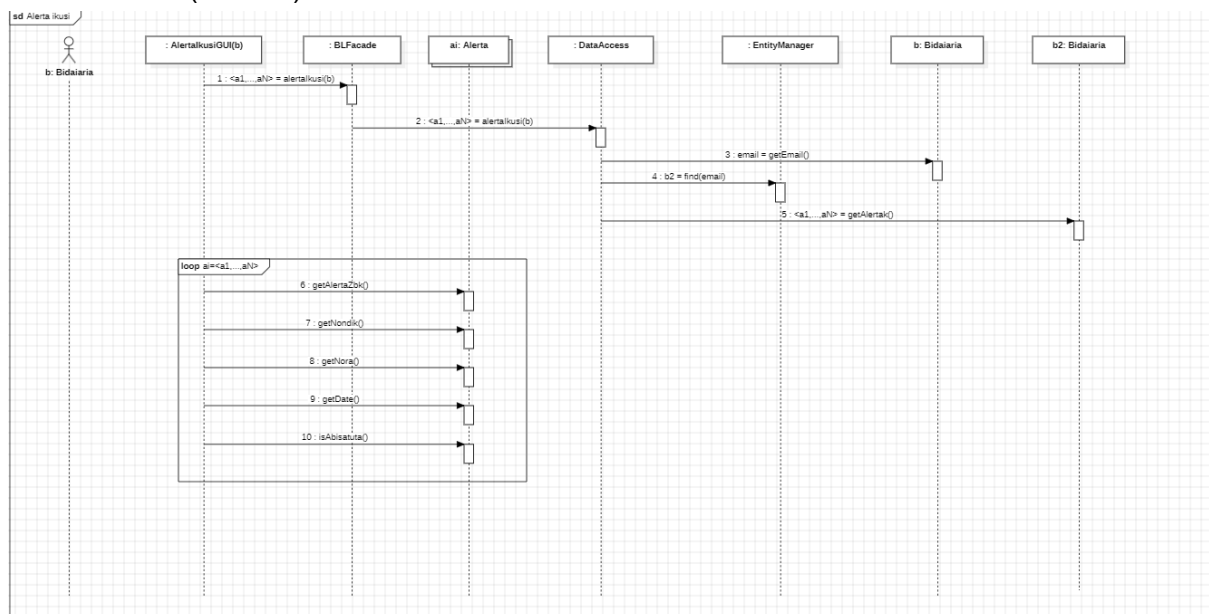
33. irudia: Balorazioa sortu bidaiari-ren sekuentzia diagrama

## 8. Balorazioa ikusi (bidaiari)



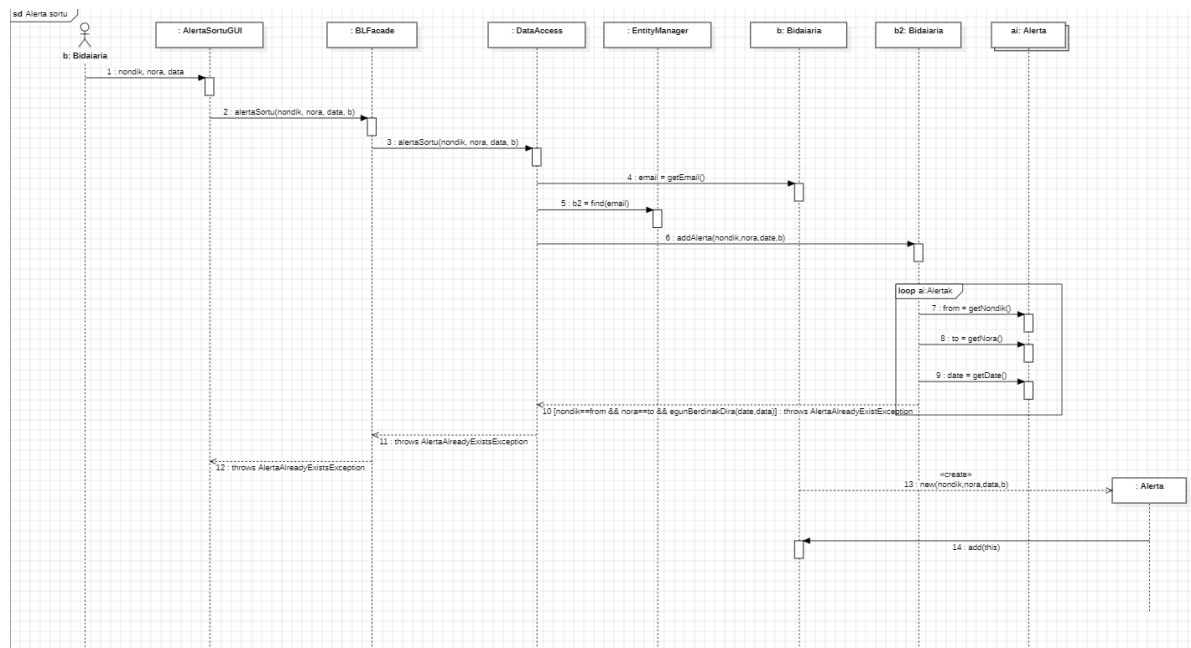
34. irudia: Balorazioa ikusi bidaiari-ren sekuentzia diagrama

## 9. Alerta ikusi (bidaiari)



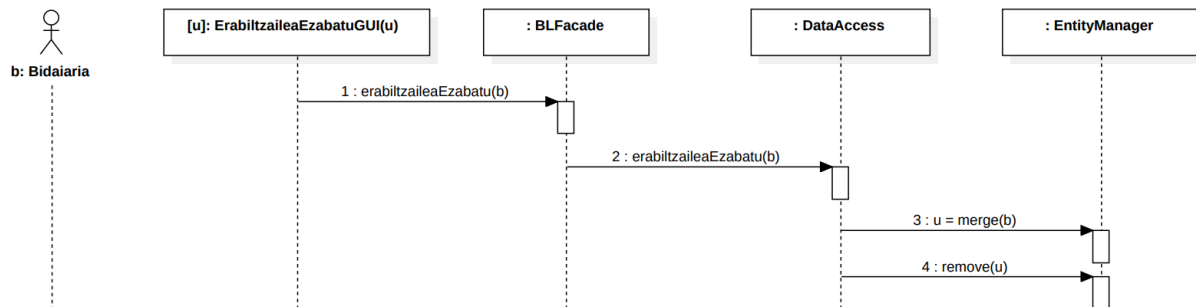
35. irudia: Alerta ikusi bidaiari-ren sekuentzia diagrama

## 10. Alerta sortu (bidaiari)



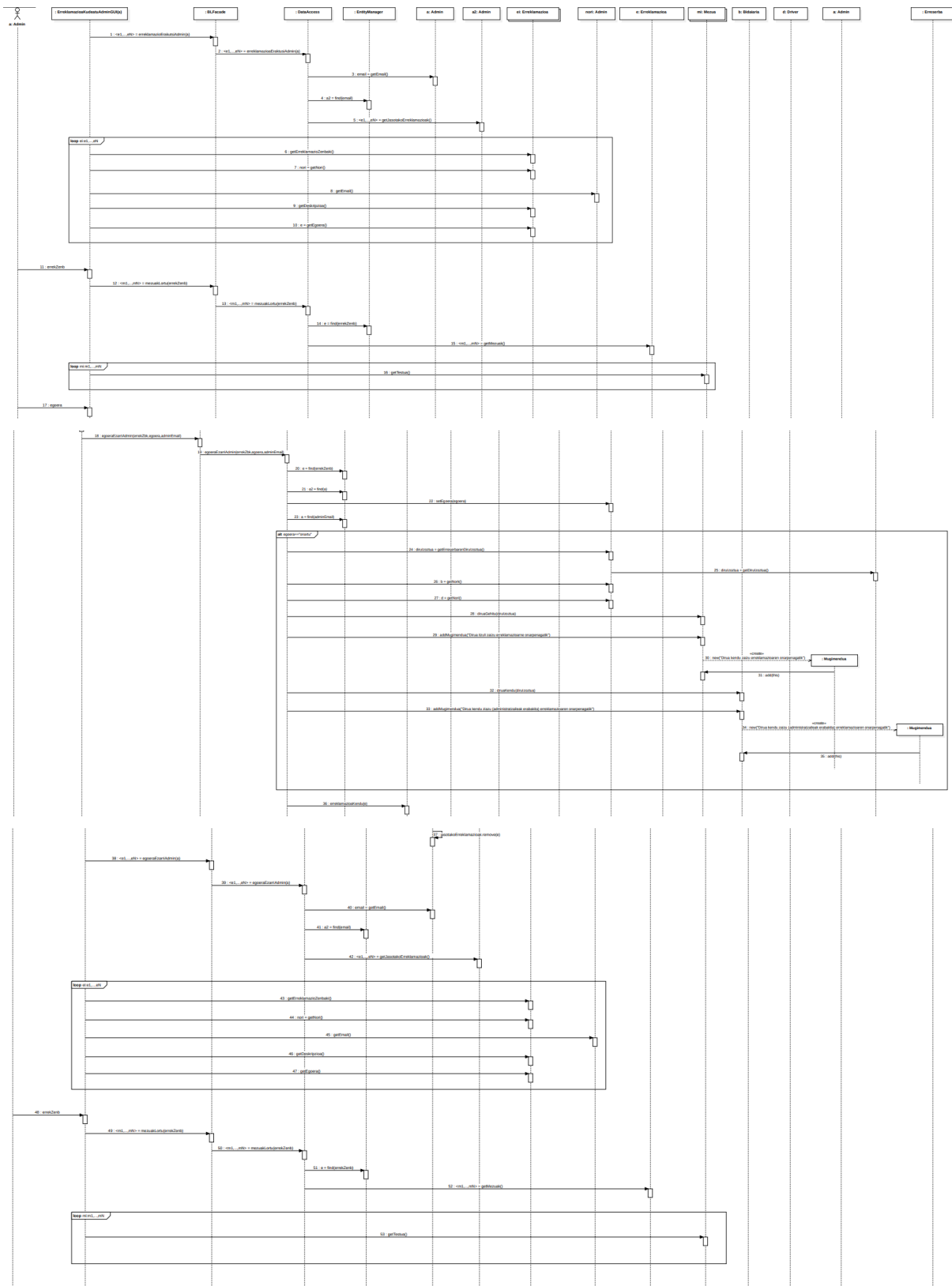
36. irudia: Alerta sortu bidaiari-ren sekuentzia diagrama

## 11. Erabiltzailea ezabatu (bidaiari)



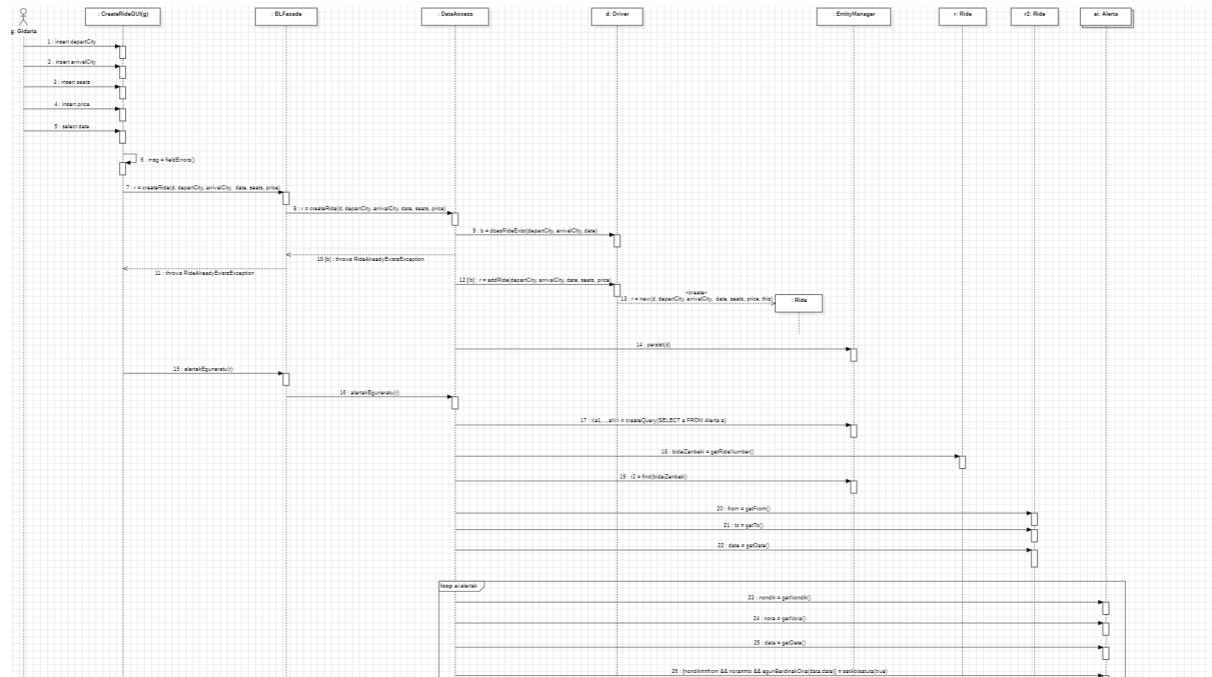
37. irudia: Erabiltzailea ezabatu bidaiari-ren sekuentzia diagrama

## 12. Erreklamazioa kudeatu (administratzailea)



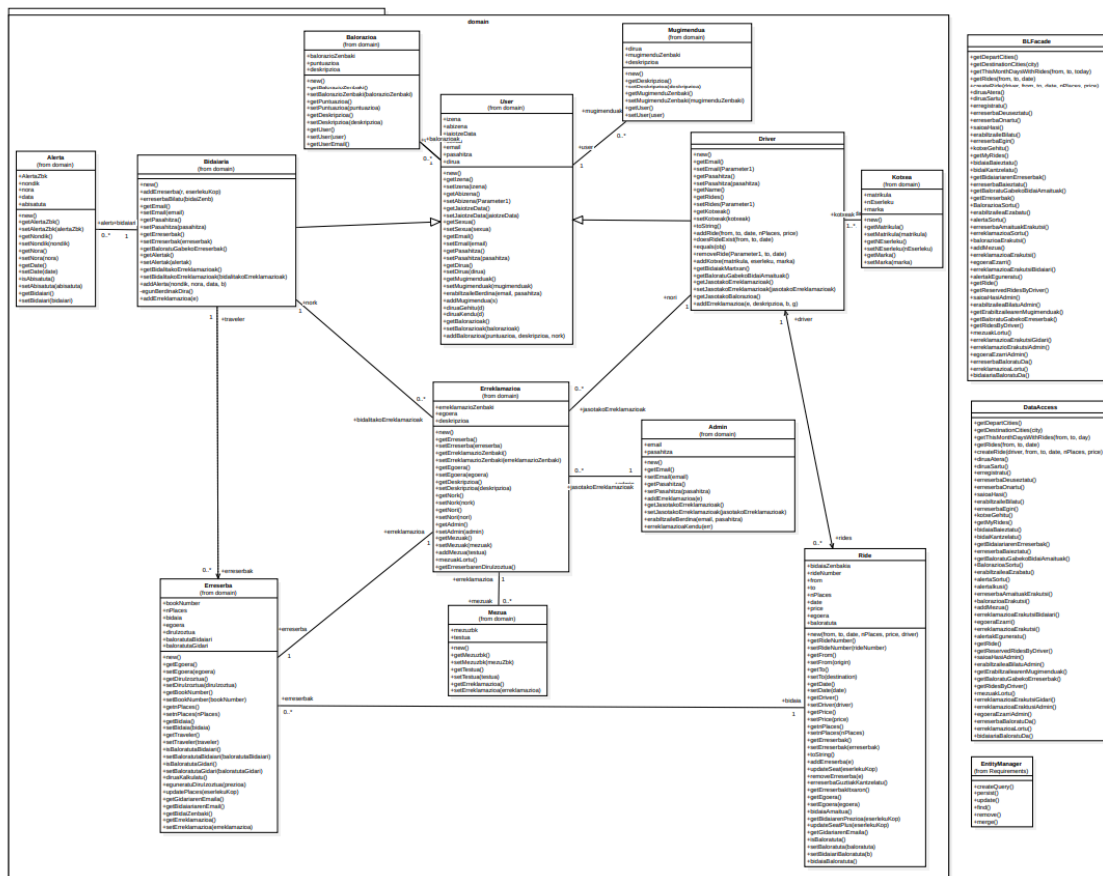
38. irudia: Erreklamazioa kudeatu admin-ren sekuentzia diagrama

### 13. Bidaiak sortu (gidari)



39. irudia: Bidaiak sortu-ren sekuentzia diagrama

## 6.2 Klase diagrama



40. irudia: aplikazioaren klase diagrama

## 7. Implementazioa

Atal honetan Negozio Logikako interfaze klaseko metodoen deskribapena egingo dugu.

```
public void initializeBD();
```

Metodo honek datu-sarbidea deitzen du, datu-basea gertaera eta galdera batzuekin hasieratzeko.

```
public List<String> getDepartCities();
```

Metodo honek bidaiak abiatzen diren hiri guztiak itzultzen ditu.

```
public List<String> getDestinationCities(String from);
```

Metodo honek hiri jakin batetik abiatzen diren bidaia guztietako helmuga guztiak itzultzen ditu.

```
public Ride createRide( String from, String to, Date date, int nPlaces, float price, String driverEmail) throws RideMustBeLaterThanTodayException, RideAlreadyExistException;
```

Metodo honek gidari batentzako bidaia bat sortzen du, emandako parametroarekin, nondik, nora, data, eserleku kopurua, prezioa eta gidariaren emaila. Metodo honek RideMustBeLaterThanTodayException eta RideAlreadyExistException salbuespenak altxa ditzake. Lehenengo salbuespena sortuko den bidaia uneko eguna baino ondoren izateko bermatzen du. Bigarren salbuespena, berriz, bermatzen du ez dela bi bidai berdin existitzen jatorri, helmuga eta data berdinarekin.

```
public List<Ride> getRides(String from, String to, Date date);
```

Metodo honek jatorri, helmuga eta data jakin bateko bidaiak berreskuratzen ditu.

```
public List<Date> getThisMonthDatesWithRides(String from, String to, Date date);
```

Metodo honek hilabetearen honetan dauden bidaiak eta haien datak berreskuratzen ditu datu-basetik.

```
public boolean erregistratu(String izena, String abizena, Date jaiotzeData, String sexua, String erabiltzaileMota, String email, String pasahitza);
```

Metodo honek kontu bat erregistratzen du eta emandako parametroak kontu horren informazio gisa gordetzen ditu.

```
public boolean saioaHasi(String email, String pasahitza);
```

Metodo honek emandako emailarekin erabiltzailea datu-basean bilatzen du, eta pasahitza ondo dagoen egiaztatzen du. Hala bada, true itzultzen du, bestela, false.

```
public boolean saioaHasiAdmin(String email, String pasahitza);
```

Metodo honek emandako administratzailearen emailarekin administratzailea bilatzen du datu-basean, eta pasahitza ondo dagoen egiaztatzen du. Hala bada, true itzultzen du, bestela, false.

```
public User erabiltzaileaBilatu(String email);
```

Metodo honek emandako emailarekin erabiltzailea bilatzen du datu-basean.

`public Admin erabiltzaileaBilatuAdmin(String email);`

Metodo honek emandako emailarekin administratzailea bilatzen du datu-basean.

`public void diruaSartu(float dirua, Bidaia t);`

Metodo honek emandako bidaia dirua gehitzen dio eta mugimendua sortzen eta gehitzen dio bidaia.

`public void diruaAtera(float dirua, Driver d);`

Metodo honek emandako gidaria dirua kentzen dio eta mugimendua sortzen eta gehitzen dio bidaia.

`public List<Erreserba> getReservedRidesByDriver(Driver d);`

Metodo honek emandako gidaria sortutako bidaia guztien erreserba guztiak bueltatzen ditu.

`public void erreserbaEgin(int bidaiZenbaki,Bidaia bidaia, int eserlekuKop);`

Metodo honek bidaia baten erreserba bat sortzen edo eguneratzen du, bidai horren eserleku batzuk erreserbatuz. Bidaia dirua izozten zaio eta mugimendua sortzen eta gehitzen dio bidaia.

`public void erreserbaOnartu(int erreserbaZenbaki, Driver gidari);`

Metodo honek gidaria dituen erreserbetatik erreserbaZenbaki duen erreserbaren egoera "onartu" bihurtzen du.

`public void erreserbaDeuseztatu(int erreserbaZenbaki);`

Metodo honek gidaria dituen erreserbetatik erreserbaZenbaki duen erreserbaren egoera "deuseztatu" bihurtu, erreserba hori egin duen bidaia dirua bueltatu, bidaiaren eserlekuak eguneratu eta mugimendua sortzen eta gehitzen dio bidaia.

`public void erreserbaBaieztatu(int erreserbaZenbaki);`

Metodo honek erreserba bat "amaituta" egoerara bihurtu, gidaria dagokion dirua gehitu, mugimendua sortu eta baldin eta erreserba guztiak "amaituta" egoeran badaude, bidaiaren egoera "amaituta" bihurtzen du.

`public Kotxea kotxeGehitu(String matrikula, int eserleku, String marka, Driver gidari) throws KotxeaAlreadyExistException;`

Metodo honek gidaria kotxe berri bat gehitzen dio eta KotxeaAlreadyExistException salbuespena altxa dezake, non bermatzen den ez dela bi kotxe matrikula berarekin existitzen.

`public int getnPlaces(String matrikula);`

Metodo honek emandako matrikula duen kotxearen eserleku kopurua bueltatzen du.

`public List<Erreserba> getBidaiaErreserbak(Bidaia b);`

Metodo honek bidaia egindako erreserba guztiak bueltatzen ditu.

`public List<Kotxea> kotxeakEskuratu(Driver d);`

Metodo honek gidaria dituen kotxe guztiak bueltatzen ditu.

```
public List<Mugimendua> getErabiltzailearenMugimenduak(User u);
```

Metodo honek erabiltzailearen mugimendu guztiak bueltatzen ditu.

```
public void bidaiKantzelatu(Driver gidari,int bidaiZenbaki);
```

Metodo honek gidariaren bidaiZenbaki hori duen bidaiaren egoera eta bidai horren erreserba guztien egoera “kantzelatu” bihurtu, dirua bueltatu eta mugimendua sortzen eta gehitzen die erreserbak egin duten bidaiari.

```
public List<Ride> getRidesByDriver(Driver d);
```

Metodo honek gidariak sortutako “martxan” egoeran dauden bidai guztiak bueltatzen ditu.

```
public List<Balorazioa> balorazioaErakutsi(User u);
```

Metodo honek erabiltzailearen balorazio guztiak bueltatzen ditu.

```
public void erabiltzaileaEzabatu(User u);
```

Metodo honek erabiltzailea datu-basetik ezabatzen du.

```
public void balorazioaSortu(String nori, int puntuazioa, String deskripzioa, User nork);
```

Metodo honek emandako puntuazioa eta deskripzioa duen balorazio berri bat gehitzen dio balorazioa jasotzen duen erabiltzaileari.

```
public void erreserbaBaloratuDa(Integer erreserbaZenbaki);
```

Metodo honek “baloratutaBidaiari” egoera true bihurtzen du, bidaiaria erresebaren bidez gidariari balorazioa egin duela esateko.

```
public void bidaiariaBaloratuDa(Integer bidaiZenbaki, String b);
```

Metodo honek “baloratutaGidari” egoera true bihurtzen du, gidaria bidaiaren erreserbaren bidez bidaiariari balorazioa egin duela esateko.

```
public void erreklamazioaSortu(int erreserbazbk, String deskripzioa, Bidaiaria b, String gidari);
```

Metodo honek bidaiariak gidariari erreserba baten inguruko erreklamazio bat sortzen dio, eta bien erreklamazio listan gehitzen da. Erreserbaren egoera “erreklamatusa” bihurtu, bidaiaria erreklamazioa sortzen du eta gidaria jasotzen du.

```
public List<Ride> getBaloratuGabekoBidaiaAmaituak(Driver d);
```

Metodo honek baloratuta gabe eta amaituta dauden gidariaren bidai guztiak bueltatzen ditu.

```
public List<Erreserba> getBidaiarenErreserbak(Ride r);
```

Metodo honek bidai baten erreserba guztiak bueltatzen ditu.

```
public List<Erreserba> getBaloratuGabekoErreserbak(Bidaiaria b);
```

Metodo honek bidaiari baten baloratuta ez dauden erreserba guztiak bueltatzen ditu.

```
public List<Alerta> alertalkusi(Bidaiaria b);
```

Metodo honek bidaiaria sortutako alerta guztiak bueltatzen ditu.



```
public void alertaSortu(String nondik, String nora, Date data, Bidaia b) throws
AlertaAlreadyExistException;
```

Metodo honek bidaia badai jakin baterako alerta bat sortzen du. Alerta hori jadanik existitzen bada salbuespena altxatzen da.

```
public List<Erreklamazioa> erreklamazioaErakutsiBidaia(Bidaia b);
```

Metodo honek bidaia bidalitako erreklamazio guztiak itzultzen ditu.

```
public List<Erreklamazioa> erreklamazioaErakutsiAdmin(Admin a);
```

Metodo honek administratzailea jasotako erreklamazio guztiak itzultzen ditu.

```
public List<Erreklamazioa> erreklamazioaErakutsiGidari(Driver g);
```

Metodo honek gidaria jasotako erreklamazio guztiak itzultzen ditu.

```
public void addMezua(String testua, int errekzkbk);
```

Metodo honek testua edukia duen mezua gehitzen dio erreklamazioari.

```
public Erreklamazioa erreklamazioaLortu(int errekzkbk);
```

Metodo honek errekzkbk duen erreklamazioa itzultzen du.

```
public void egoeraEzarriAdmin(int errekzkbk, String egoera, Admin a);
```

Metodo honek administratzaile batek erreklamazio baten egoera ezarri eta horren ondorioz egindako diru-mugimenduak sortzen ditu.

```
public void egoeraEzarri(int errekzkbk, String egoera) throws
erreklamazioaEbatzitaException;
```

Metodo honek erreklamazio baten egoera ezarri eta horren ondorioz egindako diru-mugimenduak sortzen ditu.

```
public void alertakEguneratu(Ride r);
```

Metodo honek alertaren egoera eguneratzen du, eta erabiltzaileei abisatzeko prest dauden alertak markatzen ditu bidaia bat sortzean bidai hori interesekoa denean (alertaren interesa bidaiarekin konparatu).

```
public List<Mezua> mezuakLortu(int errekzkbk);
```

Metodo honek erreklamazio baten mezu guztiak bueltatzen ditu.

```
public Ride getRide(int bidaiZenbaki);
```

Metodo honek bidai zenbakia duen bidaia bilatzen du datubasean eta aurkitutako bidaia bueltatzen du.

## 8. Ondorioak

### 8.1. Pentsatutako hobekuntzak

Diseinuari denbora gehiago eskaini, inplementazioa hasi baino lehen.

### 8.2. Lan taldearen dinamika

Denon artean egin dugu lana, horrela, zalantza sortzerakoan zuzenean taldekidearen iritzia edo laguntza jakin dezakegu.

### 8.3. Proiektuaren konplexutasuna

Proiektua oso luzea izan da eta zailtasun pixka bat du. Bereziki zailtasunaren parte handiena erroreak zuzentzea da.

### 8.4. Lortutako emaitzen poztasuna

Lortutako emaitzarekin oso pozik gaude. Azkenean egin nahi genuen guztia lortu dugu.

### 8.5. Gomendioak eta iradokizunak hurrengo kurtsoko ikasleentzat

Ahalik eta lehenago hastea proiektua egiten, ez utzi azkenengo momenturako, zeren nahiko luzea izan da proiektua. Proiektuarekin hasi baino lehen denbora hartu Eskakizun Bilketa eta diseinu on bat egiteko, horrela inplementazioa askoz ere errazago eta zuzenago izango da.

### 8.6. Proiektua garatzeko erabili den metodologia

Egin behar ditugun lanak zerrendatu eta egokia den ordenean joan egiten. Zalantzak sortuz gero, gure artean galdetu edo irakasleari.

### 8.7. Proiektuetan oinarritutako irakaskuntzari eta irakasgaiari buruzko iritzia edo kritika

Proiektua luzeegia dela uste dugu, baina, honen ondorioz, asko ikasi dugu proiektua egiterakoan.

## 9. Bideoaren URL-a

Bideoaren URL-a: <https://youtu.be/YrVWbc4tDpw>

## 10. Kodea

Proiektuaren kodea zip fitxategiaren bidez entregatzen dugu (2.1.zip).