

Bidaiak

SI

Beñat Lopez, Ekaitz Pinedo eta Urtzi Etxegarai
2.2 Taldea

Sarrera	3
Eskakizun Bilketa	4
Domeinuaren eredua:	4
linka.....	4
Entitateen eta erlazioen azalpena:.....	4
Erabilpen kasuen eredua:	7
Gertaera Fluxuak eta GUI-ak:	8
Diseinua	32
Sekuentzia diagramak eta azalpenak(Hobeto ikusteko erabili link-a, link-ak erabilpen kasuen izenak dira):	32
Klase diagrama:	40
Implementazioa	42
Ondorioak	53
Bideoaren URL-a	54

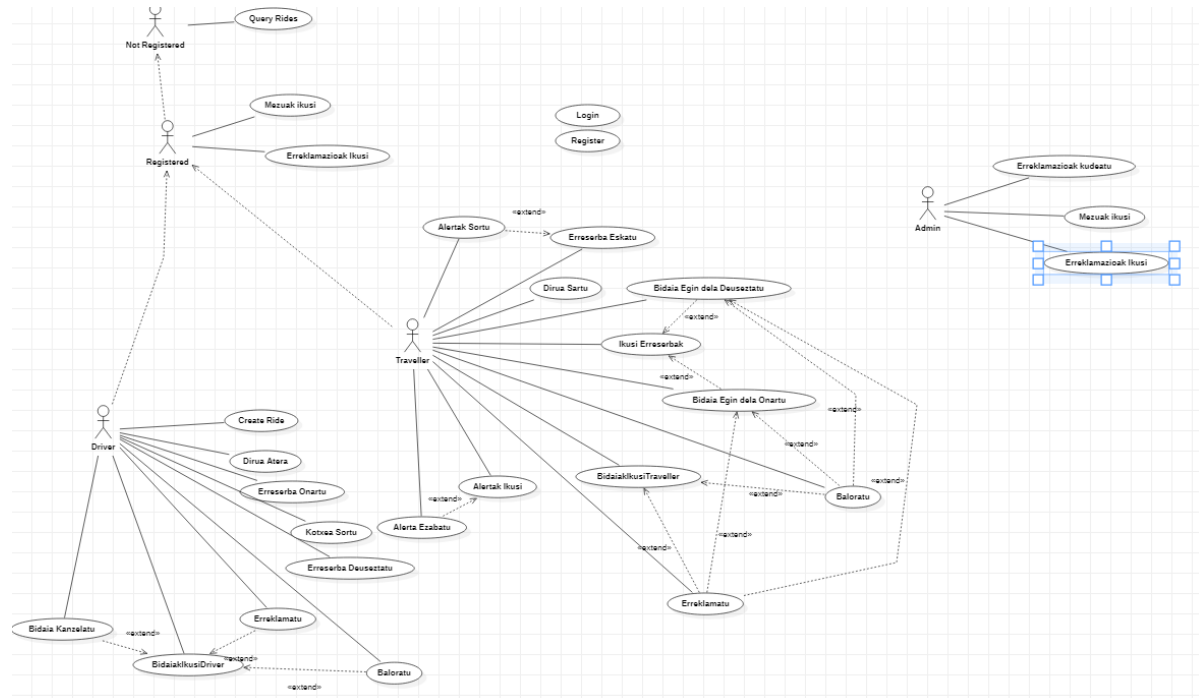
Sarrera

Gure proiektua bidaiak antolatzeko aplikazio bat da, zeinetan Web Zerbitzuak eta Internalizazioa garatu dugu. Bilatzen dugu sinplea eta intuiboa izatea erabiltzaile guztientzat, eta hau lortzen dugu interfaze argiak erabiliz eta bakarrik beharrezkoa den informazioa erakutsiz. Gure aplikazioa bidai bat hainbat geldiunekin egin daiteke, baloraketa sistema bat dauka bidaiarientzat zein gidarientzat eta erreklamazio sistema bat administratzaile batzuk kudeatuko dutena.

- **Driver:** Gidariei lotutako klasea da, Driver klasea, profile klasearen klase ume bat da, beraz Profile-k berez dituen erlazio eta atributuak izango ditu. Gainera, gidariek bidaia ezberdinak izan ditzakete. Orduan, Driver-klaseak 0 edo gehiago Ride izan ditzake, bestalde, gidari batek hainbat kotxe izan ditzake, beraz 0 edo gehiago Kotxe izan ditzake baitan
- **Admin:** Admin-ak Profile-k berezkoak dituen erlazioak heredatu egiten ditu, ez du ezer gehiago berezirik behar atributu moduan.
- **Kotxe:** Kotxe klasea, egon daitezkeen kotxeak englobatzen ditu, kotxe bat bidaia bat edo gehiagotan egon daiteke, eta kotxe bat gidari batena da, ezin dute bi gidari ezberdinek kotxe bera eduki. Beraz, kotxeak 0..* (0 edo gehiago) Ride ditu eta kotxeak 1 Driver du.
- **Ride:** Ride klaseak adierazi egiten duena da egon daitezkeen bidaia ezberdinak, helmuga, irteera Data, prezio... ezberdinekin. Bidaia bat beti izango da Driver batena, gainera Bidaia batek beti izango du kotxe bat. Horrez gain, geltoki anizkoitzeko bidai bat denez, hainbat geltoki ditu bidaia batek, gutxienez bi. Azkenik, bidaia batek hainbat erreserba eskari izan ditzake. Beraz, Ridek 1 Gidari, 1 Kotxe, 0..*(0 edo gehiago) RideRequest, 2..*(2 edo gehiago) Geltoki ditu.
- **Geltoki :** Geltoki klasea klase berezi bat da “Emmendable” moduan gordetzen dena, ez da entity bat baina Ride-en laguntzailea da, gordetzeko bidaia bat zein geltokitik pasa egiten den, beraz gutxienez menzionatzea ez dago gaizki.
- **RideRequest:** RideRequest klaseari dagokionez, request bat beti izango da bidaia batena, eta baita ere gidari batek egingo du rideRequest hori. Beraz RideRequest batek 1 Traveller eta 1 Ride ditu.
- **Alerta:** Alerta klasea da, klase bat Traveller batek sortu egiten duena jakinarazteko Alerta sortu den egunean Bidaia bat dagoen alerta pasa egiten den tokietatik pasa egiten dena. Beraz, Alerta batek 1 Traveller du ez du bidarik behar.
- **Erreklamazioa:** Erreklamazio klasea, bidaia bat pasatu eta gero, bai Traveller edo bai Driver-ek kexu bat gordetzeko sortutako klase bat da. Erreklamazio klaseari dagokionez, Traveller edo Driver batek sortu egiten du (Profile batek) eta Traveller edo Driver bateri zuzenduta dago (profile). Beraz esan daiteke Profile batek beste Profile bati egiten diola erreklamazioa. Gauza da, erreklamazio hori beste profile batek tratatu dezakeela (Admin). Gainera, erreklamazio bakoitza erreserbaka egiten da, baina bidaia bakoitza gehienez aldi batez erreklamatu daiteke. Beraz, erreklamazioak 2..3 (2 edo 3) Profile ditu (Nork eginda, nori zuzenduta, nork prozesatua) eta RideRequest 1 (Bidaia eta Traveller lotzen dituen klasea delako eta beraz klase egokiena honetarako).

- **Mezua:** Azkenik mezua klasea dugu, klaserik konplikatuena, Mezu bat Profile batentzako izango da, beraz, Mezu batek 1 Profile ditu. Gauza da, Mezua klasearen barruan 3 mezu ezberdin daude: MezuaTransaction, MezuaAlerta, MezuaErreklamazioa. Hiru hauek atributu ezberdinak behar dituzte, beraz nahiz eta mezuak izan 3-ak guztiz ezberdinak dira, eta bakoitza azalduko dut.
 - 1- MezuaTransaction: Hau, dirua mugitzen denean sortzen da(erreserbatzean, dirua sartu, dirua atara,...). Transakzio batek, 0..1 (0 edo 1)RideRequest izan ditzake, eta 0 Alerta,0 Erreklamazioa eta 0 Ride ditu.
 - 2- MezuaAlerta: MezuaAlertaren kasuan, hau sortu egiten da, Traveller batek sortutako Alerten baldintzak betetzen dituen Ride bat sortzen denean. Beraz, mezu honek 1 Alerta eta 1 Ride izango ditu(Adierazteko zein alerta betetzen duen zein Ride-k). Bestalde , 0 RideRequest eta 0 Erreklamazio ditu.
 - 3- MezuaErreklamazio: MezuaErreklamazio erreklamazio bat sortzean edo aldaketak egitean sortzen da, ez da mugitzen den dirua agertzen, baizik eta zein erreklamazio eginda dauden eta hauen egoerak(onartuta, deuseztatua). MezuaErreklamazioak, erreklamazioei lotuta dagoenez, 1 Erreklamazio du, adierazteko zein erreklamaziori dagokion. Bestalde 0 Ride, 0 RideRequest eta 0 Alerta ditu.
- **Enum klaseak:** Azkenik, jarritako domeinu ereduan agertzen ez diren 3 klase jarri ditugu, ez baitgenekigun nola jarri enum klase bat Domeinu ereduan. Klase hauek balio dute “egoera” atributuak hobe kontrolatzeko, atributu hauek dituzten klaseak hauek dira: Ride, Erreklamazioa, RideRequest. Beraz, Ride-ek 1 **Egoera Ride.class** du, RideRequest-ek 1 **Egoera RideRequest.class**, eta Erreklamazioek 1 **Egoera Erreklamazio.class** du.

Erabilpen kasuen eredua:



[linka](#)

Gertaera Fluxuak eta GUI-ak:

1- QueryRides Gui:

The screenshot shows the QueryRides GUI. It has two dropdown menus for 'Depart City' (selected 'a') and 'Arrival City' (selected 'b'). To the right is a calendar for May 2025. The date May 19 is highlighted in blue. Below the calendar, a table titled 'Viajes: May 19, 2025' displays ride information. The table has columns for Driver, Seats, Price, and rating. The first row shows a ride by 'Ekaitz/Ekaitz' with 1 seat, a price of 4.0, and a rating of 3.6. A 'Close' button is at the bottom.

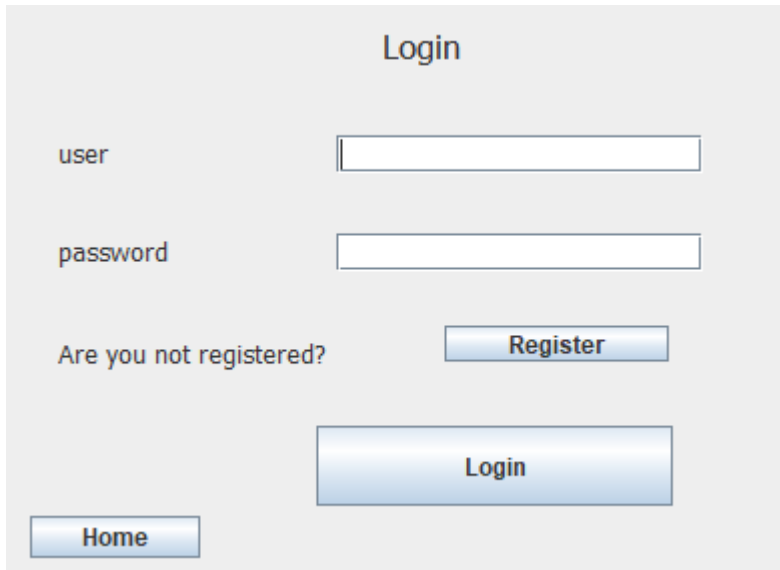
Driver	Seats	Price	rating
Ekaitz/Ekaitz	1	4.0	3.6

Query Rides Flow of events

Basic Flow

1. System displays all **cities** where **rides** depart from
2. user selects a departing **city**
3. System displays all destination **cities** for **rides** that depart from a **selected city**.
4. user selects an arrival **city**
5. System highlights in a Calendar the days where **rides** exist from the **depart** to destination **cities** in a selected month
6. user selects a **date** in a Calendar
7. System displays the **rides** from the selected **departin city** to the selected **arrival city** on that **date**.

2- Login



The image shows a login form with a light gray background. At the top center is the title "Login". Below it are two input fields: the first is labeled "user" and the second is labeled "password". To the right of the "password" field is a "Register" button. Below the "password" field is a "Login" button. At the bottom left is a "Home" button. The text "Are you not registered?" is positioned to the left of the "Register" button.

Login-en gertaera fluxua

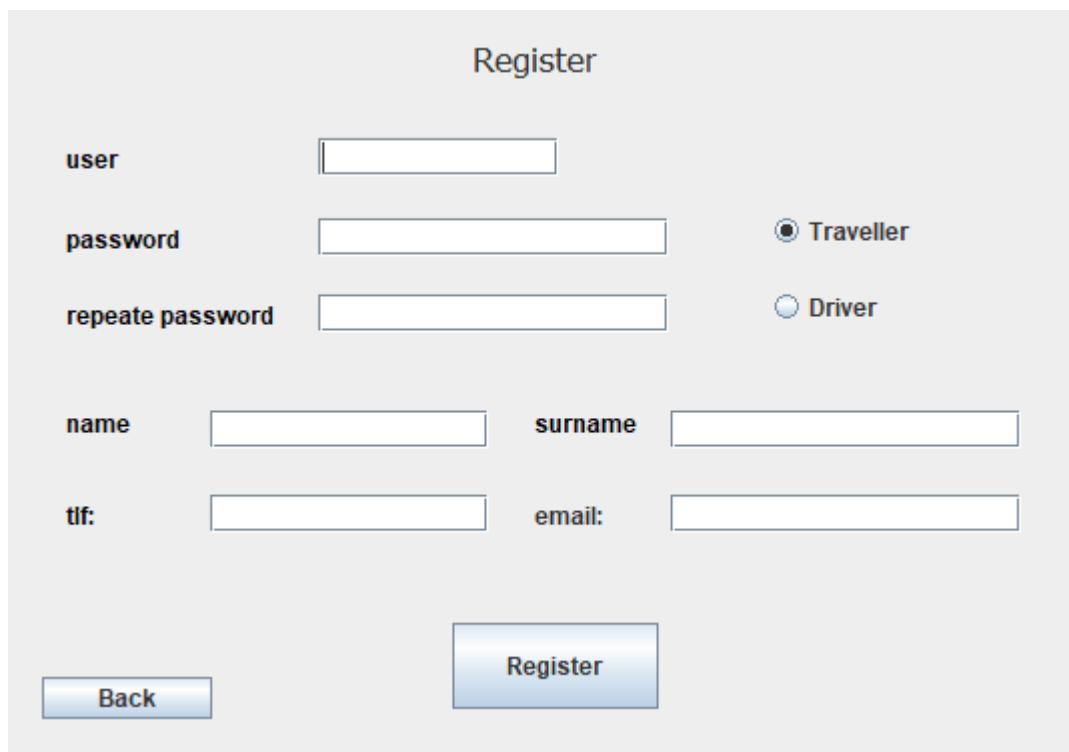
Oinarrizko fluxua

1. *Erabiltzaileak* dagozkion tauletan bere **usuario eta kontraseina** jarri eta ea **gidari edo bidaiari** den.
2. *Sistemak* datuak egiaztatu eta home kargatu egiten du bere kontuarekin.

Fluxu alternatiboa

1. Erabiltzaile edo kontraseina ez dago ondo. Sistemak erabiltzaileari jakinaraziko dio zein dagoen gaizki.
2. Ez du gidaria edo bidaiari den botoia sakatu, Login botoia ez da aktibatuko aukera bat sakatu arte.

3- Register



The image shows a web form titled "Register". It contains several input fields and two radio buttons. The fields are labeled "user", "password", "repeate password", "name", "surname", "tlf:", and "email:". The radio buttons are labeled "Traveller" and "Driver". At the bottom, there are two buttons: "Back" and "Register".

Register

user

password ☒ Traveller

repeate password ☐ Driver

name surname

tlf: email:

Register-en gertaera fluxua

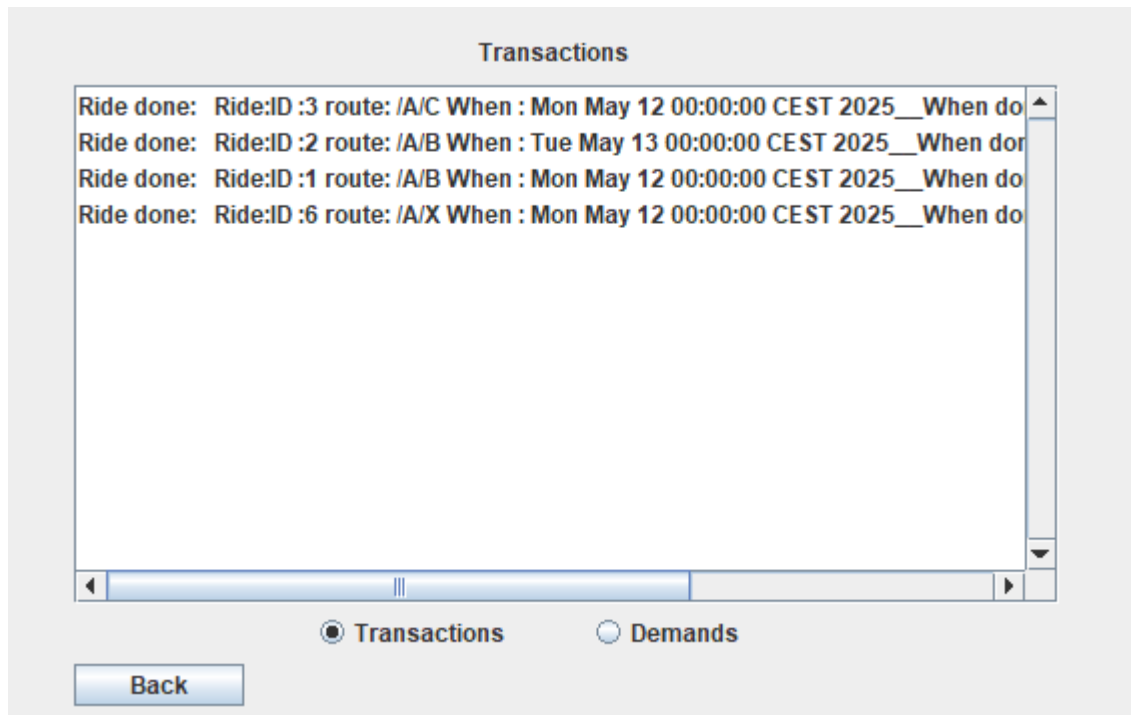
Oinarrizko fluxua

1. *Sistemak* datuak eskatuko dizko erabiltzaileari(usuaria, kontraseina, email-a, izena, gidaria edo bidaiaria den ...).
2. *Erabiltzaileak* datuak beteko ditu
3. *Sistemak* datuak ondo daudela egiaztatu eta informazioa gorde egiten du.
4. *Sistemak* automatikoki egiten dio login erabiltzaileari sartutako datuekin.

Fluxu alternatiboa

1. **Datuak falta** dira. Sitemak hau jakinaraziko dio erabiltzaileari eta ezingo du login egin datuak kargatu arte.
2. **Datuak gaizki** kargatu dira. Sistemak arazoa non dagoen jakinaraziko du eta register botoia desaktibatuta egongo da konpondu arte.
3. Kontraseina sartzean, **sartutako bi kontraseinak ez dira berdinak**. Sistemak hau jakinaraziko du eta ezingo da register egin.
4. Aukeratutako **usuaria edo email-a jadanik existitzen da**. Sistemak hau jakinaraziko dio erabiltzaileari.
5. **Ez du aukeratu bidaiari edo gidaria den**. Sistemak eskatu egingo du arlo hau betetzeko, bestela ezingo da register egin.

4- MezuakIkusi

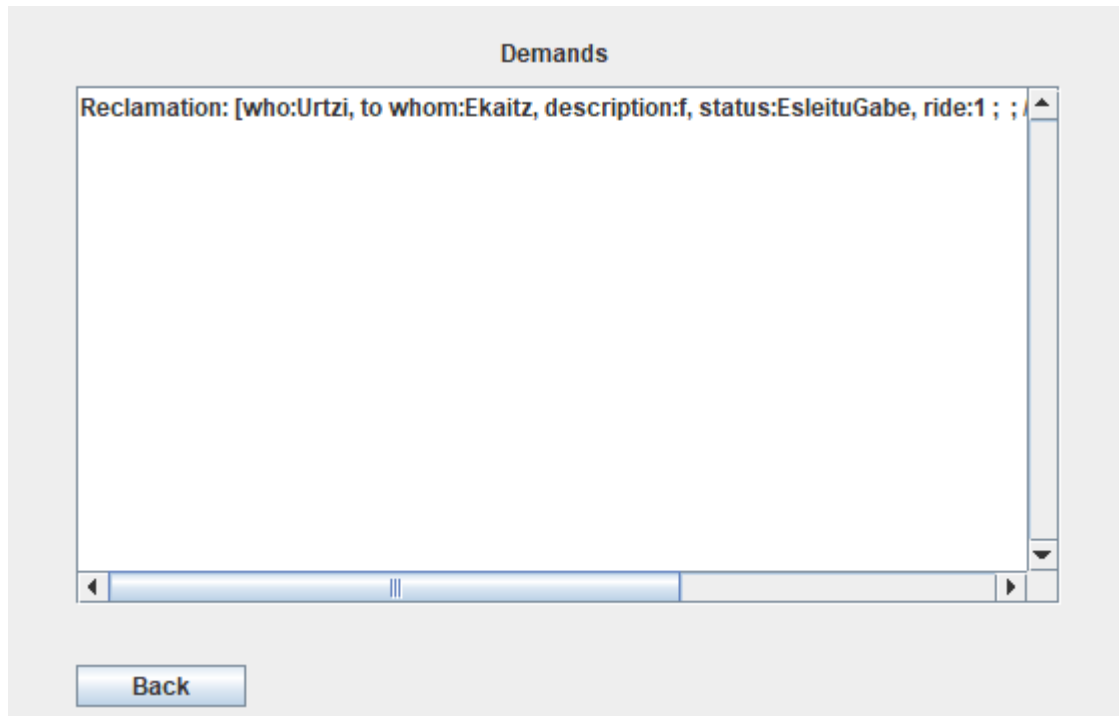


Mezuak ikusi-ren gertaera fluxua

Oinarrizko fluxua

1. **Erabiltzaileak** transactions botoia sakatu
2. ****Sistemak **** dirua mugitzean sortzen diren mezuak erakutsiko dizkio **erregistratutako erabiltzaileari.**
3. **Erabiltzaileak** demands botoia sakatu
4. **Sistemak** Jadanik ebatzitako(onartutako/deuseztatutako) erreklamazioak erakutsiko ditu.

5- Erreklamazioak Ikusi



The screenshot shows a window titled "Demands". Inside the window, there is a text area containing the following text: "Reclamation: [who:Urtzi, to whom:Ekaitz, description:f, status:EsleituGabe, ride:1 ; ;]". Below the text area, there is a horizontal scrollbar. At the bottom of the window, there is a button labeled "Back".

Erreklamazioak ikusi-ren gertaera fluxua:

Oinarrizko fluxua

1. *Sistemak * egindako erreklamazioak erakutsiko dizkio *erregistratutako erabiltzaileari*.

6- ErreserbaEskatu

Depart City

Arrival City

How many seats:

Ride date

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

Viajes: May 19, 2025

Driver	Seats	Price	Rating
Ekaitz/Ekaitz	1	4.0	3.6

Create alert

Close

Erreserbatu

Erreserba eskatu-ren gertaera fluxua

Oinarrizko fluxua

1. *Bidaiariak* nahi duen **datan** eta nahi duen **ibilbidean** bidaia bat aukeratzen du eta **erreserba** botoia sakatu egiten du(Login eginda badauka bidaiari moduan)
2. *Sistemak* jakinaraziko dio zenbat kobratuko dion.
3. *Bidaiariak* onartuko du kantitate hori kobratzea.
4. *Sistemak* jakinaraziko dio erreserba gauzatu dela eta gidariari bidaliko zaio eskaria, mezu bat gehituko zaio *gidariari* gastatutako dirua adieraziz.

Fluxu alternatiboa

1. *Bidaiariak* nahi duen **bidaia** ez da existitzen. Bidaia posibleen lista hutsik agertzen da, eta **erreserba** botoia desaktibatuta.
2. *Bidaiariak* ez du diru nahikorik **wallet-ean**. Erreserba deuseztatu egingo da eta eskatuko zaio diru gehiago sartzea.
3. Erabiltzaileak ez badu login egin **bidaiari** moduan ez da erreserba botoia agertuko.

7- AlertakSortu

From:

To:

May 2025

	Sun	M...	Tue	W...	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

Alertak sortu-ren gertaera fluxua

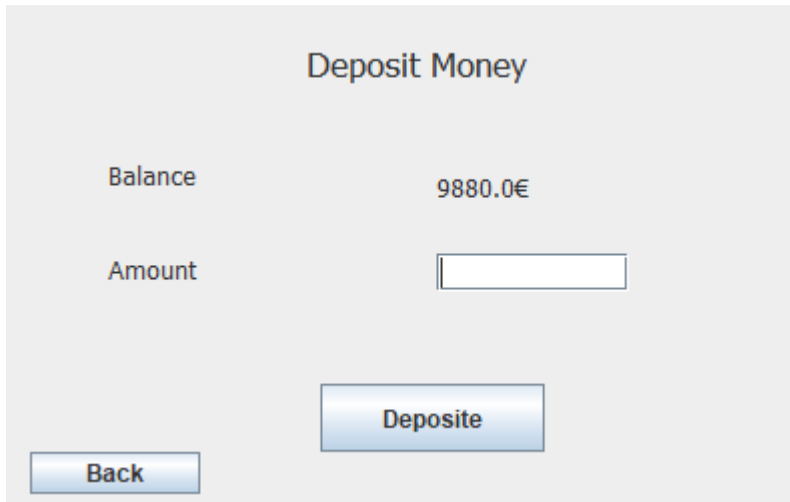
Oinarrizko fluxua

1. **Traveller**-ak "create alert" botoia sakatuko du "erreserbatuGUI-an".
2. ****Sistemak** Traveller**-ari eskatuko dio Bidaia **nondik, nora** eta **noiz** izatea nahi den sartzeko.
3. **Traveller**-ak informazio hori bete eta **Create Alert** botoia sakatzen du.
4. **Sistemak** Alerta sortu egingo du, **Traveller**-ari Alerta esleitu eta aldaketak gordeko ditu.

Fluxu Alternatiboa:

5. **Alerta jadanik sortua bada, Sistemak** jakinaraziko dio **Traveller**-ari.
6. ****Bidaia**ren bat existitzen bada guk sortu nahi dugun alertaren ezaugarriek

8- DiruaSartu



The image shows a user interface for depositing money. At the top, the title "Deposit Money" is centered. Below it, the "Balance" is displayed as "9880.0€". Underneath, the "Amount" label is followed by an empty rectangular input field. At the bottom, there are two buttons: a "Back" button on the left and a "Deposit" button on the right. The entire interface is set against a light gray background.

Dirua sartu-ren gertaera fluxua

Oinarrizko fluxua

1. *Sistemak* eskatu egiten dio bidaiariari zenbat diru sartu nahi duen.
2. *Bidaiariak* nahi duen diru kantitatea jarri egin eta **deposit** botoia sakatu egiten du.
3. *Sistemak* datuak egiaztatu, eta bidaiariaren **wallet-a** eguneratu , mezu bat jarri egiten du dena ondo joan dela jakinaraziz eta mezu bat gehituko zaio **traveller**-eri sartutako dirua adieraziz.

Fluxu alternatiboa

1. Diru kantitatearen lekuan letrak edo zenbaki negatiboak sartu dira. Sistemak jakinarazi egingo du eta ezingo da **deposit** botoia sakatu.

9- IkusiErreserbak

Reservations

user Urtzi RideRequest 30 Seats1

From : a

To : b

When : Tue May 20 00:00:00 CEST 2025

Price 3.0

Driver a a user: Ekaitz

When asked: Sat May 17 18:39:04 CEST 2025

When Answered:

Seats requested: 1

☐ Rejected

☒ Unhandled

☐ Accepted

☐ Passed

Back

Erreserbak ikusi-ren gertaera fluxua

Oinarrizko fluxua

1. Bidaiariak aukeratuko du ze erreserba mota ikusi nahi dituen: "Onartuak", "Deuseztatuak", "Tratatu Gabeak", "Pasatuak"
2. Sistemak Bidaiariari aukeratutako egoeran dauden erreserbak erakutsiko dizkio.
3. Bidaiariak erreserba bati-klik egitean sistemak erreserbaren informazio garrantzitsua emango dio.

10-BidaiaEginDelaOnartu

Reservations

From :
To :
When :
Price
Driver
When asked:
When Answered:
Seats requested:

☐ Rejected ☐ Unhandled ☐ Accepted ☒ Passed

Yes
No

Back

Bidaia Egin dela Onartu ikusi-ren gertaera fluxua

Oinarrizko fluxua

1. *Bidaiariak* ikusi egiten ditu erreserbak eta "pasatutako bidaiak" botoia sakatuko du.
2. *Sistemak bidaiari*-ari erakutsiko dio, onartuta dituen erreserbak eta bidaiaren data pasa direnak.
3. *Bidaiariak* aukeratuko du bidaia ondo egin dela.
4. *Sistemak* bidaia berdinean dagoen **Bidaiari** berdinari dagozkion erreserba guztiak onartuko ditu eta dagokion gidariari bidaiaren dirua gehituko dio eta mezu bat gordeko zaio bidaiaren gidariari adieraziz dirua depositatu zaiola. Erreserbaren egoera: "Eginda"(Erreserba bakoitzeko).

11-BidaiaEginDelaDeuseztatu

Reservations

From :
To :
When :
Price
Driver
When asked:
When Answered:
Seats requested:

☐ Rejected ☐ Unhandled ☐ Accepted ☒ Passed

Yes
No

Back

Bidaia Egin dela Deuseztatu ikusi-ren gertaera fluxua

Oinarrizko fluxua

1. *Bidaiariak* ikusi egiten ditu erreserbak eta "pasatuta" botoia sakatuko du.
2. *Sistemak bidaiari*-ari erakutsiko dio, onartuta dituen erreserbak eta bidaiaren data pasa direnak.
3. *Bidaiariak* aukeratuko du bidaia gaizki egin dela.
4. *Sistemak* bidaia berean dauden **bidaiariaren** erreserba guztiak not done egoeran jarriko ditu, Erreserba bakoitzeko bidaiariari dirua itzuliko dio eta mezu bat gordeko da **Traveller**-en adieraziz zenbat diru itzuli zaion. Erreserbaren egoera, "Ez Egina"(Erreserba bakoitezko)

12-BidaiakIkusiTraveller

Reservations

1 ; ; route:/A/B ; Mon May 12 00:00:00 CEST 2025

2 ; ; route:/A/B ; Tue May 13 00:00:00 CEST 2025

3 ; ; route:/A/C ; Mon May 12 00:00:00 CEST 2025

4 ; ; route:/A/C ; Tue May 13 00:00:00 CEST 2025

6 ; ; route:/A/X ; Mon May 12 00:00:00 CEST 2025

5 ; ; route:/A/X ; Tue May 13 00:00:00 CEST 2025

From : A

To : C

When : Tue May 13 00:00:00 CEST 2025

Driver Ekaitz Pinedo

Route A/C

☐ Passed

☐ Cancelled

☐ not Done

☐ Not Passed

☒ Done

Rate

Claim

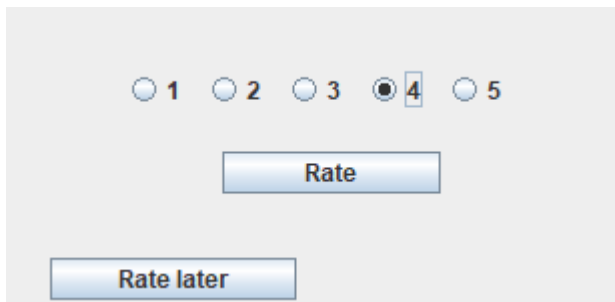
Back

#BidaiakIkusiDriver ikusi-ren gertaera fluxua

Oinarrizko fluxua

1. Erabiltzailea **Driver denean: Sistemak** *Driver*-ak egindako bidaiak erakusten ditu
2. Erabiltzaileak **not passed** botoia sakatu
3. **Sistemak** oraindik martxan daduen bidaiak erakusten ditu
4. Erabiltzaileak **Passed** botoia sakatu
5. **Sistemak** pasatuta diren eta Eginda edo ez eginda oraindik aukeratuta ez daduen bidaiak erakusten ditu
6. Erabiltzaileak **Cancelled** botoia sakatu
7. **Sistemak** kanzelatuak izan diren bidaiak erakusten ditu
8. Erabiltzaileak **not Done** botoia sakatu
9. **Sistemak** pasatuta dauden eta ez eginda bezala jarri diren bidaiak erakusten ditu
10. Erabiltzaileak **Done** botoia sakatu
11. **Sistemak** pasatuta dauden eta eginda bezala jarri diren bidaiak erakusten ditu

13-Baloratu

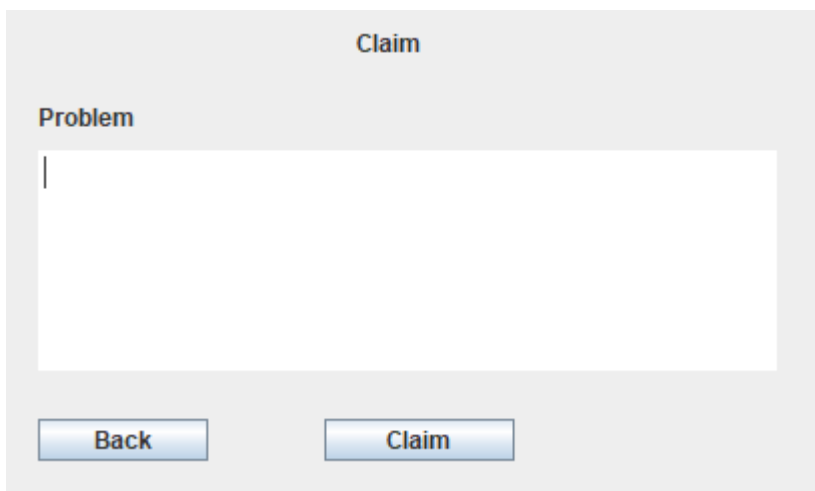


#Baloratu-ren gertaera fluxua

Oinarrizko fluxua

1. ****Sistemak **** erabiltzaileari eskatu 1-etik 5-erako balorazio bat aukeratzea.
2. **Erabiltzaileak** balorazio bat jarri
3. **Sistemak** balorazioa gordeko dio tokatzen zaion pertsonari.

14-Erreklamatu



#Erreklamatu-ren gertaera fluxua

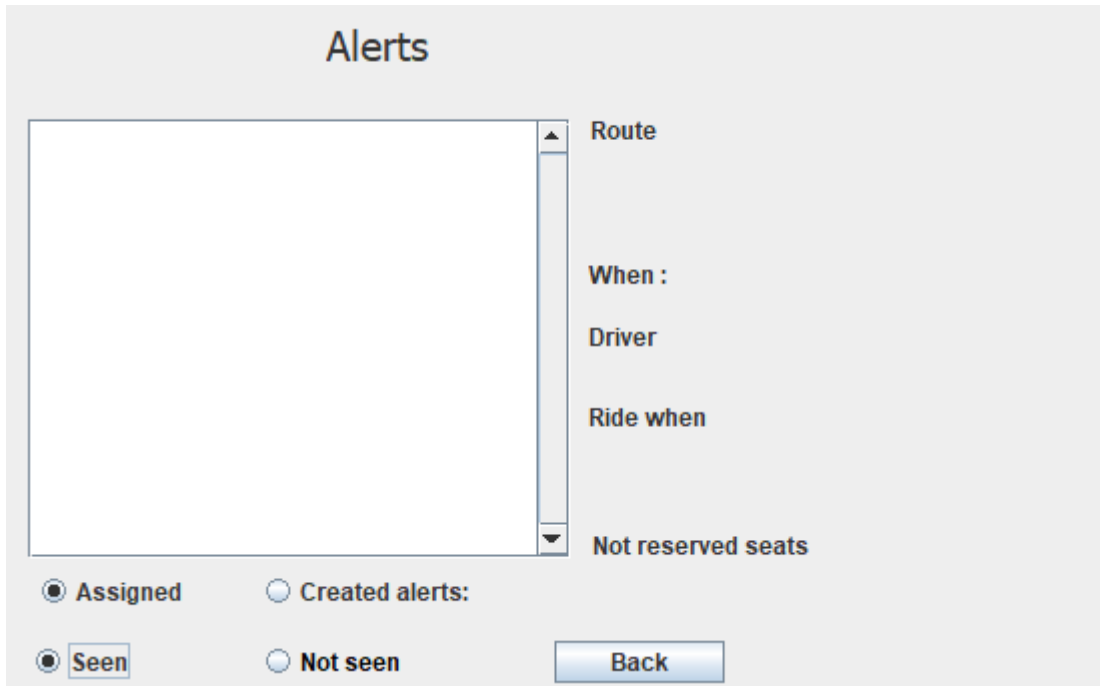
Oinarrizko fluxua

1. ****Sistemak **** erabiltzaileari eskatu erreklamazio bat idazteko.
2. **Erabiltzaileak** erreklamazio bat idatzi
3. **Sistemak** erreklamazio bat sortuko du, zehaztuz nork egin duen, nori egin dion eta mezua, eta informazioa gorde, eta erabiltzaileari esleitu Erreklamazioa.

##Fluxu alternatiboa

1. Erabiltzaileak Ez badu ezer idazten eta bidali egiten badu erreklamazioa, sistemak jakinaraziko dio zerbait idatzi behar duela, eta ez da erreklamazioa sortuko

15-AlertakIkusi



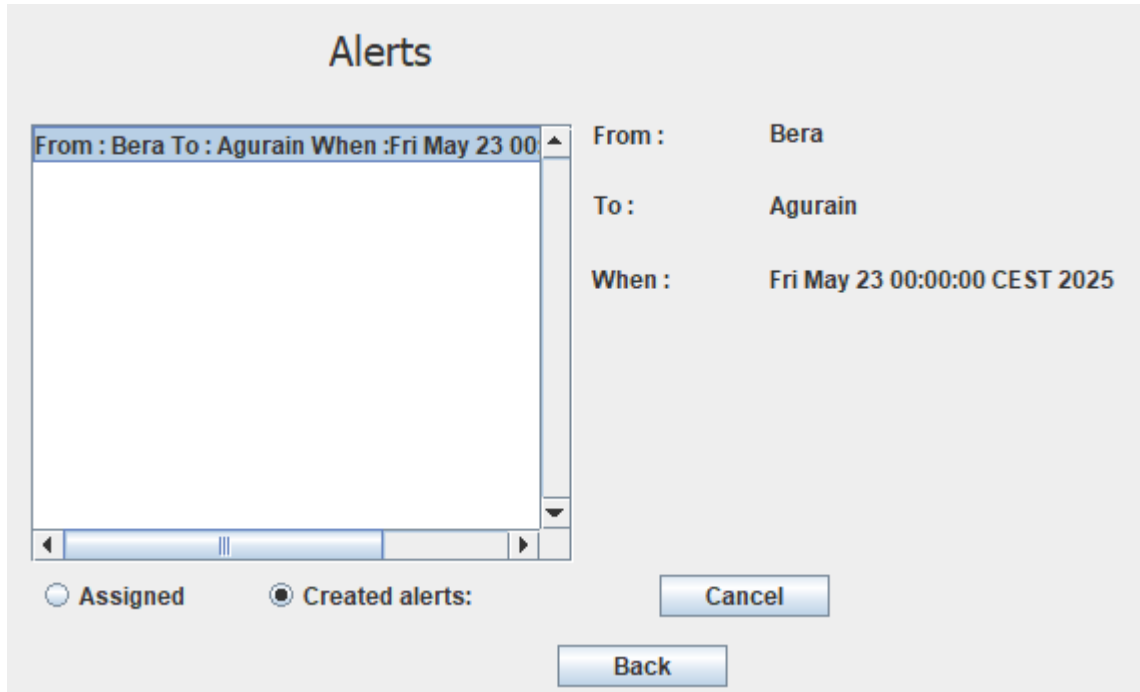
The image shows a web interface titled "Alerts". It features a large empty rectangular box on the left, likely for displaying a list of alerts. To the right of this box is a vertical scrollbar. Further to the right, there are several labels: "Route", "When :", "Driver", "Ride when", and "Not reserved seats". Below the main box, there are four radio buttons: "Assigned" (selected), "Created alerts:", "Seen" (selected), and "Not seen". A "Back" button is located at the bottom right of the interface.

#Alertak ikusi-ren gertaera fluxua

Oinarrizko fluxua

1. **Traveller**-ak "sortutako alertak" botoia sakatu.
2. ***Sistemak *** sortutako alertak erakutsiko dizkio **erregistratutako erabiltzaileari**.
3. **Traveller**-ek "irakurri gabeko mezuak" botoia sakatu.
4. **Sistemak traveller**-ek irakurri gabe dituen alerta mezuak erakutsiko ditu.
5. **Traveller**-ek "irakurritako alerta mezuak" botoia sakatu.
6. **Sistemak traveller**-ak irakurrita dituen alerta mezuak erakutsiko ditu.

16-Alerta Ezabatu



The image shows a software window titled "Alerts". On the left is a list box containing one entry: "From : Bera To : Agurain When : Fri May 23 00". Below the list box are two radio buttons: "Assigned" (unselected) and "Created alerts:" (selected). To the right of the list box, the details for the selected alert are displayed: "From : Bera", "To : Agurain", and "When : Fri May 23 00:00:00 CEST 2025". At the bottom right are two buttons: "Cancel" and "Back".

#Alerta ezabatu ikusi-ren gertaera fluxua

Oinarrizko fluxua

1. *Traveller*-ak sortutako alerta bat aukeratu eta *cancel* botoia sakatu.
2. *Sistemak* "ezabatu" egingo du alerta hori, hau da, ez da gehiago agertuko *sortutako alertak* botoia sakatzean.

17-CreateRide

Depart City Bera

Arrival Ci... Toledo

Price 5

Select a car : Seat Toledo 932..

Number of Seats 4

Ride date

	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

Ride created

Are there more stops?

☐ Yes ☒ No

Create Ride **Close**

Create Ride Flow of events

Basic Flow

1. *Driver*-ak aukeratu egiten du ez duela geltoki anitzik nahi: Aukeratu " geltotikiRik ez" button.
2. *Driver* insert the **departing city**, **arrival city**, **number of seats** and **price** and **car**
3. *Driver* selects a **ride date**
4. *System* creates a **ride** with inserted data and assigns to the **Driver**.

#####

Create Ride (Bidaia Anitza Mode)

Depart City

Bera

Arrival City

Agurain

In this format: city1/city2/...

Added cities:

Tolosa

Edit

Save cities

Tolosa-->Agurain

Added prices:4.0/7.0

Price

7

Put price

Select a car :

a a a

Number of Seats

4

Are there more stops?

☒ Yes

☐ No

Create Ride

Close

Ride date

May

2025

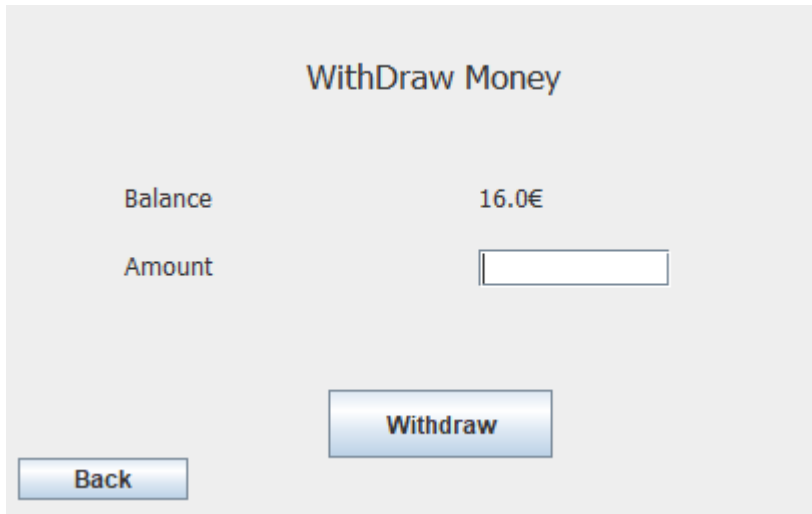
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18					1	2	3
19	4	5	6	7	8	9	10
20	11	12	13	14	15	16	17
21	18	19	20	21	22	23	24
22	25	26	27	28	29	30	31

5. *Driver*-ak aukeratu egiten du nahi dituela geltoki anitzak: Aukeratu " geltotikiak bai" button.
6. *Driver*-ak sartu egiten ditu **Depart City**, **Arrival City**, **Geltokiak**,**prezioak**, **kotxea** eta **eserlekuak**
7. *Driver*-ak aukeratu **data** bat
8. *System* creates a **ride** with inserted data and assigns to the **Driver**.

Alternative flow

1. Some of fields are empty. Final.
2. **number of seat** or **price** is not a number. System informs the user.
3. **Ride date** is before today. System informs the user.
4. the **ride** already exist for this **driver**. Ride is not created.System informs the user.
5. Bidaia anitzak direnean,** prezioakKop!= geltokiGuztiak** bada(hau da, prezio kopurua desegokia bada), error. **Sistemak** erabiltzailea jakinarazi.
6. Hiriren bat behin baina gehiagotan errepikatzen bada, error, Sistemak erabiltzaileari jakinarazi.

18-DiruaAtera



The image shows a web form titled "WithDraw Money". It has a light gray background. At the top, the title "WithDraw Money" is centered. Below the title, there are two labels: "Balance" and "Amount". The "Balance" label is followed by the text "16.0€". The "Amount" label is followed by a text input field. At the bottom of the form, there are two buttons: "Back" on the left and "Withdraw" on the right. The "Withdraw" button is slightly larger and has a blue gradient.

WithDraw Money	
Balance	16.0€
Amount	<input type="text"/>
<input type="button" value="Back"/>	<input type="button" value="Withdraw"/>

Dirua atera-ren gertaera fluxua

Oinarrizko fluxua

1. *Sistemak* eskatu egiten dio bidaiariari zenbat diru atera nahi duen.
2. *Gidariarak* nahi duen diru kantitatea jarri egin eta **withdraw** botoia sakatu egiten du.
3. *Sistemak* datuak egiaztatu, eta gidariaren **wallet-a** eguneratu eta mezu bat gorde egiten dio gidariari ateratako dirua adieraziz.

19- Erreserba Onartu

Travels	106 ; ; route:/a/f ; Tue May 20 00:00:00 CEST 2025 ▼		
Requests	user Urtzi RideRequest 111 Seats1 ▼		
Who requested:	Urtzi Etxegarai	Rating	3.00 ratings: 1
When requested:	Wed May 14 01:08:33 CEST 2025	Reserves:	2
Request price:	4.0	Seats requested:	1
		Seats available:	1
From requested:	a	To Requested:	f
		When :	Tue May 20 00:00:00 CEST 2025
<input type="button" value="Back"/>		<input type="button" value="Accept"/>	
		<input type="button" value="Reject"/>	

Erreserba onartu-ren gertaera fluxua

Oinarrizko fluxua

1. *Sistemak* gidariak dituen bidaiak aurkezten ditu.
2. *Gidariak* **bidaiak** bat aukeratu egiten egiten du.
3. *Sistemak* bidaiak horretan eskatutako **erreserbak** erakusten ditu.
4. *Gidariak* **erreserba eskakizun** bat aukeratu egiten du.
5. *Sistemak* **eskakizun** hori nork eta noiz eginda dagoen erakutsiko du eta eskakizunaren informazioa.
6. *Gidariak* **accept** botoia sakatuko du.
7. *Sistemak* onartutako erreserba-ren bidaiariari mezu bat bidaliko dio , gidariari ere. Geratzen diren eserlekuak baina gehiago dituzten erreserbak automatikoki deuseztatu eta dirua itzuli

Fluxu alternatiboa

1. Ez dago erreserba eskaririk. Botoiak desaktibatuko dira.
2. *Gidariak* ez du bidaiarik, hutsa agertuko da lista.

20-ErreserbaDeuseztatu

Travels	106 ; ; route:/a/f ; Tue May 20 00:00:00 CEST 2025 ▼		
Requests	user Urtzi RideRequest 111 Seats1 ▼		
Who requested:	Urtzi Etxegarai	Rating	3.00 ratings: 1
When requested:	Wed May 14 01:08:33 CEST 2025	Reserves:	2
Request price:	4.0	Seats requested:	1
		Seats available:	1
From requested:	a	To Requested:	f
		When :	Tue May 20 00:00:00 CEST 2025
<input type="button" value="Back"/>		<input type="button" value="Accept"/>	
		<input type="button" value="Reject"/>	

Erreserba deuseztatu-ren gertaera fluxua

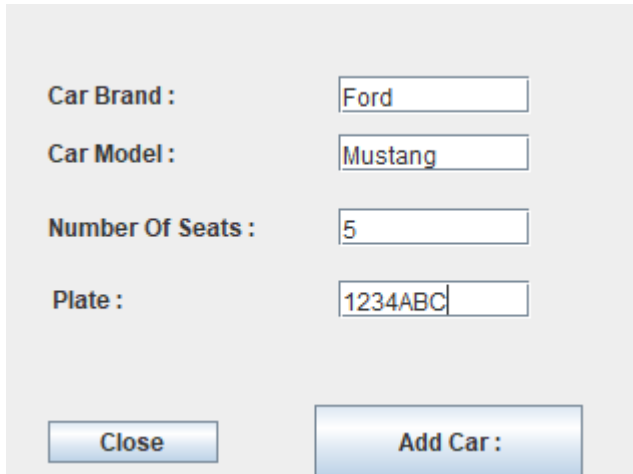
Oinarrizko fluxua

1. *Sistemak* gidariak dituen bidaiak aurkezten ditu.
2. *Gidariak* **bidaia** bat aukeratu egiten egiten du.
3. *Sistemak* bidaia horretan **eskatutako erreserbak** erakusten ditu.
4. *Gidariak* **erreserba eskakizun** bat aukeratu egiten du.
5. *Sistemak* **eskakizun** hori nork eta noiz eginda dagoen erakutsiko du eta eskakizunaren informazioa.
6. *Gidariak* **reject** botoia sakatuko du.
7. *Sistemak* deuseztatutako erreserba-ren bidaiariari mezu bat bidaliko dio eta dirua itzuliko zaio.

Fluxu alternatiboa

1. Ez dago erreserba eskaririk. Botoiak desaktibatuko dira.
2. *Gidariak* ez du bidaiarik, hutsa agertuko da lista

21-KotxeaSortu



Erreserba kotxea sortu-ren gertaera fluxua

Oinarrizko fluxua

1. *Sistemak* gidariari kotxearen matrikula, marka, modeloa eta eserleku kopurua jartzeko eskatuko dio.
2. *Gidariak* matrikula, marka, modeloa eta eserleku kopurua jarriko ditu.
3. *Sistemak* datu hauek egiaztatu, eta kotxea gordeko du gidariari atxikituz.

Fluxu alternatiboa

1. Eserleku kopuruaren tokian letrak jartzen ditu *Gidariak*. *Sistemak* eskatuko dio datu onekin betetzea eserleku kopurua.
2. Eserleku kopuruaren tokian zenbaki negatibo bat jartzen ditu *Gidariak*. *Sistemak* eskatuko dio datu onekin betetzea eserleku kopurua.
3. Eserleku kopuruaren tokian 9 eserleku baina gehiago edo jartzen ditu *Gidariak*. *Sistemak* eskatuko dio datu onekin betetzea eserleku kopurua.
4. Daturenbat ez badago beteta *Sistemak* eskatuko du datu hau betetzea.
5. Kotxea jadanik sortuta badago, ez da berriz ere sortuko.

Reservations

<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> 1 ; ; route:/A/B ; Mon May 12 00:00:00 CEST 20 2 ; ; route:/A/B ; Tue May 13 00:00:00 CEST 20 3 ; ; route:/A/C ; Mon May 12 00:00:00 CEST 20 4 ; ; route:/A/C ; Tue May 13 00:00:00 CEST 20 5 ; ; route:/A/X ; Tue May 13 00:00:00 CEST 20 6 ; ; route:/A/X ; Mon May 12 00:00:00 CEST 20 </div> <div style="border: 1px solid black; height: 20px; width: 100%;"></div>	From : A To : X When : Tue May 13 00:00:00 CEST 2025 Driver Ekaitz Pinedo Requests: 4 Available seats 3 Route A/X <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Urtzi Urtzi Etxegarai</div>
--	---

☐ Passed ☐ Cancelled
☐ not Done ☐ Not Passed
☒ Done

Urtzi Urtzi Etxegarai

Rate

Claim

Back

BidaiaIkusiDriver ikusi-ren gertaera fluxua

Oinarrizko fluxua

1. Erabiltzailea **Driver** denean: **Sistemak** *Driver*-ak egindako bidaiak erakusten ditu
2. Erabiltzaileak **not passed** botoia sakatu
3. **Sistemak** oraindik martxan daduen bidaiak erakusten ditu
4. Erabiltzaileak **Passed** botoia sakatu
5. **Sistemak** pasatuta diren eta Eginda edo ez eginda oraindik aukeratuta ez daduen bidaiak erakusten ditu
6. Erabiltzaileak **Cancelled** botoia sakatu
7. **Sistemak** kanzelatuak izan diren bidaiak erakusten ditu
8. Erabiltzaileak **not Done** botoia sakatu
9. **Sistemak** pasatuta dauden eta ez eginda bezala jarri diren bidaiak erakusten ditu
10. Erabiltzaileak **Done** botoia sakatu
11. **Sistemak** pasatuta dauden eta eginda bezala jarri diren bidaiak erakusten ditu

23-BidaiaKanzelatu

Reservations

59 ; ; route:/bERA/LESAKA ; Mon May 19 00:00:00 CEST 2025

72 ; ; route:/a/b ; Mon May 19 00:00:00 CEST 2025

73 ; ; route:/a/b ; Tue May 27 00:00:00 CEST 2025

106 ; ; route:/a/f ; Tue May 20 00:00:00 CEST 2025

From : a

To : b

When : Tue May 27 00:00:00 CEST 2025

Driver Ekaitz Pinedo

Requests: 7

Available seats 4

Route a/b

☐ Passed

☐ Cancelled

☐ not Done

☒ Not Passed

☐ Done

Cancel

Back

BidaiaKanzelatu-ren gertaera fluxua

Oinarrizko fluxua

1. *Sistemak * gidariari erakutsi egingo dizkio berak dituen bidaia guztiak eta oraindik martxan daudenak("Not passed" botoia sakatzean).
2. *Gidariak* bidaia bat aukeratuko du kanzelatzeko eta kanzelatuko du.
3. *Sistemak* bidaia hori "kanzelatu" egoeran jarriko du, onartu gabe edo onartuta zeuden erreserba guztiak deuseztatu eta dirua itzuliko dio bidaiarieri

24-Erreklamazioak Kudeatu

Reclamation: [who:Ekaitz, ...]

Problem

a

Claimant Ekaitz Pinedo

Defendant Ekaitz Pinedo

From : A

To : X

When : Mon May 12 00:00:00 CEST 2025

Price 4.0€

Amount 7

☒ Accept ☐ Reject

Back Take new Finish

#Erreklamazioak kudeatu-ren gertaera fluxua

Oinarrizko fluxua

1. *Sistemak * **Admin**-ari erakutsiko dizkio, adminak esleituta eta bukatu gabe dituen erreklamazioak.
2. **Admin**-ak horietako bat aukeratu edo **take new** botoia sakatuko du beste erreklamazio berri bat nahi badu.
3. **Sistemak admin**-ak take new botoia sakatu badu, esleitu gabeko erreklamazio zaharrena esleituko dio **Adminari**.
4. **Adminak** erreklamazio bat aukeratu eta onartu edo deuseztatu egingo duen aukeratu, baita ere zenbat diru itzuliko duen(itzuli nahi badu).
5. **Sistemak** aldaketak gorde eta mezu bat bidaliko die parte hartu dutenei transakzioak egon badira.

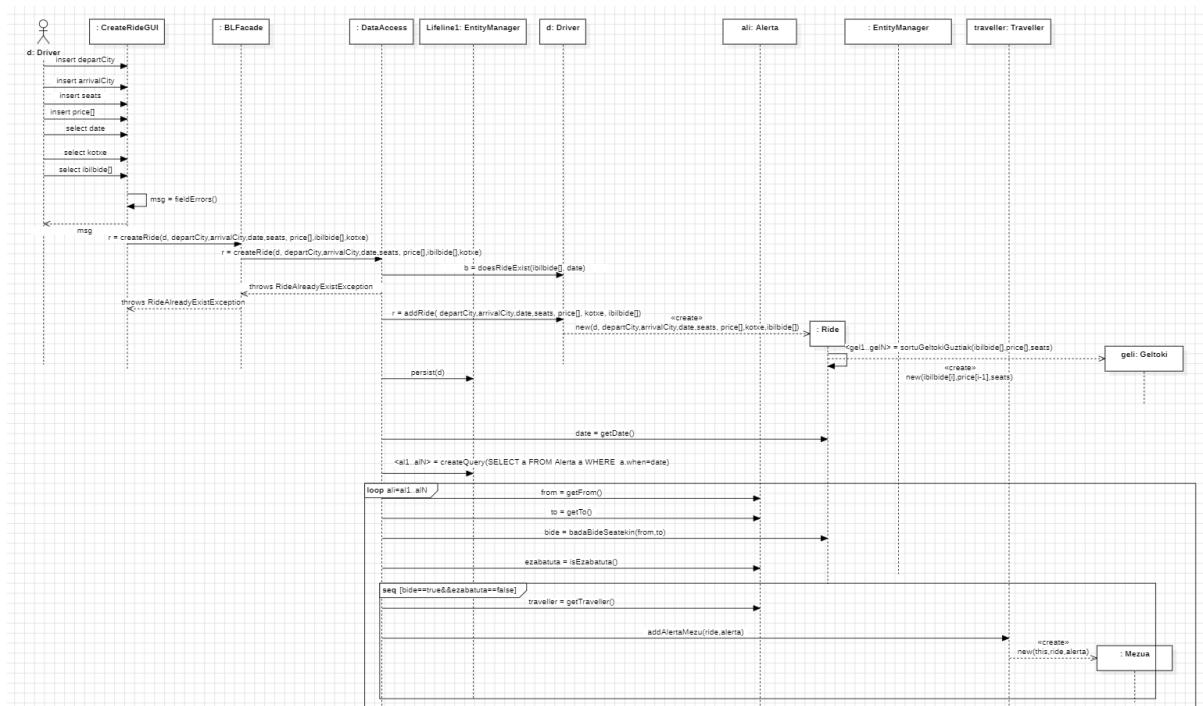
Fluxu alternatiboak

6. **Adminak** ez du erreklamaziorik aukeratu, jakinaraziko zaio hori **sistemaren** bidez.
7. Ez dago esleitu gabeko erreklamaziorik take new sakatzean. **Sistemak** hori jakinarazi adminari.

Diseinua

Sekuentzia diagramak eta azalpenak (Hobeto ikusteko erabili link-a, link-ak erabilpen kasuen izenak dira):

Create Ride:



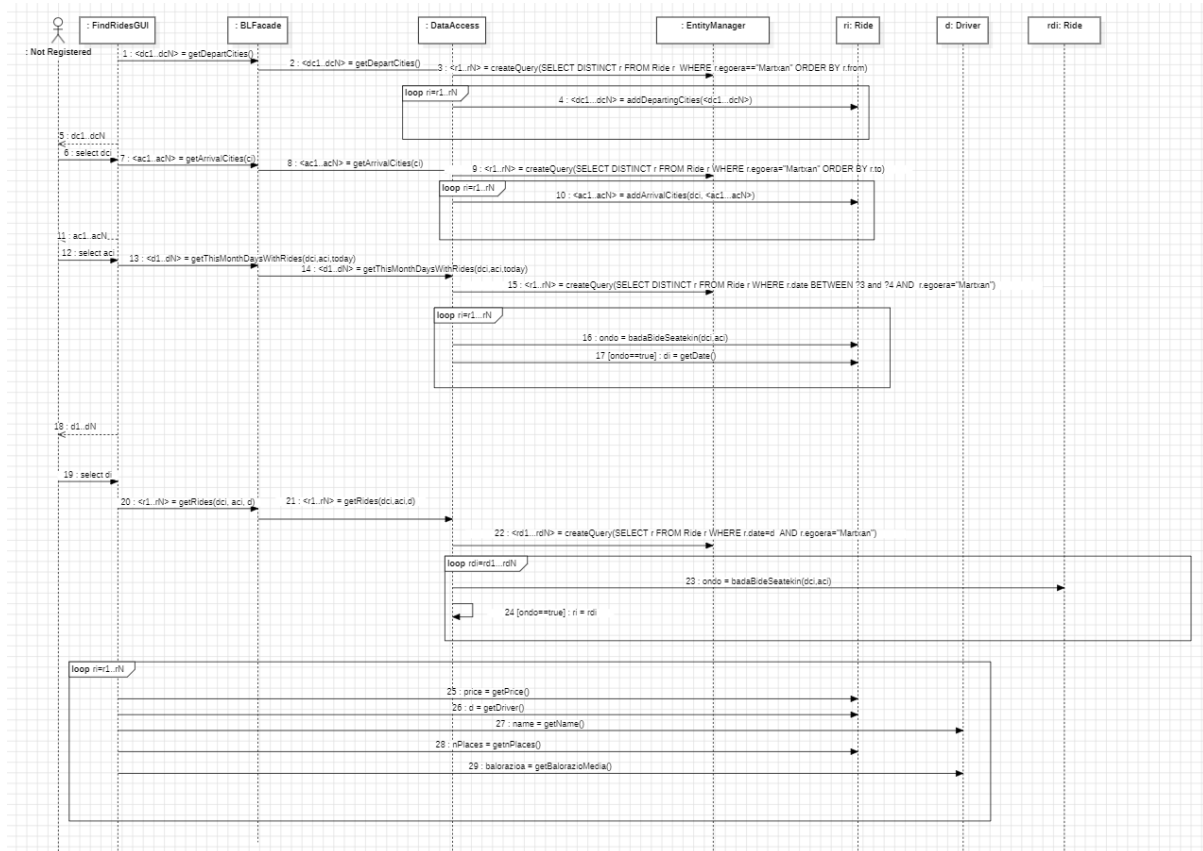
Gidariari hirteera hiria, helmuga hiria, eserleku kopurua, prezioen lista bat, data, erabiliko den kotxea eta egingo den ibilbidea sartzea eskatzen zaio. Idazterakoan errorerik egon baden konprobatzen da. BLFacaderi createRide funtzio pasa informazio guztiarekin eta honek data access-eri.

Bidaia sortzen bada konprobatzen da. Hala bada, existitzen dela exepzioa bidaltzen da, bestela gidariari bidai berri gehitzen zaio. Honek bidaia sortzen du. Geltokiak baldin baditu bidai berriak geltokia sortuko ditu.

Aldaketa guztiak datu basean gordetzen dira.

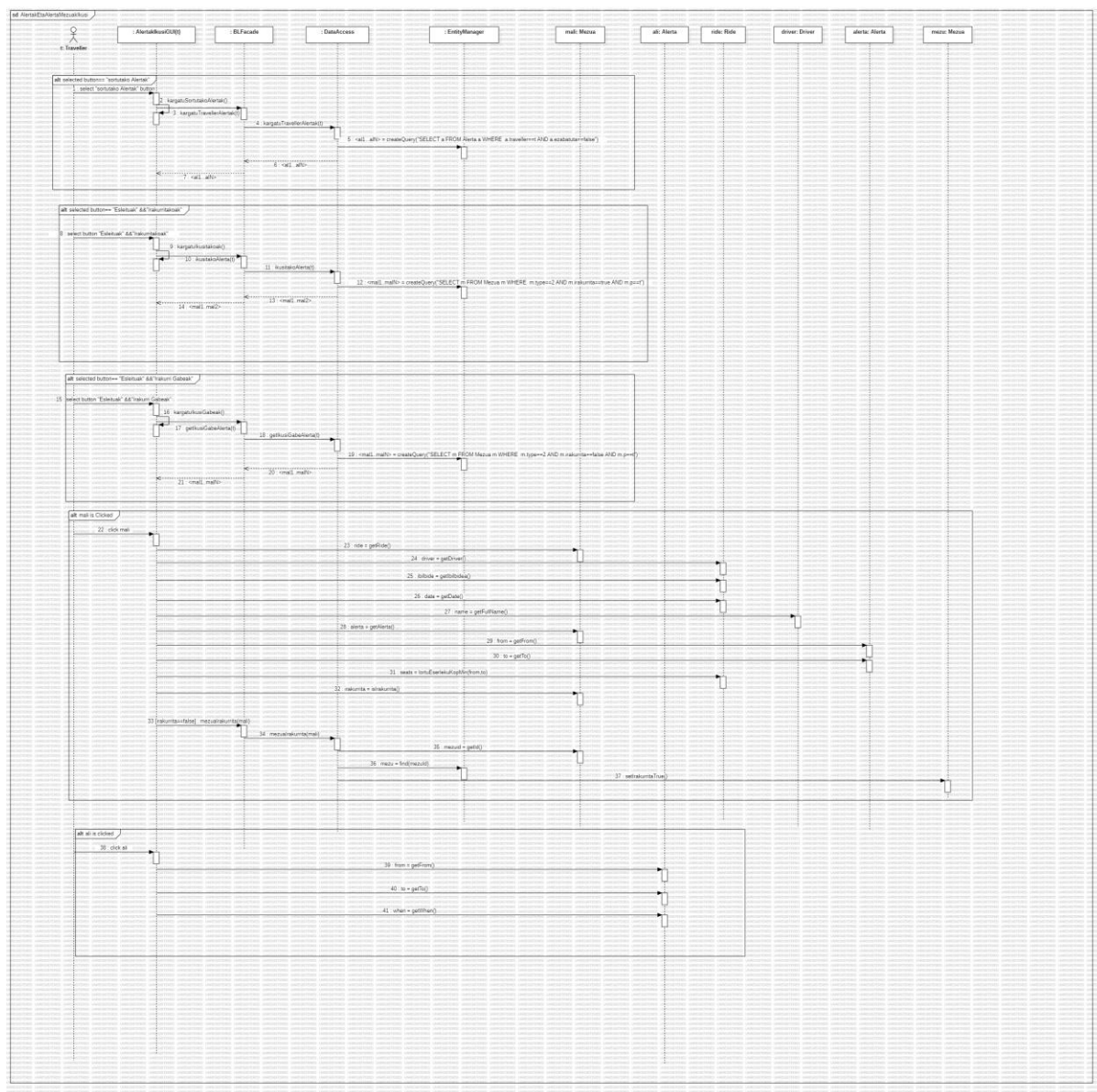
Data horretan sortu diren alerten lista sortzen da. Alerta guzti horiek begiratzeko dira eskatzen duten bidaia, sortutako ibilbidean badagoen konprobatzen da eta hala bada mezu bat bidaltzen zaio alerta sortu duen bidaiariari.

Query Rides:



Sistemak irteera hiriak lista batean gordetzen ditu. Erabiltzaileak horietako bat aukeratzen du eta sistemak irteera hiri hori duten bidaien helmugak erakusten ditu. Erabiltzaileak helmuga bat aukeratzen du eta sistema aurkeztutako egutegian egun bat aukeratzen du. Sistemak irteera hiri eta helmuga hori dituzten bidai guztiak hartzen ditu eta beraien prezioa, gidaria, gidariaren izena, eserleku kopurua eta gidariaren balorazioa erakusten du.

Alertak ikusi eta alerta mezuak ikusi:

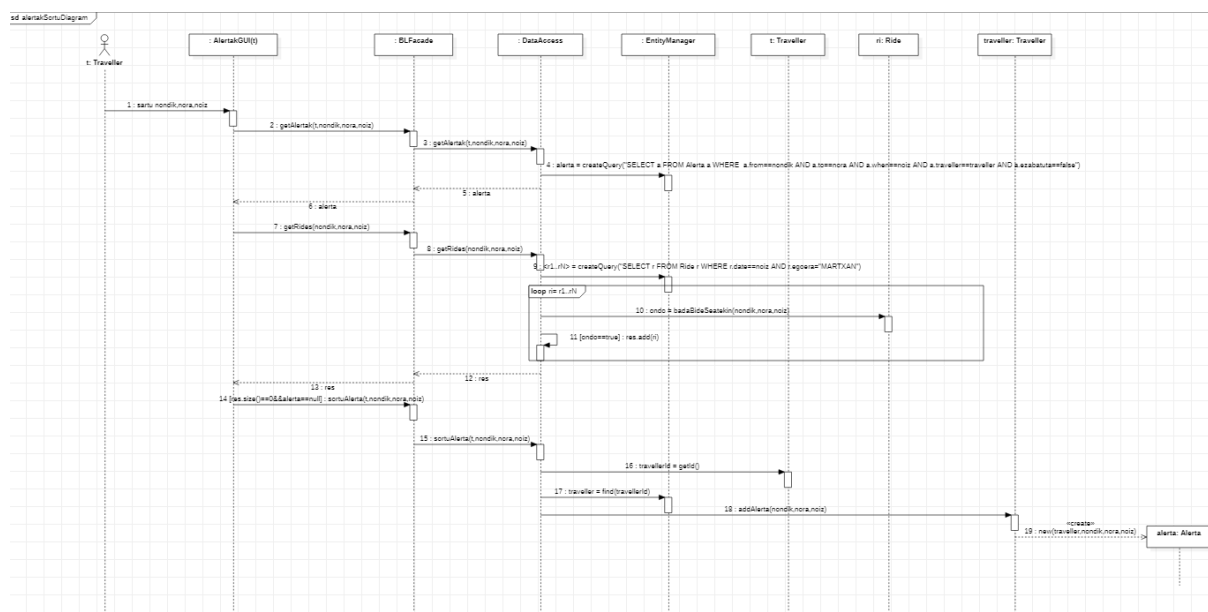


Bidaiariak aukeratu duen botoiaren arabera sistema alerta batzuk edo besteak erakusten ditu.

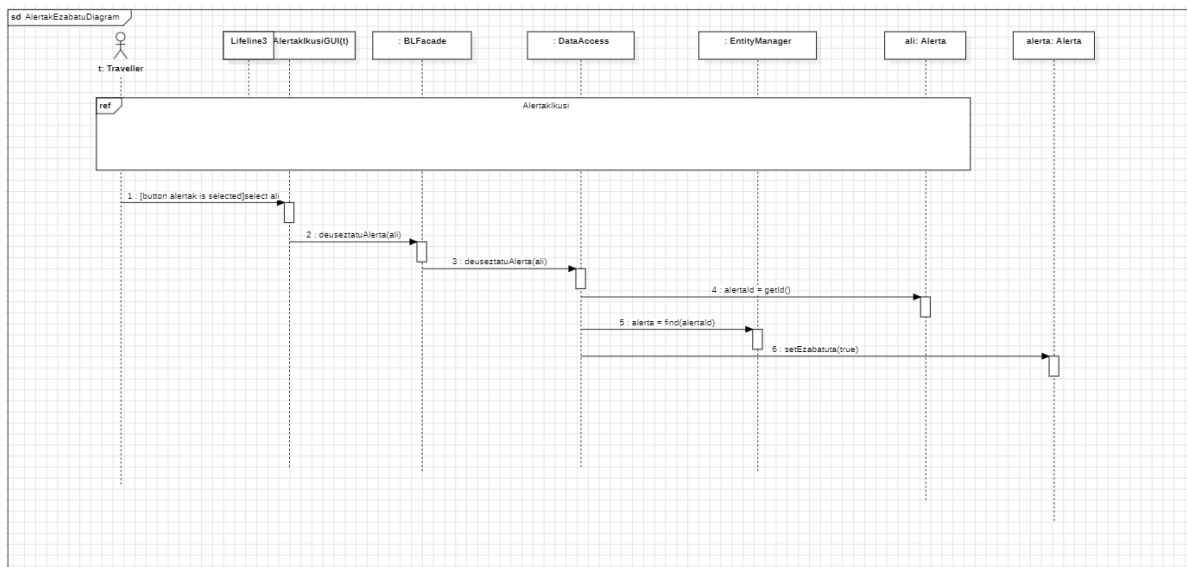
Sortu dituen alertak ikusi nahi baditu botoi hori sakatu eta sistemak bidaiari horrek sortu dituen alertak erakusten ditu, ezabatuak izan ez direnak.

Esleituak eta irakurritakoak botoiak sakatzean sistemak bidaiari horrek sortu dituen alertak eskatutako bidaia dutenak erakutsiko ditu, baldin eta bidaiariak aurretik alerta hori ikusi badu.

Bidaia-irakurketa bat aukeratzen badu, alertan gordetako irteera hiria, helmuga eta data pantailaratuko da.

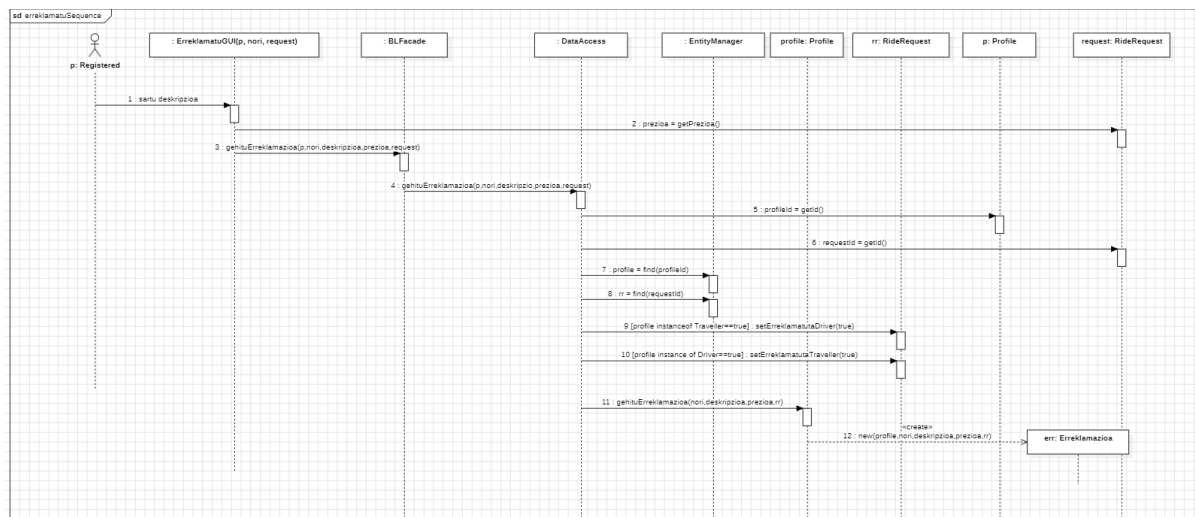


Alertak ezabatu:



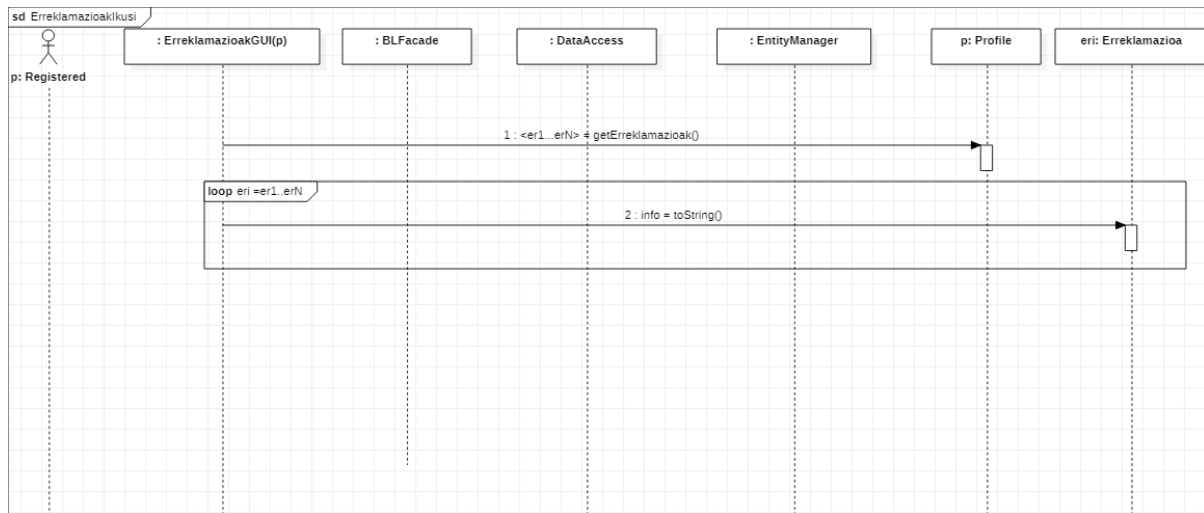
Alertak ikusiri erreferentzia egiten zaio. Bidaiariak alerta bat aukeratu badu eta kantzeltatu botoia sakatzen badu sistemak alerta deuseztatuko du ezabatua bezala gordetzen eta datu basean eguneratzen.

Erreklamatu:



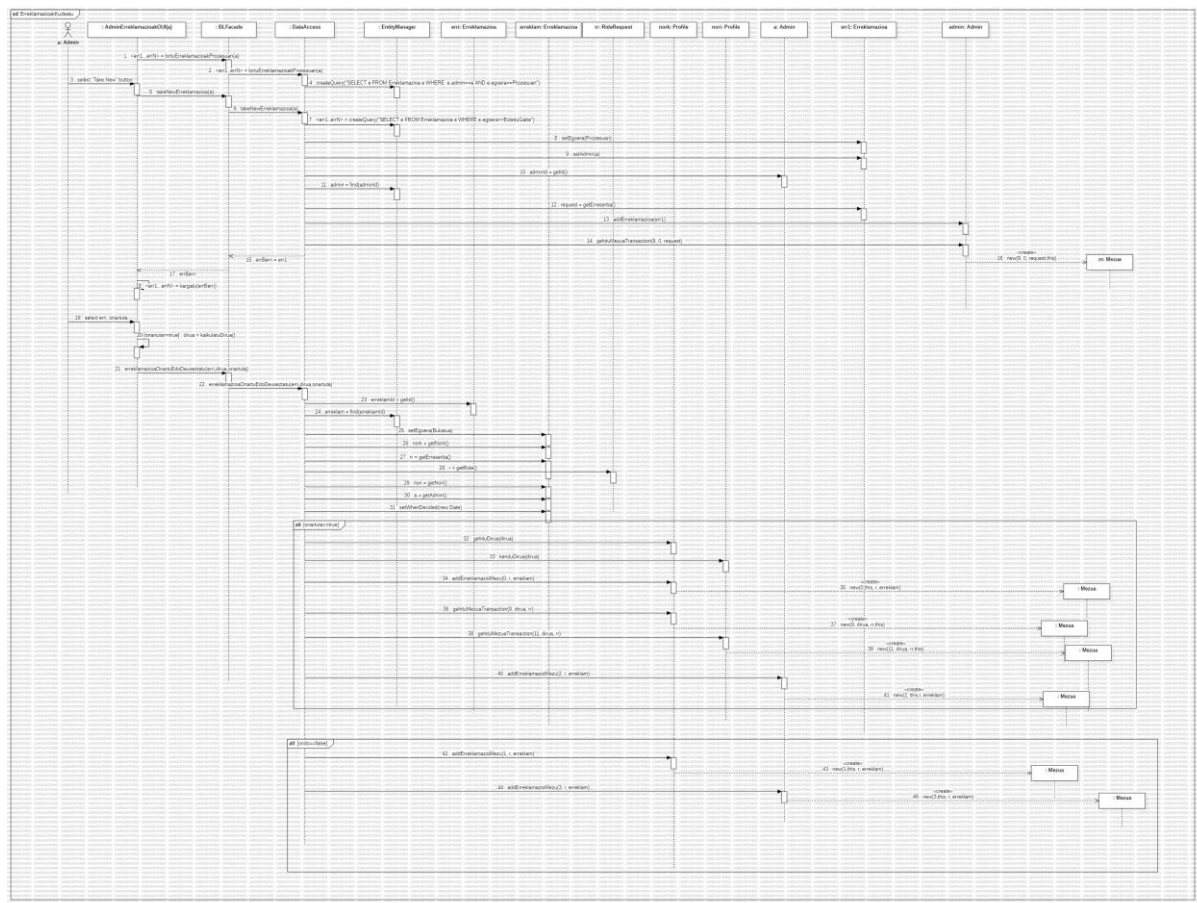
Erabiltzaileak bere arazoa idazten du. Sistema erreklamazio berria sortuko du, horretarako bidaiaren erreserbatik prezioa hartzen du eta erreklamatzailerak, erreklamaturak, arazoa eta erreserbarekin batera hartzen du. Bilatzen da zein den erreklamatzailerak eta erreserban apuntatzen da erreklamaturak gidaria edo bidaiaria izan den. Gero, erreklamatzailerari erreklamazio berria gehitzen zaio.

Erreklamazioak ikusi:



Sistemak erabiltzailearen profileko erreklamazioak hartzen ditu eta huen informazioa erakusten du.

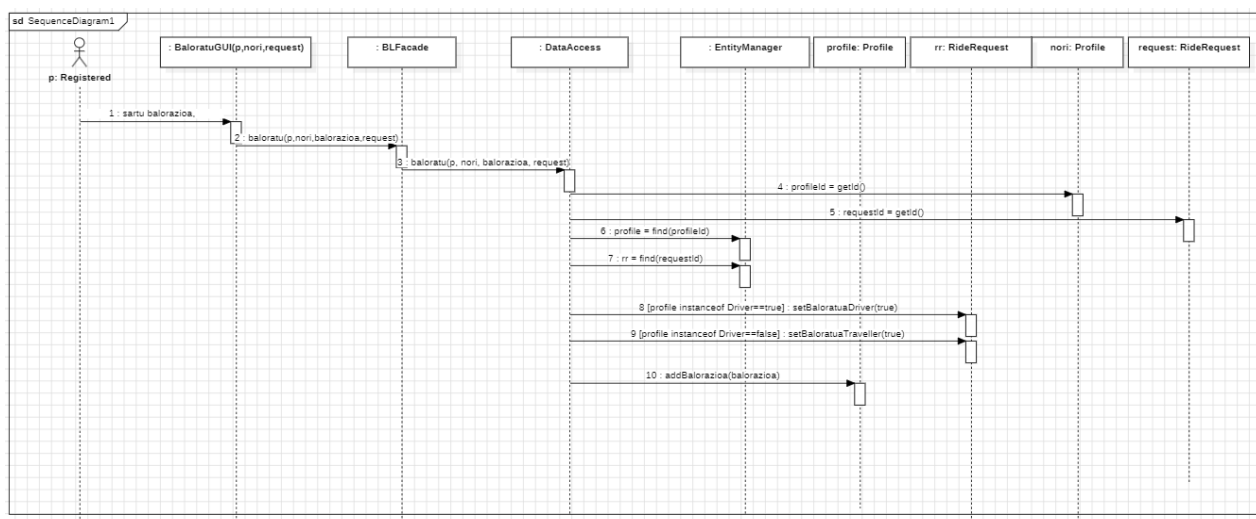
Erreklamazioak kudeatu:



Sistemak administratzaileari prozesuan dituen erreklamazioak erakusten dizkio. Administratzailea erreklamazio berri bat eskatzen badio sistemak esleitu gabeko erreklamazio bat bilatzen du, ‘prozesuan’ egoeran jartzen du eta administratzailea gehitzen dio, administratzaileari erreklamazio hau esleitzen dio eta mezu bat bidaltzen dio. Azkenik, erreklamazioaren informazioa pantailaratzen du.

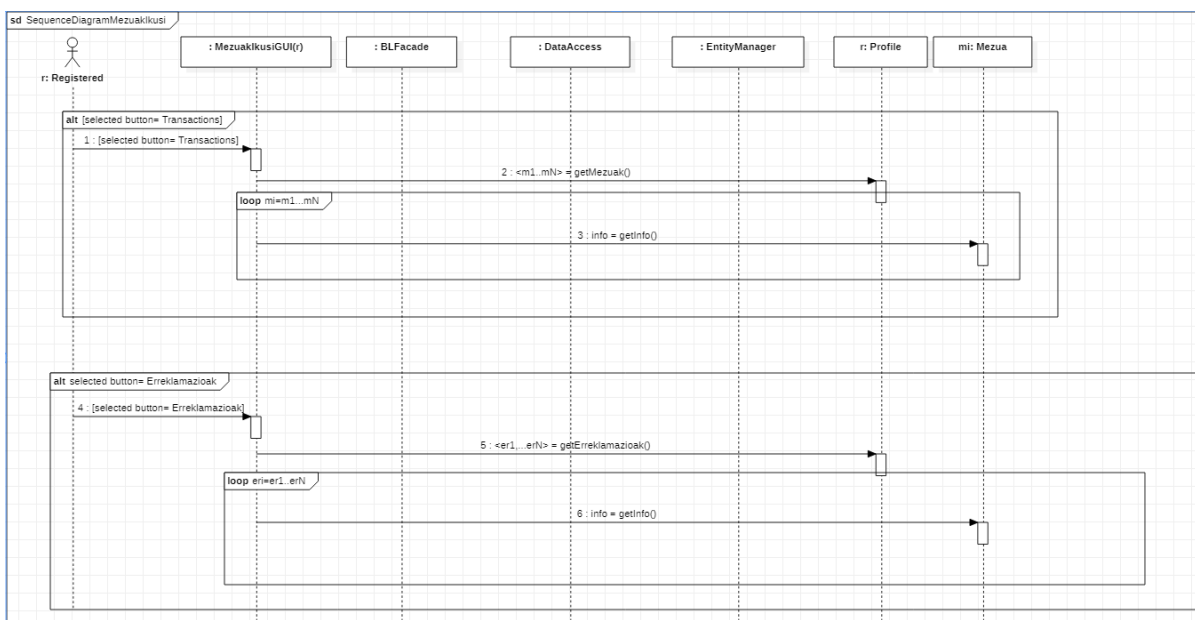
Erreklamazioa onartzen edo deuseztatzen badu, sistema jarri duen diru kopurua hartzen du erantzuana eta erreklamazioarekin batera. Erreklamazioa, bere informazioarekin, datu basetik hartzen du. Erreklamazioa ‘bukatuta’ egoerara pasatzen da. Onartua izan bada, erreklamatzailerari pasatutako diru kopurua gehitzen dio eta erreklamatuari diru kantitate hori kentzen dio. Biei transakzioari buruzko mezua bidaliko die, eta administratzaileari eta erreklamatzailerari erreklamazioaren erabakia mezu batean bidaltzen die. Aldiz, deuseztatua izan bada, bakarrik administratzailea eta erreklamatzailleak erabakia azaltzen duen mezua jasoko dute.

Baloratu:



Erabiltzaile batek balorazioa sartzen du. Sistema baloratzailerak, baloratuak, balorazioak eta ze erreserban egin den hartzen ditu. Datu basean, bi erabiltzaleak eta erreserba bilatzen dira. Erreserban apuntatzen da nor izan den baloratuak eta batuaren balorazio listan gehituko da balorazioa.

Mezuak ikusi



Erabiltzaileak “Transaction” botoia sakatzen badu, ikusiko dira diru mugimendua inplikatzeko duten mezuak: “Erreserba deuseztatu, erreserbatu, erreklamazioa onartu”... Erabiltzaileak “Demands” botoia sakatzen badu, erakutsi egingo dira, amaitutako erreklamazioak, hau da, onartuak edo deuseztatuak izan direnak, prozesuan daudenak ez.

Klase diagrama:

Implementazioa

Atal honetan agertuko dira BLFacadeImplementation klasean dauden metodoen azalpenak. Haien azpian, deitzen duten DataAccess klaseko metodoan zer egiten den laburki azalduko da.

```
/**
 * Metodo honek itzultzen ditu irteera hiri guztiak
 *
 * @return hirien collection
 */
@WebMethod public List<String> getDepartCities();
* Metodo honek itzultzen ditu helmuga hiri guztiak, emandako hiri
baten bidai guztietatik aterata
* @param from irteera hiria
* @return helmuga hiri guztiak
/**
 * Metodo honek itzultzen ditu helmuga hiri guztiak, emandako hiri baten bidai
guztietatik aterata
*
* @param from irteera hiria
* @return helmuga hiri guztiak
*/
@WebMethod public List<String> getDestinationCities(String from);

/**
 * Metodo honek sortzen du bidai bat gidari batentzat
 *
 * @param from bidaiaren irteera hiria
 * @param to bidaiaren helmuga hiria
 * @param date bidaiaren data
 * @param nPlaces eserleku erabilgarriak
 * @param driver bidaia gehituko zaion gidaria
 *
 * @return sortutako bidaia, edo null, edo exzepzio bat
 * @throws RideMustBeLaterThanTodayException bidaiaren data gaurkoa baino aurrekoa
bada
* @throws RideAlreadyExistException bidai berdina jada existitzen bada
*/
@WebMethod public Ride createRide(String from, String to, Date date, int nPlaces,
LinkedList<Float> price, String driverEmail, Kotxe kotxe, LinkedList<String>
ibilbide) throws RideMustBeLaterThanTodayException, RideAlreadyExistException;
```

```

/**
 * Metodo honek bueltatzen ditu bi tokietatik pasatzen diren eta eslekuak libre
   dituzten bidaiak
 *
 * @param from bidaiaren lehendabiziko dagoen geltokia
 * @param to bidaiaren "from" eta gero dagoen geltokia
 * @param date bidaiaren data
 * @return bidaien lista
 */
@WebMethod public List<Ride> getRides(String from, String to, Date date);

/**
 * Metodo honek bueltatzen ditu datu basetik hilabete bateko datak non jardunaldiak
   dauden
 *
 * @param from bidaiaren lehendabiziko dagoen geltokia
 * @param to bidaiaren "from" eta gero dagoen geltokia
 * @param date bidaiaren data
 * @return daten lista
 */
@WebMethod public List<Date> getThisMonthDatesWithRides(String from, String to,
Date date);

/**
 * Metodo honek deitzen du data access hasieratzeko datu basea jardunaldi eta
   galdera batzuekin. Bakarri hasieratzen da "initialize" deklaratu badago tag
   dataBaseOpenMode of resources/config.xml file -an
 *
 */
@WebMethod
public void initializeBD();

/**
 *
 * Metodo honek profil berria erregistratzen du datu basean
 *
 * @param email erregistratutakoaren emaila
 * @param name erregistratutakoaren izena
 * @param surname erregistratutakoaren abizena
 * @param user erregistratutakoaren erabiltzailea
 * @param password erregistratutakoaren pasahitza
 * @param telf erregistratutakoaren telefonoa
 * @param type bidaiaria edo gidaria den
 *
 * @return profile berria

```

```

*/
@WebMethod
public Profile register(String email, String name, String surname, String user,
String password, String telf, String type);

/**
 * Metodo honek erabiltzaile bat eta pasahitza bat sartuz, erabiltzaile eta
pasahitza
 * horiek dituen profila itzuliko du
 *
 * @param user bilatutako erabiltzailea
 * @param password erabiltzaile horren pasahitza
 * @return aurkitutako profila
 */
@WebMethod public Profile login(String user, String password);

/**
 * Metodo honek pasatutako erabiltzaileari pasatutako diru kantitatea gehituko dio
 * bere kontura
 *
 * @param dirua gehituko den diru kantitatea
 * @param p diru hori sartuko zaion erabiltzailearen profila
 *
 */
@WebMethod public void gehituDirua(float dirua, Profile p);

/**
 * Metodo honek pasatutako erabiltzaileari pasatutako diru kantitatea kenduko dio
 * bere kontutik
 *
 * @param dirua kenduko den diru kantitatea
 * @param p diru hori kenduko zaion erabiltzailearen profila
 *
 */
@WebMethod public void kenduDirua(float dirua, Profile p);

/**
 * Metodo honek pasatutako bidaiariari pasatutako diru kantitatea gehituko dio
 * bere kontura
 *
 * @param dirua gehituko den diru kantitatea
 * @param p diru hori sartuko zaion bidaiaria
 *
 */
@WebMethod public void sartuDirua(float dirua, Traveller traveller);

/**
 * Metodo honek pasatutako gidariari pasatutako diru kantitatea aterako dio
 * bere kontutik
 *
 * @param dirua aterako den diru kantitatea
 * @param p diru hori aterako zaion gidaria

```

```

*/
@WebMethod public void ateraDirua(float dirua, Driver driver);

/**
 * Metodo honek bidai baten erreserba eskakizuna sortuko du
 *
 * @param time erreserba eskaeraren data (momentuan egidako data)
 * @param ride erreserbatu nahi den bidaia
 * @param traveller bidaia erreserbatu nahi duen bidaiaria
 * @param seats erreserbatu nahi dituen eserleku kopurua
 * @param from bidaian dagoen geltoki bat
 * @param to bidaian from eta gero dagoen geltoki bat
 *
 * @return erreserba eskakizuna
 */
@WebMethod public RideRequest erreserbatu(Date time, Ride ride, Traveller traveller,
int seats, String from, String to);

/**
 * Metodo honek erreserba eskakizun bat onartzen edo deuseztatzen du, behar den
pertsoneari dirua itzuliz (beharrezkoa bada)
 *
 * @param request onartzen edo deuseztatzen den erreserba eskakizuna
 * @param onartuta eskakizuna onartu baden gordetzen du
 *
 */
@WebMethod public void onartuEdoDeuseztatu(RideRequest request, boolean onartuta);

/**
 * Metodo honek gidari baten bidaiak, martxan daudenak eta leku libreak
 * dituztenak, lista batean bueltatzen ditu
 * RideContainer itzultzen dugu web zerbitzua dela eta
 * @param d gidaria
 * @return gidariaren bidaien lista
 */
@WebMethod public List<RideContainer> getRidesOfDriver(Driver d);

/**
 * Metodo honek bidai baten erreserba eskakizun guztiak lista batean bueltatzen
ditu
 *
 * @param ride bidaia
 * @return bidaiaren erreserba eskakizun guztien lista
 */
@WebMethod public List<RideRequest> getRidesRequestsOfRide(Ride ride);

```

```

/**
 * Metodo honek erabiltzaile batek duen diru kantitatea itzultzen du
 *
 * @param p erabiltzaile horren profila
 * @return bere kontuan duen diru kantitatea
 */
@WebMethod public float getMoney(Profile p);

/**
 * Metodo honek bidaiari baten erreserba eskakizun guztiak lista batean bueltatzen ditu
 *
 * @param traveller bidaiaria
 * @return bidaiariaren erreserba eskakizun guztien lista
 */
@WebMethod public List<RideRequest> getRidesRequestsOfTraveller(Traveller traveller);

/**
 * Metodo honek gidari baten bidaia guztiak (Container-ak) lista batean bueltatzen ditu
 *
 * @param d gidaria
 * @return gidariaren bidaien lista(Container-ak)
 */
@WebMethod public List<RideContainer> getAllRidesOfDriver(Driver driver);

/**
 * Metodo honek erreserba eskakizun baten bidaia itzultzen du
 *
 * @param erreserba erreserba eskakizuna
 * @return erreserba eskakizunaren bidaia
 */
@WebMethod public Ride getRideFromRequest(RideRequest erreserba);

/**
 * Metodo honek gidari baten kotxe guztiak lista batean bueltatzen ditu
 *
 * @param driver gidaria
 * @return gidariaren kotxeen lista
 */
@WebMethod public List<Kotxe> getKotxeGuztiak(Driver driver);

/**
 * Metodo honek gidari bati kotxe berri bat gehitzen dio
 * eta ondo gehitu den edo ez itzultzen du
 *
 * @param marka kotxearen marka
 * @param modelo kotxearen modeloa
 * @param matrikula kotxearen matrikula

```

```

    * @param tokiKop kotxearen eserleku kopurua
    * @param driver gidaria
    * @return boolean ondo gehitu den edo ez
    */
@WebMethod public boolean createCar(String marka, String modelo, String matrikula,
int tokiKop, Driver driver);

/**
 * Metodo honek erabiltzaile batek dituen mezuen lista itzultzen du
 *
 * @param p erabiltzaile horren profila
 * @return mezuen lista
 */
@WebMethod public List<Mezua> getMezuak(Profile p);

/**
 * Metodo honek erabiltzaile batek dituen erreklamazio mezuen lista itzultzen du
 *
 * @param p erabiltzaile horren profila
 * @return erreklamazio mezuen lista
 */
@WebMethod public List<Mezua> getErreklamazioMezuak(Profile p);

/**
 * Metodo honek bidai bat kantzelatzen du
 *
 * @param r kantzelatuko den bidaia
 */
@WebMethod public void Kantzelatu(Ride r);

/**
 * Metodo honek erreserbatutako bidai bat gauzatu den jarriko du, horretarako,
bidaia bereko profile bereko request guztiak eginda edo ez eginda bezala jarriko
ditu. Aldaketa egiten den erreklamazio bakoitzeko dirua gehituko dio driverra-ri edo
traveller-ari, segun bidaia egin dela esan den ala ez

```

```

    *
    * @param request bidaiaren erreserba
    * @param onartuta bidaia gauzatu bada edo ez
    *
    */
@WebMethod public void egindaEdoEzEgina(RideRequest request, boolean onartuta);

/**
 * Metodo honek gidari batek martxan dituen bidaiak itzultzen ditu nahiz
 * eta tokirik gabe izatea.
 * RideContainer itzultzen dugu web zerbitzuagatik
 *
 * @param d gidaria
 * @return gidariak martxan dituen bidaien lista

```

```

*/
@WebMethod public List<RideContainer> getEginRidesOfDriver(Driver d);

/**
 * Metodo honek erabiltzaile bat baloratzen du bidai bategatik
 *
 * @param balorazioa eman zaion baloraketa
 * @param nori baloratuko den erabiltzailea
 * @param r baloratu den bidaiaren erreserba
 *
 */
@WebMethod public void baloratu(int balorazioa, Profile nori, RideRequest r);

/**
 * Metodo honek gidari baten balorazioen media bueltatzen du
 * @param driver gidaria
 * @return float balorazioa
 */
@WebMethod public float getBalorazioMedia(Driver driver);

/**
 * Metodo honek alerta bat sortzen du bidai batez abisatzeko
 *
 * @param from irteera hiria
 * @param to helmuga hiria
 * @param when noiz nahi duzun bidaia egitea
 * @param t alerta sortu duen bidaiaria
 *
 */
@WebMethod public void sortuAlerta(Traveller t, String from, String to, Date when);

/**
 * Metodo honek pasatutako parametroak dituen alerta itzultzen du
 *
 * @param from irteera hiria
 * @param to helmuga hiria
 * @param when bidaiaren data
 * @return aurkitu den alerta
 */
@WebMethod public Alerta getAlerta(String from, String to, Date when);

/**
 * Metodo honek itzultzen ditu erabiltzaile batek egin dituen erreklamazio guztiak
 *
 * @param p erabiltzailea
 * @return erreklamazioen lista

```



```

*/
@WebMethod public List<Erreklamazioa> getErreklamazioak(Profile p);

/**
 * Metodo honek erreklamazio berri bat sortzen du eta esleitu gabe moduan gordetzen
da datu basean
 *
 * @param p erreklamatzailerak
 * @param nori erreklamatuak
 * @param deskripzioa erreklamazioaren arrazoia
 * @param prezioa erreklamatzaren ari den bidai erreseerbaren prezioa
 * @param r bidai erreseerba
 *
 */
@WebMethod public void gehituErreklamazioa(Profile p, Profile nori, String
deskripzioa, float prezioa, RideRequest r);

/**
 * Metodo honek administratzaile bateri esleitu gabeko erreklamazio bat esleitzen
dio
 *
 * @param a erreklamazio berria eskatu duen administratzailea
 * @return esleitu zaion erreklamazioa
 *
 */
@WebMethod public Erreklamazioa takeNewErreklamazioa(Admin a);

/**
 * Metodo honek itzultzen ditu bidaia hori eginda izan dela esan duten bidaiariak
 *
 * @param ride begiratu nahi den bidai
 * @return bidaiariaren lista
 *
 */
@WebMethod public List<Traveller> getTravellersOfRideDone(Ride ride);

/**
 * Metodo honek itzultzen ditu bidaia hori eginda ez dela izan esan duten bidaiariak
 *
 * @param ride begiratu nahi den bidai
 * @return bidaiariaren lista
 *
 */
@WebMethod public List<Traveller> getTravellersOfRideNotDone(Ride ride);

/**
 * Metodo honek erreklamazio bat onartu edo deuseztatzen den gordetzen du.
 * Onartu bada, erreklamatuari dirua kendu eta erreklamatzailerari diru sartuko dio
eta erreklamazio emaitzaren mezuak bidaliko dira
 * Deuseztatu bada, bakarrik mezuak bidaliko dira
 *
 * @param erreklamazioa aukeratu den erreklamazioa

```

```

    * @param kantitatea onartu baden erreklamatu duen erabiltzaileari itzuliko zaion
    diru kantitatea
    * @param onartuta onartu edo deuseztatu baden erreklamazioa
    */
@WebMethod      public      void      erreklamazioaOnartuEdoDeuseztatu(Erreklamazioa
erreklamazioa, float kantitatea, boolean onartuta);

/**
 * Metodo honek egiten duena da, Pasatutako mezua irakurrita= true moduan jarri
egiten du.
 * @param alerta Alerta mezua
 *
 */
@WebMethod      public void      mezuaIrakurrita(Mezua alerta);

/**
 * Metodo honek itzultzen ditu bidaiari baten ikusitako alertak lista batean
 *
 * @param traveller bidaiaria
 * @return bidaiariaren ikusitako alerten lista
 */
@WebMethod      public List<Mezua> ikusitakoAlerta(Traveller traveller);

/**
 * Metodo honek itzultzen ditu bidaiari baten ikusi gabeko alertak lista batean
 *
 * @param traveller bidaiaria
 * @return bidaiariaren ikusi gabeko alerten lista
 */
@WebMethod      public List<Mezua> getIkusiGabeAlerta(Traveller traveller);

/**
 * Metodo honek itzultzen ditu bidaiari baten alertak lista batean
 *
 * @param traveller bidaiaria
 * @return bidaiariaren alerten lista
 */
@WebMethod      public List<Alerta> kargatuTravellerAlertak(Traveller traveller);

/**
 * Metodo honek alerta bat deuseztatzen du
 *
 * @param alerta deuseztatuko den alerta
 */
@WebMethod      public void      deuseztatuAlerta(Alerta alerta);

/**
 * Metodo honek erreserba eskaera baten bidaiaria edo gidaria baloratu baden
bueltatzen du

```

```

*
* @param request erreserba eskaera
* @param gidari gidaria edo bidaiaria begiratu behar baden jakiteko
* @return erreklamaturata badagoen
*/
@WebMethod public boolean isBaloratua(RideRequest request, boolean gidari);

/**
* Metodo honek erreserba eskaera baten bidaiaria edo gidaria erreklamatu baden
bueltatzen du
*
* @param request erreserba eskaera
* @param gidari gidaria edo bidaiaria begiratu behar baden jakiteko
* @return erreklamaturata badagoen
*/
@WebMethod public boolean isErreklamaturata(RideRequest request, boolean gidari);

/**
* Metodo honek pasatutako bidaiaren lehenengo eskaera lortzen du
*
* @param r bilatu nahi den bidaia
* @param t bilatu nahi den bidaiaria
* @return bidaiariaren lehenengo erreserba eskaera
*/
@WebMethod public RideRequest lortuLehenRequestBidaiakoa(Ride r, Traveller t);

/**
* Metodo honek itzuli egiten duena da, Bidaia batean, request-aren ibilbidea
egitean, zenbat eserleku dauden. Hau da RideRequest-aren from-etik to-ra zenbat
eserleku dauden libre
*
* @param ride Bidaia
* @param request Bidaia Eskaera
* @return eserleku kopurua(Integer)
*/
@WebMethod public int lortuZenbatEserlekuGeratu(Ride ride, RideRequest request);

/**
* Metodo honek administratzaile batek prozesuan dituen erreklamazioak itzultzen
ditu
*
* @param a administratzailea
* @return erreklamazioen lista
*/
@WebMethod public List<Erreklamazioa> lortuErreklamazioakProzesuan(Profile a);

```

}

Ondorioak

Proiektu honi esker software baten fase guztiak garatzeko aukera paregabea izan dugu; proiektua buruzko eskemetatik hasita, programa garatzera arte. Hauek dira ateratako ondorioak:

- Proiektu konplexuetan bereziki, ez da batere erraza hutsetik hastea, eta dirudiena baino garrantzitsuagoa da modelaketa on bat egitea, ondorengo ataleetan arazoak saihesteko.
- Taldeetan lan egitea oso ohikoa da, eta batzutan zaila izan daiteke talde moduan lan egitea; horregatik iruditzen zaigu funtsezkoa dela taldekideekin komunikazio eta koordinazio on bat izatea lehen momentutik, hurrengo arazoei aurre egiteko oso lagungarria izango baita.
- Proiektua zehar arazoak izatea ohikoa dela ere ikasi dugu, horrelako proiektu luzeetan oso ohikoa da gauzak gaizki ateratzea, eta dena ez joatea pentsatuta zegoen modura. Baina erronka horiei aurre egiteak ondo egiten duen kode batera hurbiltzen gaitu.
- Dokumentazio zuzen eta argi bat egiteak ere asko lagundu digu, adibidez, aurretik idatzitako kode bat aldatu behar ezker, kode horretan jadanik dokumentazioa egoteak denbora asko aurreztu dezake kodigoa ulertzeko garaian.
- Denbora ondo planifikatzea ere garrantzitzua dela ikusi dugu, beti sortzen baitira arazoak, eta azken momentua ez arrapatzeko oso lagungarria ikusi dugu meta txikiak jartzea entregaren aurretik.
- Probak egitea ezinbestekoa dela iruditu zaigu, kodea garatzen den bitartean, nahiz eta kodigok errorerik ez botatzea, izan leike praktikan ondo ez egitea, eta horretarako ezinbestekoa iruditu zaigu kodea garatzen den bitartean probak egiten joatea.
- Kodea funtzionatzea batzutan ez da nahikoa, batzutan izan leike kodigo batek ondo egitea, baina era oso motel batean, edo gero erabiltzailearentzat zaila egingo zaion modu batean. Horregatik garrantzitzua iruditzen zaigu kodigoa ondo egiteaz gain, era erraz eta eraginkor batean ere egitea.

Azkenean, proiektu honi esker, software bat garatzeko orduan izan ahal ditugun erronkak ezagutzeko balio digu, eta erronka horiei soluzioak eman ahal izateko; eta antolaketa eta talde lana on bat izatea ere garrantzitzua dela. Ikasgai honi esker, software bat garatzeko esperientzia eta ezagutza eman digu, eta etorkizunean egin beharreko proiektuetarako oso lagungarria izango diren hainbat tresna, metodo eta trikimailu.

Azkenik, aldatzeko gauza bat gomendatuko bagenu, izango litzateke azken iterazioko lan karga, guk zortea dugu eta taldekoak ez ditugu global asko, baina bada jendea global-ak dituenak, eta azken iterazioak denbora gutxi ematen du beste ikasgaiak prestatzeko txostena dela eta, txostena oso luzea delako. Gure gomendioa izango litzateke txostenean gauza gutxiago eskatzea, edo gutxienik emandako azalpen guztiak 3. iteraziokoak izatea.

Bideoaren URL-a

<https://youtu.be/BU0cDk3ZcoQ>