HORSE RACES DOKUMENTAZIO FINALA

1. Egilea: Julen Larrañaga

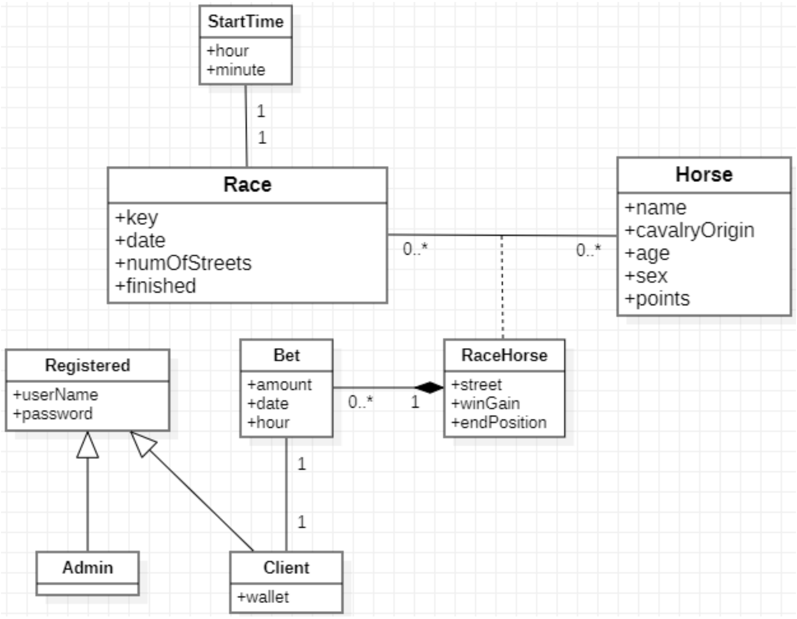
2. Web Zerbitzuak eta Internalizazioa: Internalizazioa bakarrik garatu dut.

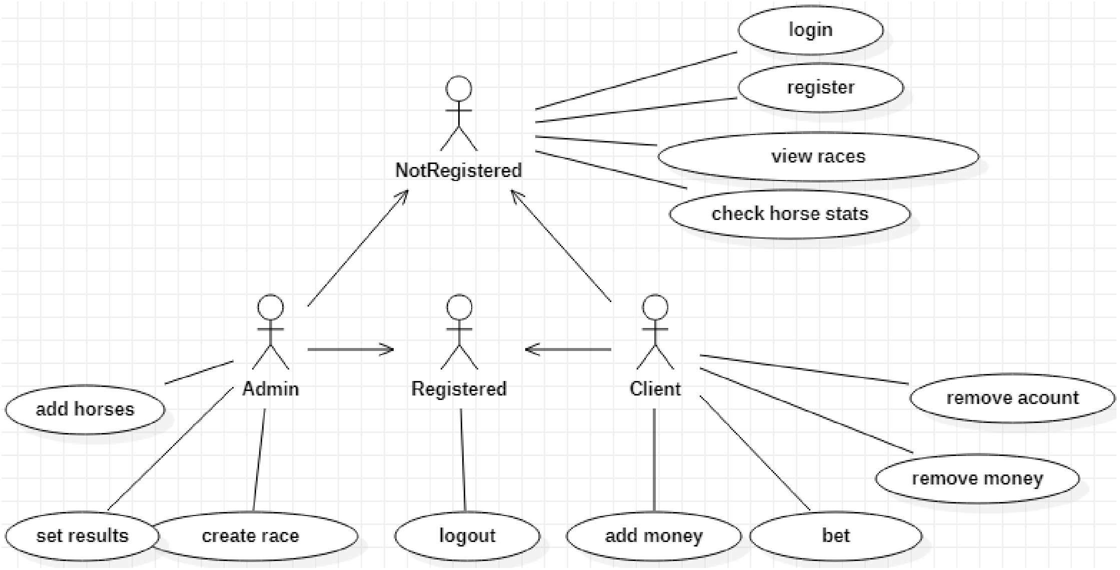
3. Sarrera:

HorseRaces zaldi lasterketen apustu sistema bat da, non, lasterketa horietako zaldi baten gainean apustu egin daitekeen. Administratzailea arduratuko da lasterketa sortu, lasterketara zaldiak txertatu eta lasterketaren emaitzak ipintzeaz.

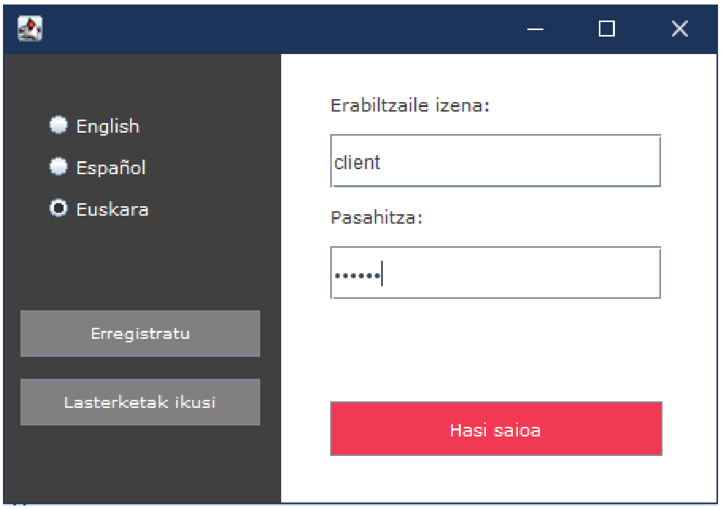
Apustu egiteko beharrezkoa da bezero bezala erregistratua egotea, eta nahikoa diru izatea dagokion bezero kontuan. Horregatik, bezeroek bere erabiltzaile izen eta pasahitza erabiliz erregistratu eta saioa hasi ondoren, kontuari dagokion "kartera" kudeatzeko aukera izango dute, besteak beste.

4. Eskakizun Bilketa:





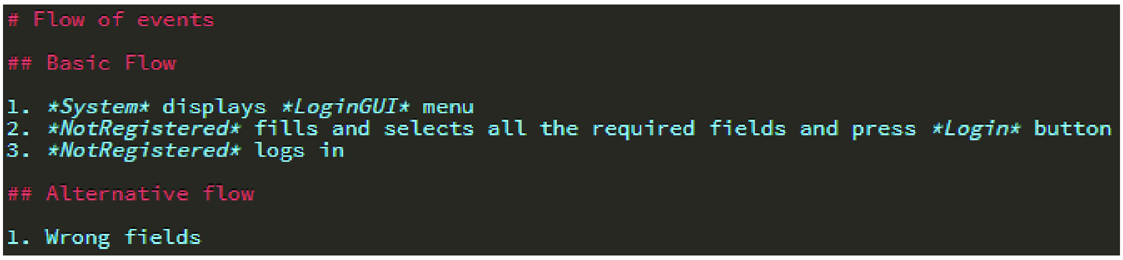
LOGIN



○ English
○ Español
● Euskara

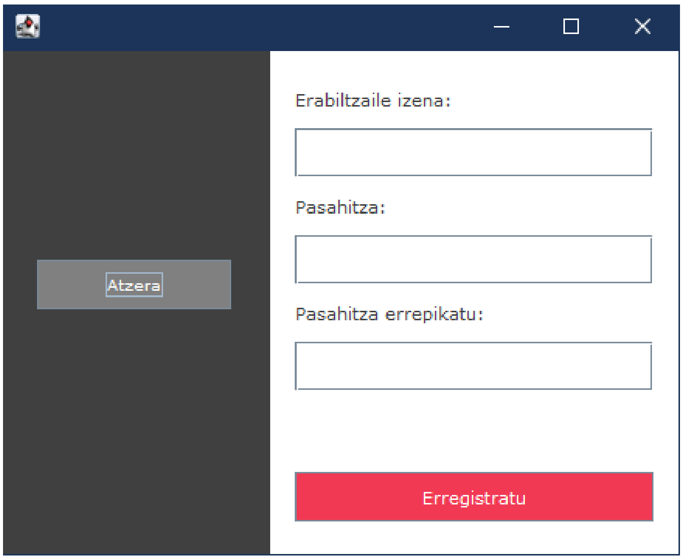
Erabiltzaile izena: client

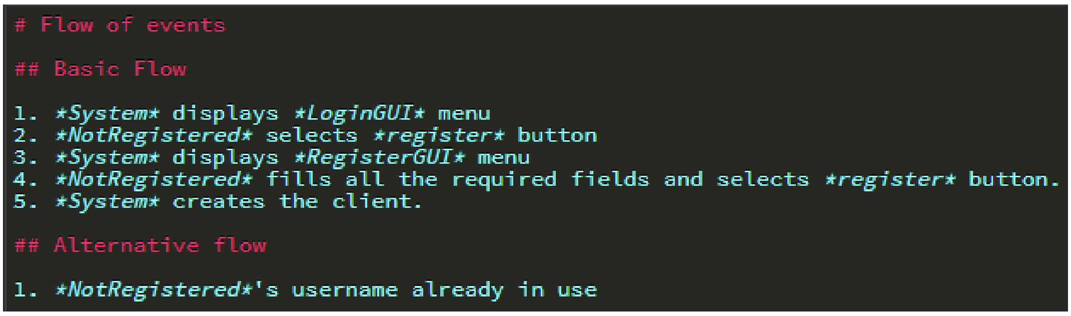
Pasahitza: ••••••



```
# Flow of events

## Basic Flow

1. *System* displays *LoginGUI* menu
2. *NotRegistered* fills and selects all the required fields and press *Login* button
3. *NotRegistered* logs in

## Alternative flow

1. Wrong fields
```

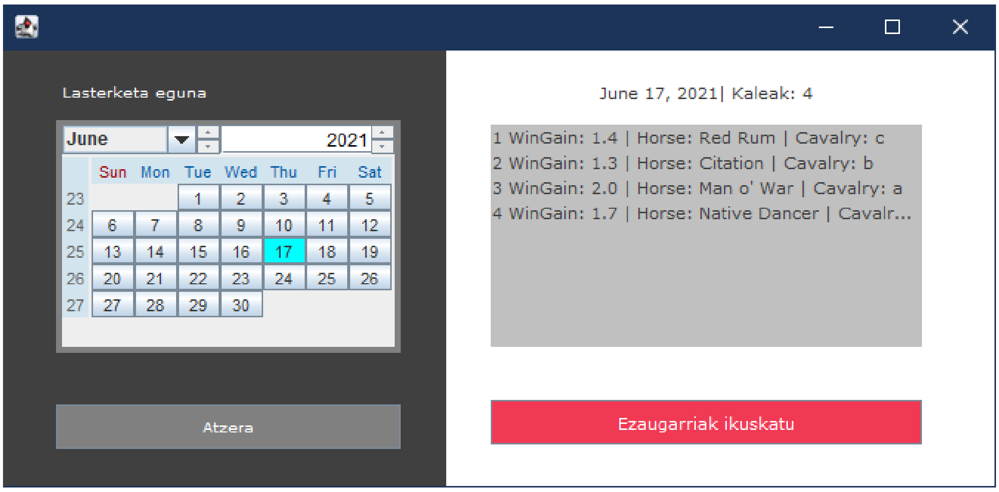
REGISTER





```
# Flow of events

## Basic Flow

1. *System* displays *LoginGUI* menu
2. *NotRegistered* selects *register* button
3. *System* displays *RegisterGUI* menu
4. *NotRegistered* fills all the required fields and selects *register* button.
5. *System* creates the client.

## Alternative flow

1. *NotRegistered*'s username already in use
```

VIEW RACES



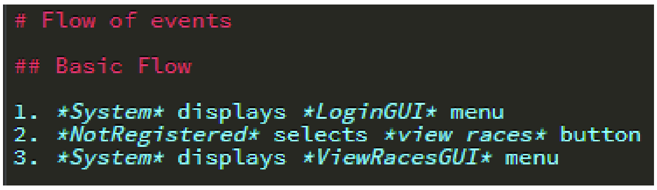
Lasterketa eguna

June 2021

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

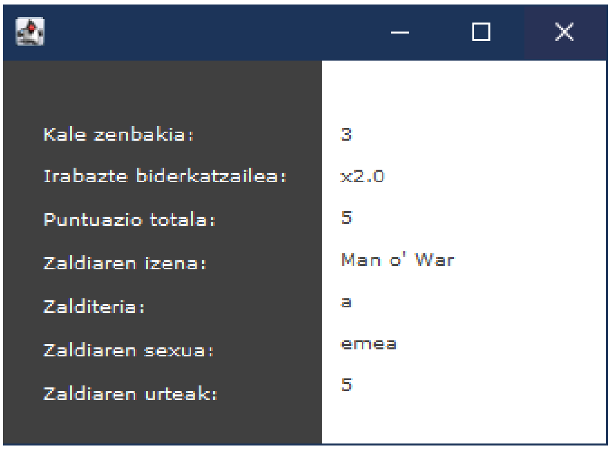
June 17, 2021| Kaleak: 4

```
1 WinGain: 1.4 | Horse: Red Rum | Cavalry: c
2 WinGain: 1.3 | Horse: Citation | Cavalry: b
3 WinGain: 2.0 | Horse: Man o' War | Cavalry: a
4 WinGain: 1.7 | Horse: Native Dancer | Cavalr...
```



```
# Flow of events

## Basic Flow

1. *System* displays *LoginGUI* menu
2. *NotRegistered* selects *view races* button
3. *System* displays *ViewRacesGUI* menu
```

CHECK HORSE STATS



Kale zenbakia: 3

Irabazte biderkatzailea: x2.0

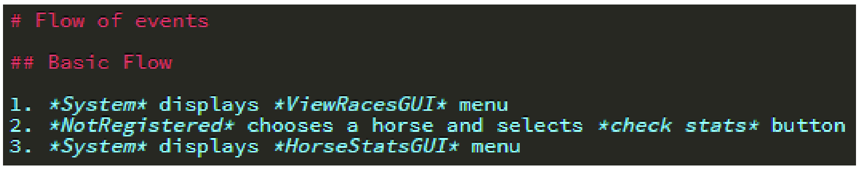
Puntuazio totala: 5

Zaldiaren izena: Man o' War

Zalditeria: a

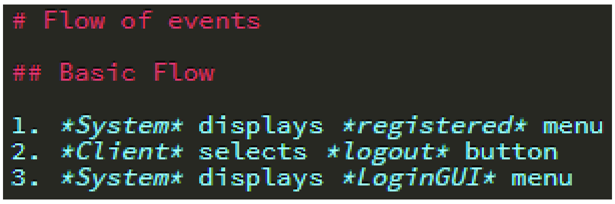
Zaldiaren sexua: emea

Zaldiaren urteak: 5



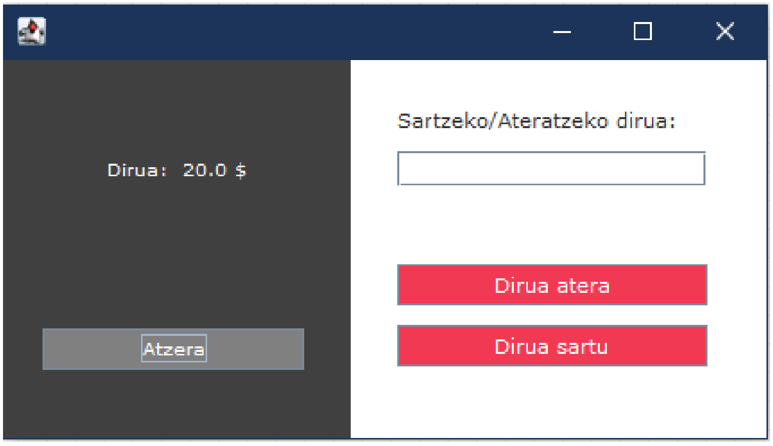
```
# Flow of events

## Basic Flow

1. *System* displays *ViewRacesGUI* menu
2. *NotRegistered* chooses a horse and selects *check stats* button
3. *System* displays *HorseStatsGUI* menu
```

LOGOUT



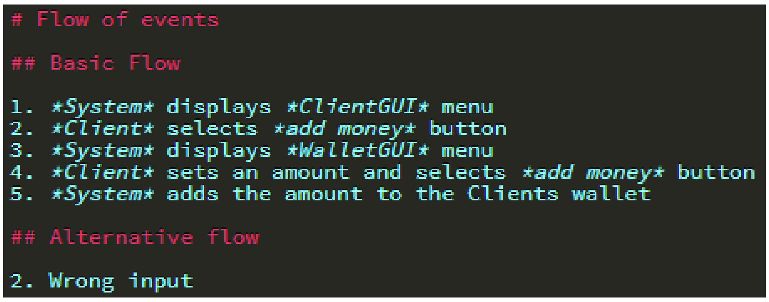
```
# Flow of events

## Basic Flow

1. *System* displays *registered* menu
2. *Client* selects *logout* button
3. *System* displays *LoginGUI* menu
```

ADD / REMOVE MONEY



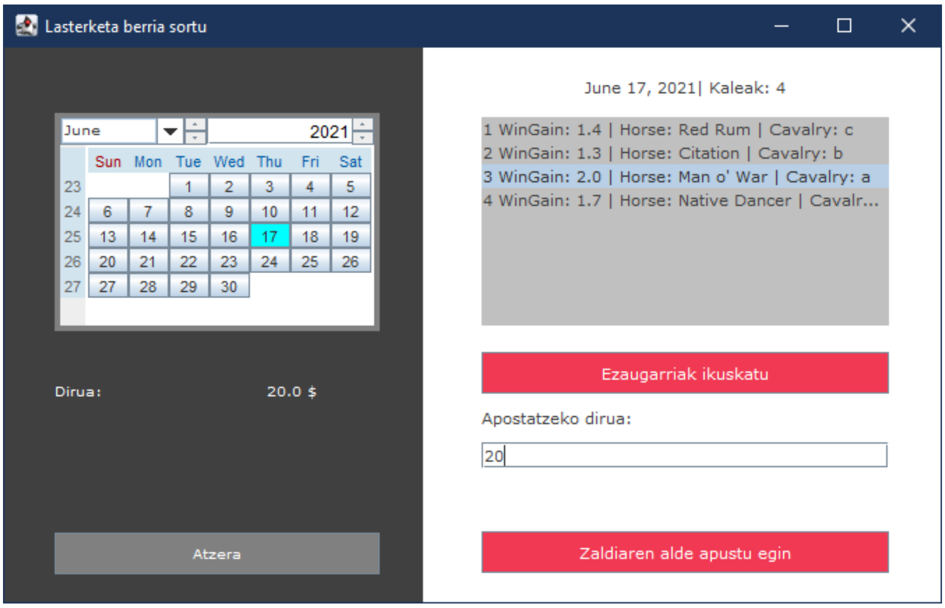
Dirua: 20.0 $

Sartzeko/Ateratzeko dirua:



```
# Flow of events

## Basic Flow

1. *System* displays *ClientGUI* menu
2. *Client* selects *add money* button
3. *System* displays *WalletGUI* menu
4. *Client* sets an amount and selects *add money* button
5. *System* adds the amount to the Clients wallet

## Alternative flow

2. Wrong input
```

BET



June 2021

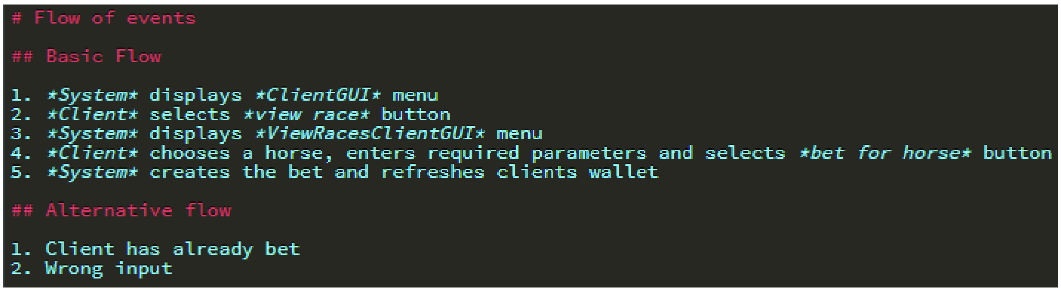
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
23			1	2	3	4	5
24	6	7	8	9	10	11	12
25	13	14	15	16	17	18	19
26	20	21	22	23	24	25	26
27	27	28	29	30			

Dirua: 20.0 $

June 17, 2021| Kaleak: 4

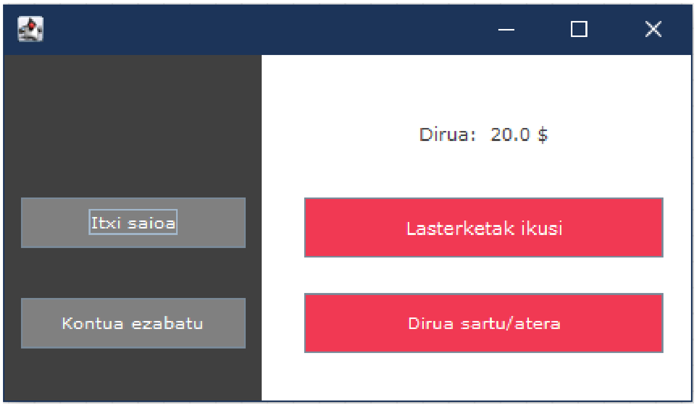
1 WinGain: 1.4 | Horse: Red Rum | Cavalry: c
2 WinGain: 1.3 | Horse: Citation | Cavalry: b
3 WinGain: 2.0 | Horse: Man o' War | Cavalry: a
4 WinGain: 1.7 | Horse: Native Dancer | Cavalr...

Apostatzeko dirua: 20

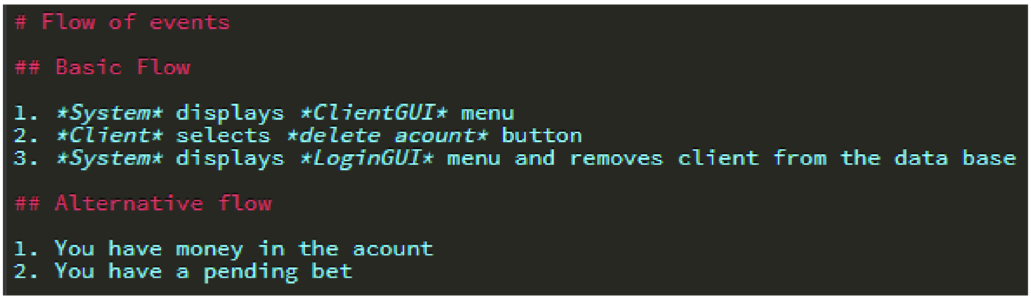


```
# Flow of events

## Basic Flow

1. *System* displays *ClientGUI* menu
2. *Client* selects *view race* button
3. *System* displays *ViewRacesClientGUI* menu
4. *Client* chooses a horse, enters required parameters and selects *bet for horse* button
5. *System* creates the bet and refreshes clients wallet

## Alternative flow

1. Client has already bet
2. Wrong input
```

REMOVE ACCOUNT

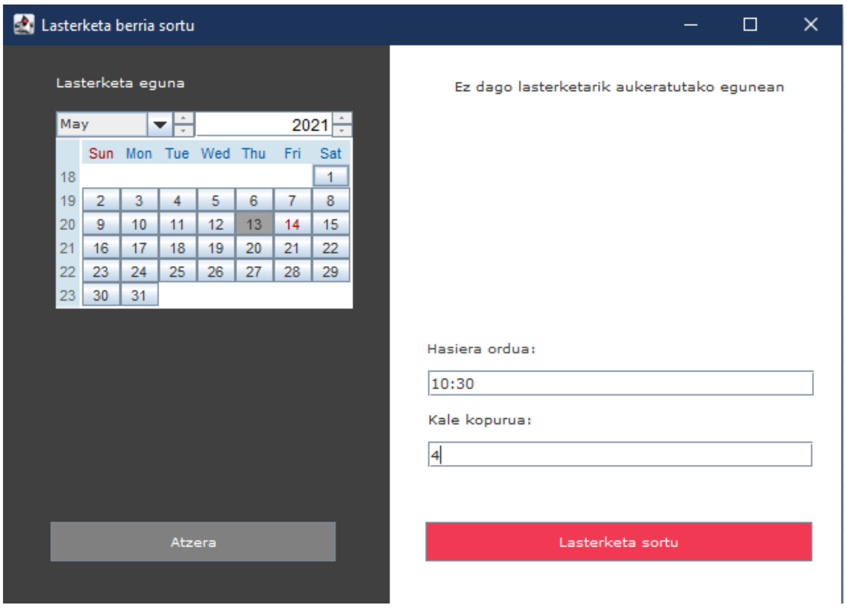


Dirua: 20.0 $



```
# Flow of events

## Basic Flow

1. *System* displays *ClientGUI* menu
2. *Client* selects *delete acount* button
3. *System* displays *LoginGUI* menu and removes client from the data base

## Alternative flow

1. You have money in the acount
2. You have a pending bet
```

CREATE RACE



Lasterketa eguna

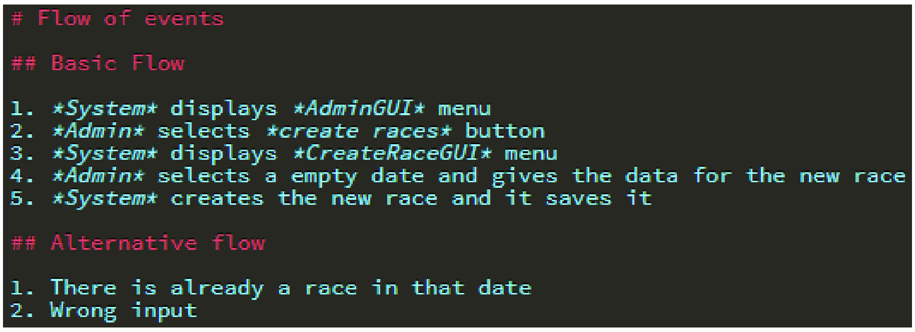
May 2021

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

Ez dago lasterketarik aukeratutako egunean

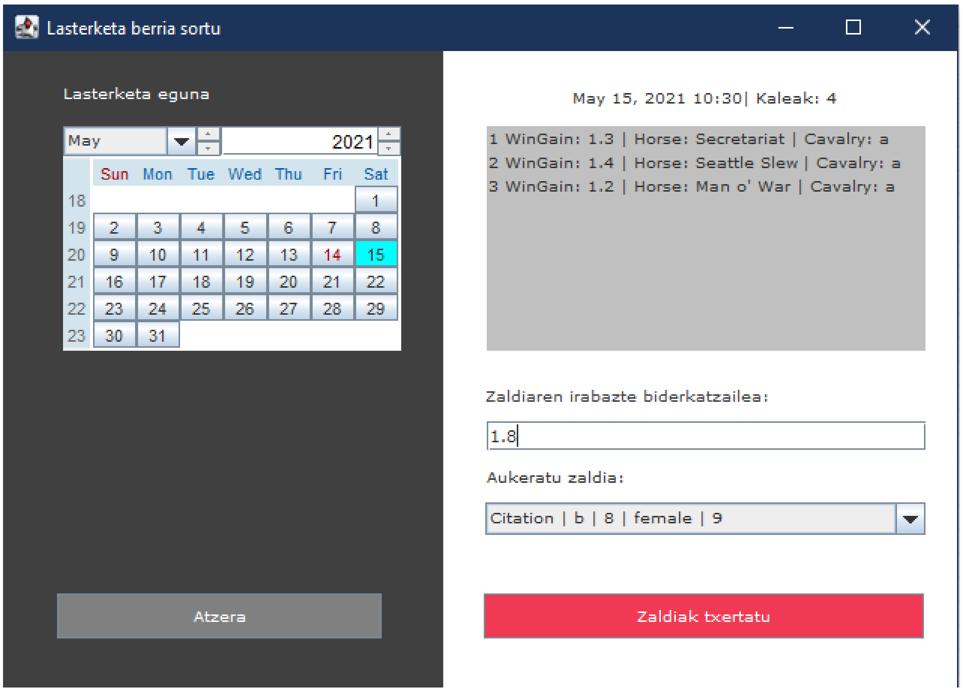
Hasiera ordua: 10:30

Kale kopurua: 4



```
# Flow of events

## Basic Flow

1. *System* displays *AdminGUI* menu
2. *Admin* selects *create races* button
3. *System* displays *CreateRaceGUI* menu
4. *Admin* selects a empty date and gives the data for the new race
5. *System* creates the new race and it saves it

## Alternative flow

1. There is already a race in that date
2. Wrong input
```

ADD HORSES



Lasterketa berria sortu

Lasterketa eguna

May 2021

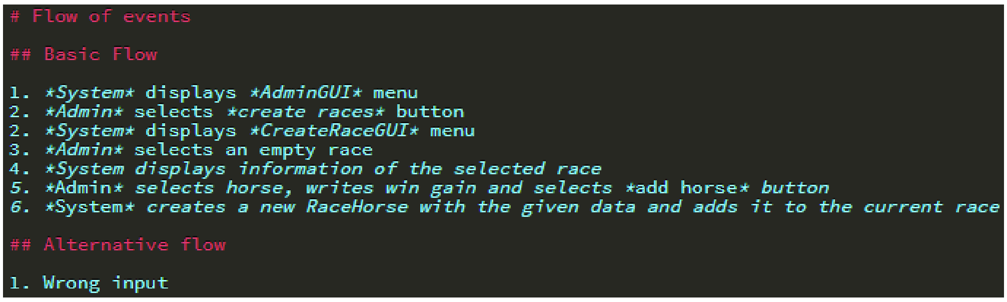
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18							1
19	2	3	4	5	6	7	8
20	9	10	11	12	13	14	15
21	16	17	18	19	20	21	22
22	23	24	25	26	27	28	29
23	30	31					

May 15, 2021 10:30| Kaleak: 4

```
1 WinGain: 1.3 | Horse: Secretariat | Cavalry: a
2 WinGain: 1.4 | Horse: Seattle Slew | Cavalry: a
3 WinGain: 1.2 | Horse: Man o' War | Cavalry: a
```

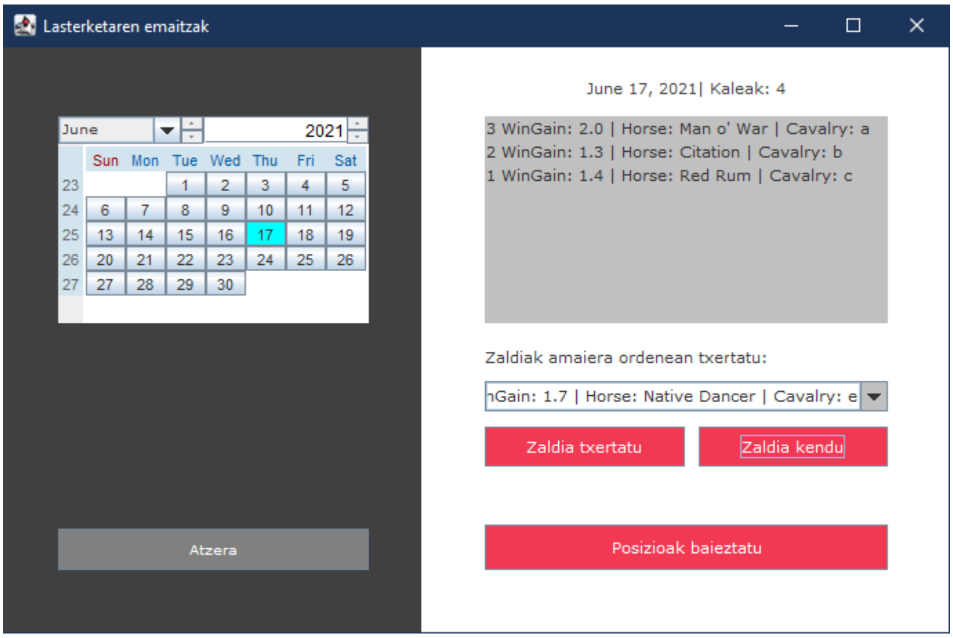
Zaldiaren irabazte biderkatzailea: 1.8

Aukeratu zaldia: Citation | b | 8 | female | 9



```
# Flow of events

## Basic Flow

1. *System* displays *AdminGUI* menu
2. *Admin* selects *create races* button
2. *System* displays *CreateRaceGUI* menu
3. *Admin* selects an empty race
4. *System displays information of the selected race
5. *Admin* selects horse, writes win gain and selects *add horse* button
6. *System* creates a new RaceHorse with the given data and adds it to the current race

## Alternative flow

1. Wrong input
```

SET RESULTS



Lasterketaren emaitzak

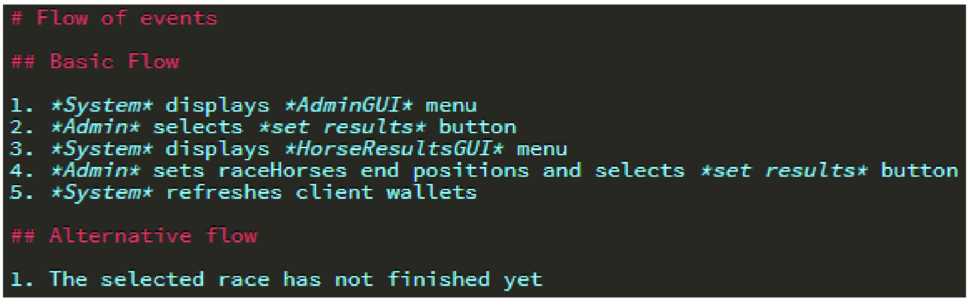
June 2021

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

June 17, 2021| Kaleak: 4

```
3 WinGain: 2.0 | Horse: Man o' War | Cavalry: a
2 WinGain: 1.3 | Horse: Citation | Cavalry: b
1 WinGain: 1.4 | Horse: Red Rum | Cavalry: c
```

Zaldiak amaiera ordenean txertatu: ıGain: 1.7 | Horse: Native Dancer | Cavalry: e



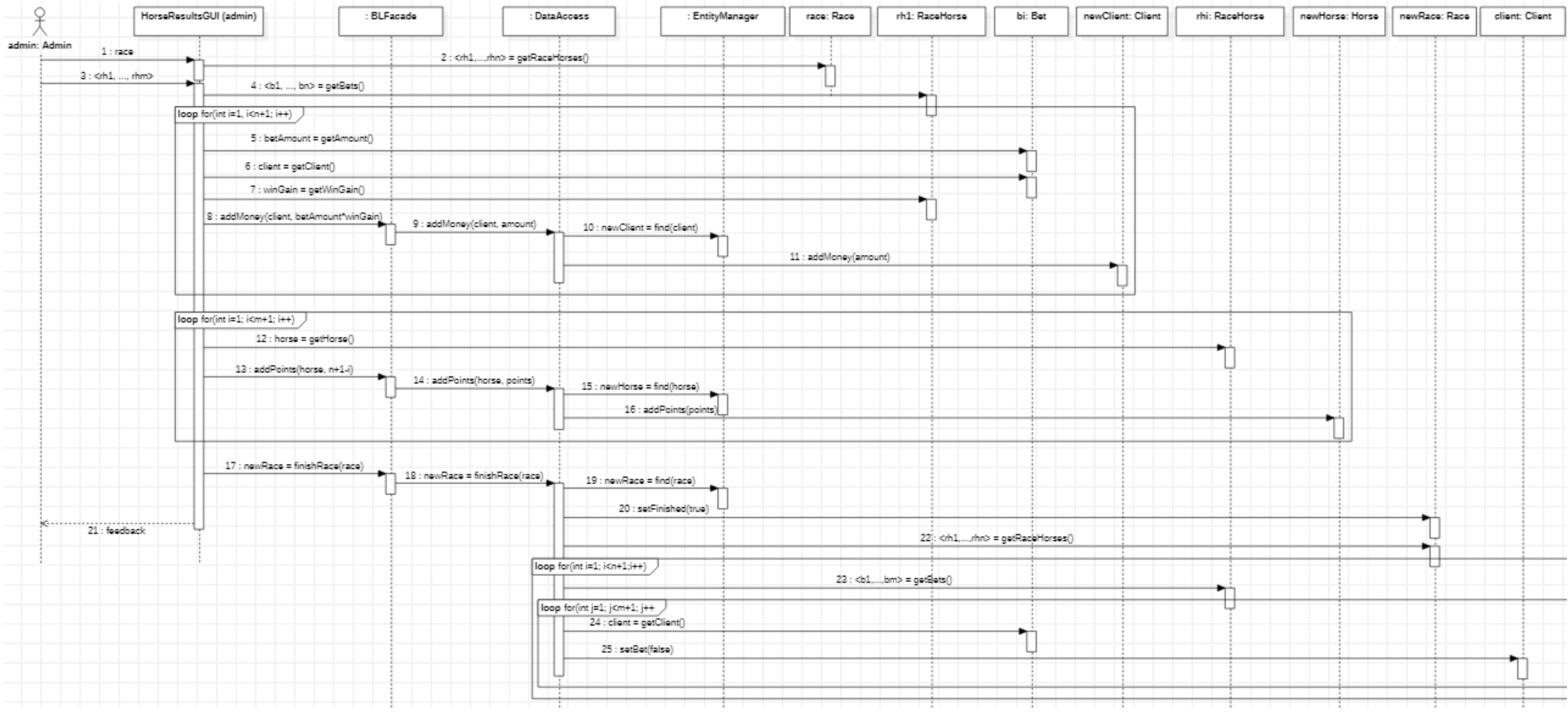
```
# Flow of events

## Basic Flow

1. *System* displays *AdminGUI* menu
2. *Admin* selects *set results* button
3. *System* displays *HorseResultsGUI* menu
4. *Admin* sets raceHorses end positions and selects *set results* button
5. *System* refreshes client wallets

## Alternative flow

1. The selected race has not finished yet
```

5. Diseinua:

SET RESULTS

Pausuak:

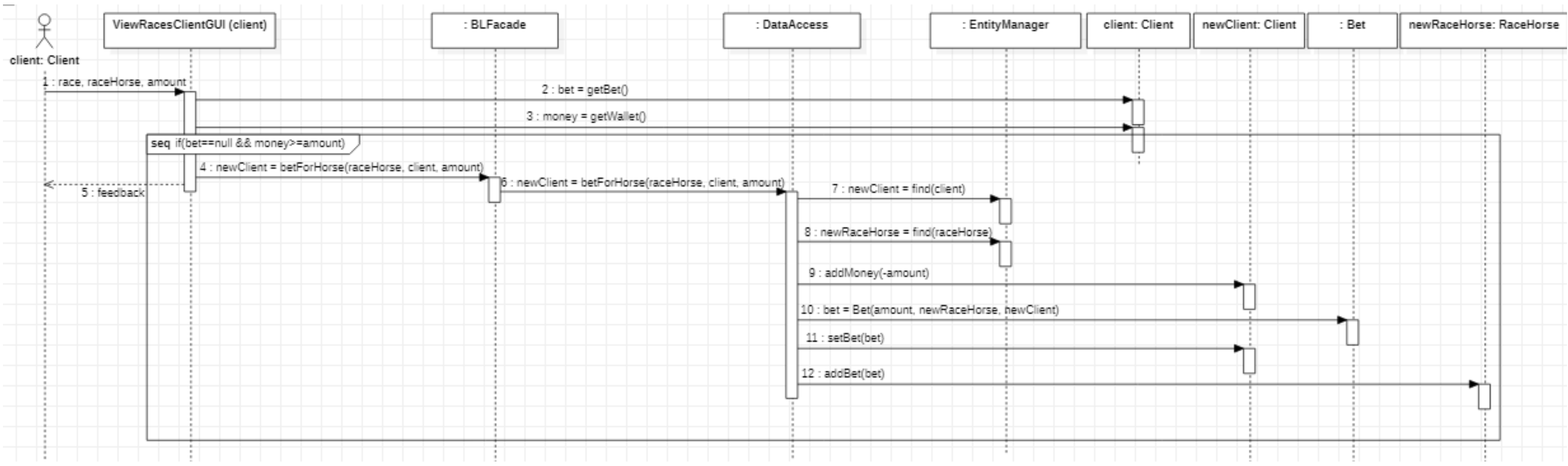
1. Lasterketako zaldiak lortu
2. Zaldiak bukatu duten ordenean pasa
3. Zaldi irabazleari egindako apustuen arabera dagokien bezeroen kontuak eguneratu
4. Lasterketako zaldien puntuazioa gehitu bukatu duten posizioaren arabera
5. Lasterketa amaitutzat ezarri


admin: Admin
HorseResultsGUI (admin)
: BLFacade
: DataAccess
: EntityManager
race: Race
rh1: RaceHorse
bi: Bet
newClient: Client
rhi: RaceHorse
newHorse: Horse
newRace: Race
client: Client
1 : race
2 : <rh1,...,rhn> = getRaceHorses()
3 : <rh1, ..., rhm>
4 : <b1, ..., bn> = getBets()
loop for(int i=1, i<n+1; i++)
5 : betAmount = getAmount()
6 : client = getClient()
7 : winGain = getWinGain()
8 : addMoney(client, betAmount*winGain)
9 : addMoney(client, amount)
10 : newClient = find(client)
11 : addMoney(amount)
loop for(int i=1; i<m+1; i++)
12 : horse = getHorse()
13 : addPoints(horse, n+1-i)
14 : addPoints(horse, points)
15 : newHorse = find(horse)
16 : addPoints(points)
17 : newRace = finishRace(race)
18 : newRace = finishRace(race)
19 : newRace = find(race)
20 : setFinished(true)
21 : feedback
22 : <rh1,...,rhn> = getRaceHorses()
loop for(int i=1; i<n+1;i++)
23 : <b1,...,bm> = getBets()
loop for(int j=1; j<m+1; j++
24 : client = getClient()
25 : setBet(false)

BET

Pausuak:

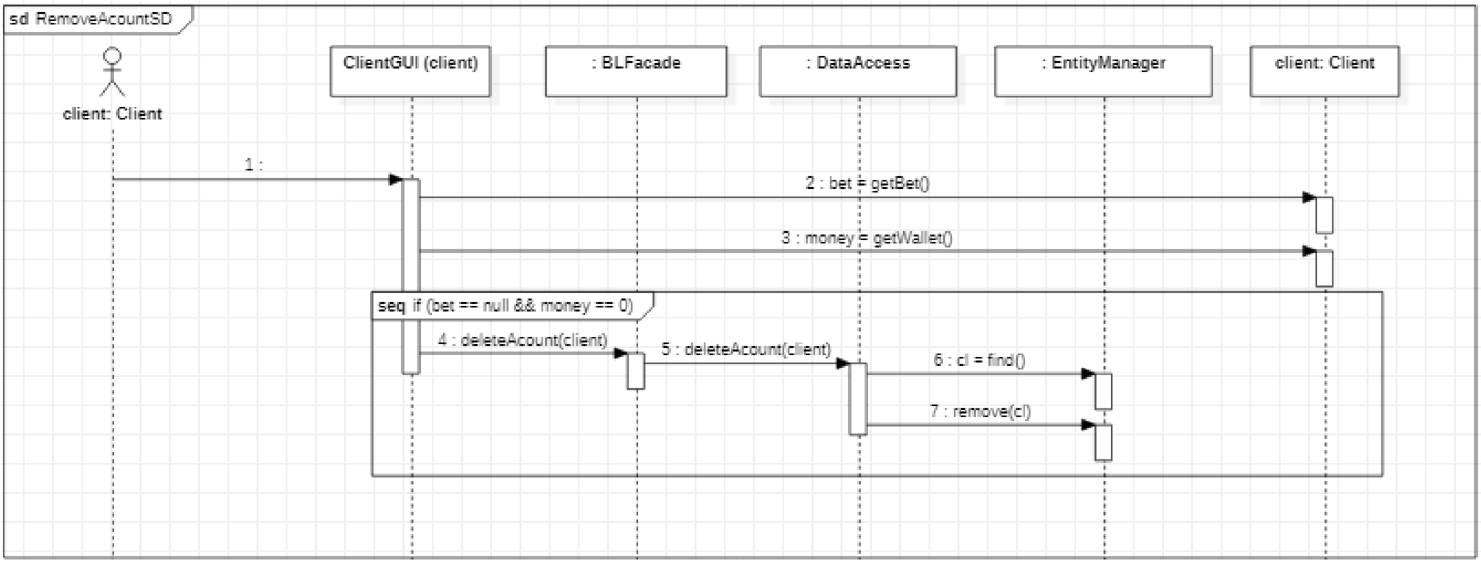
1. Apustu egiteko zaldia eta diru kopurua pasa
2. Bezeroak bukatu gabeko apusturik ez duela eta apustu egiteko nahikoa diru duela ikusi.
3. Apustua sortu pasatako datuekin

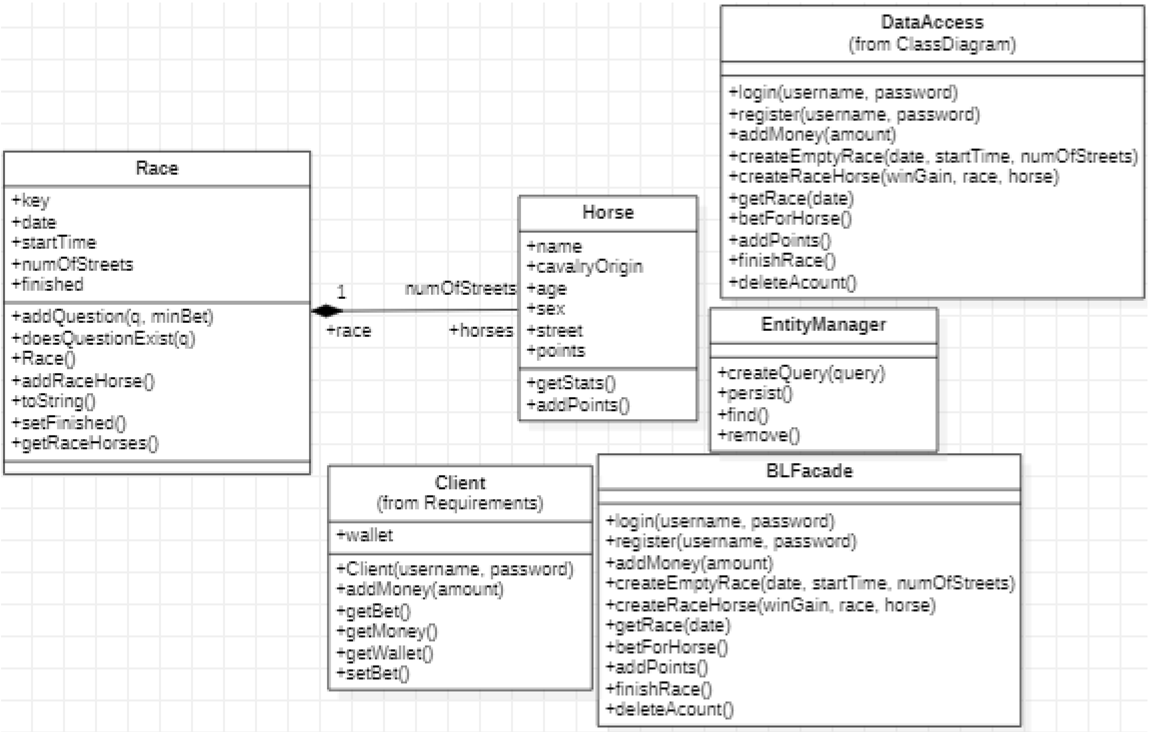

client: Client
ViewRacesClientGUI (client)
: BLFacade
: DataAccess
: EntityManager
client: Client
newClient: Client
: Bet
newRaceHorse: RaceHorse
1 : race, raceHorse, amount
2 : bet = getBet()
3 : money = getWallet()
seq if(bet==null && money>=amount)
4 : newClient = betForHorse(raceHorse, client, amount)
5 : feedback
6 : newClient = betForHorse(raceHorse, client, amount)
7 : newClient = find(client)
8 : newRaceHorse = find(raceHorse)
9 : addMoney(-amount)
10 : bet = Bet(amount, newRaceHorse, newClient)
11 : setBet(bet)
12 : addBet(bet)

REMOVE ACOUNT

Pausuak:

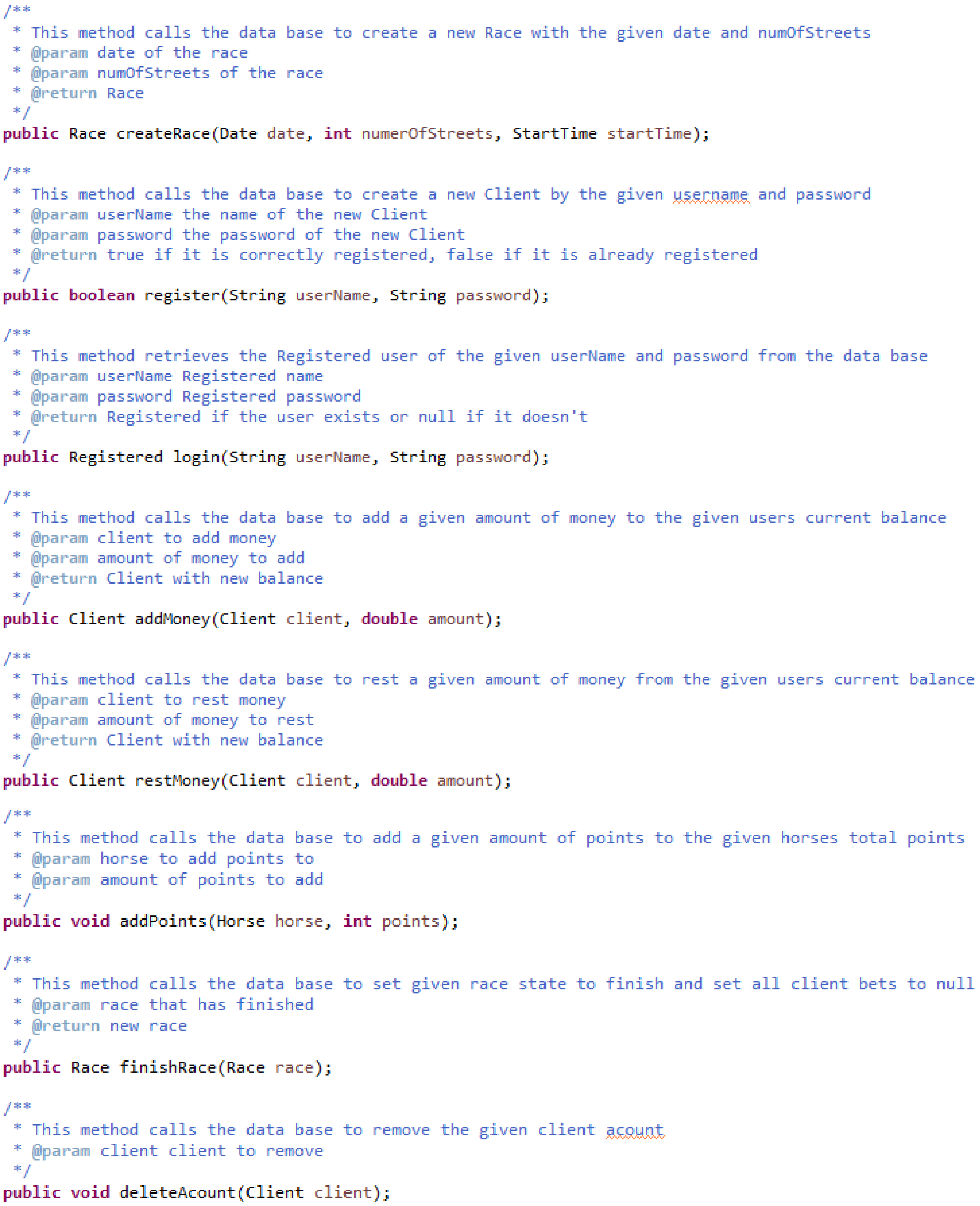
1. Bezeroak bukatu gabeko apusturik eta dirurik ez duela ikusi
2. Bezeroa ezabatu datu basetik





6. Inplementazioa:

```
/**
 * This method calls the data base to create a new bet
 * @param raceHorse of the new Bet
 * @param client of the new Bet
 * @param amount bet of the new Bet
 * @return Client with actualized data
 */
public Client betForHorse(RaceHorse raceHorse, Client client, double amount);

/**
 * This method calls the data base to create a raceHorse and add it to the given race
 * @param winGain multiplier of the new RaceHorse
 * @param race of the new RaceHorse
 * @param horse of the new RaceHorse
 * @return RaceHorse
 */
public RaceHorse createRaceHorse(double winGain, Race race, Horse horse);

/**
 * This method retrieves all raceHorses of a given race from the data base
 * @param race from which to take the raceHorses
 * @return List<RaceHorse> raceHorses of the given race
 */
public List<RaceHorse> getRaceHorses(Race race);

/**
 * This method retrieves all horses from the data base
 * @return List<Horse> all horses of the DB in a list
 */
public List<Horse> getHorses();

/**
 * This method retrieves the events of a given date from the data base
 * @param date in which events are retrieved
 * @return collection of events
 */
public Race getRace(Date date);
```



```
/**
 * This method calls the data base to create a new Race with the given date and numOfStreets
 * @param date of the race
 * @param numOfStreets of the race
 * @return Race
 */
public Race createRace(Date date, int numerOfStreets, StartTime startTime);

/**
 * This method calls the data base to create a new Client by the given username and password
 * @param userName the name of the new Client
 * @param password the password of the new Client
 * @return true if it is correctly registered, false if it is already registered
 */
public boolean register(String userName, String password);

/**
 * This method retrieves the Registered user of the given userName and password from the data base
 * @param userName Registered name
 * @param password Registered password
 * @return Registered if the user exists or null if it doesn't
 */
public Registered login(String userName, String password);

/**
 * This method calls the data base to add a given amount of money to the given users current balance
 * @param client to add money
 * @param amount of money to add
 * @return Client with new balance
 */
public Client addMoney(Client client, double amount);

/**
 * This method calls the data base to rest a given amount of money from the given users current balance
 * @param client to rest money
 * @param amount of money to rest
 * @return Client with new balance
 */
public Client restMoney(Client client, double amount);

/**
 * This method calls the data base to add a given amount of points to the given horses total points
 * @param horse to add points to
 * @param amount of points to add
 */
public void addPoints(Horse horse, int points);

/**
 * This method calls the data base to set given race state to finish and set all client bets to null
 * @param race that has finished
 * @return new race
 */
public Race finishRace(Race race);

/**
 * This method calls the data base to remove the given client acount
 * @param client client to remove
 */
public void deleteAcount(Client client);
```

7. Ondorioak:

Aurkezpenean aipatu nuen bezala, irakasgaia planteatzen den modua egokia iruditzen zait. Nire ustez praktikoki lan egitea modu oberena da software ingenieri batek behar dituen oinarrizko baliabideak eta metodologiak erabiltzen ikasteko.

Horretaz aparte, lan talde dinamika ere egokia iruditzen zait, normalean enpresetan talde lana ere egiten delako. Dena dela, nire ustez obeto egongo litzateke git baliabideak lehenago lantzea laborategietan, eta baliabide horiek orokorragoak izatea, gitHubDesktop eta honakoak bezalakoak.

8. Bideoa: ***DEMOA***

9. Kodea: ***GITLAB***