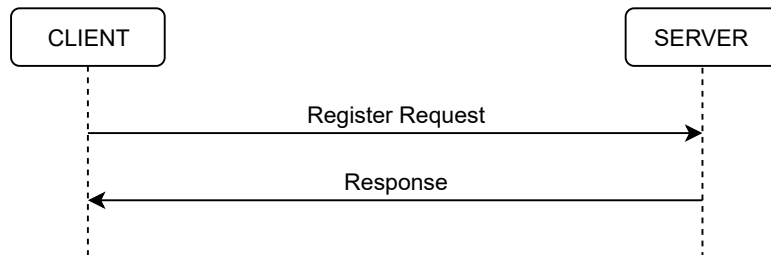


Scenarios

i. Client registers into app

- Server needs to check if username is unique.
- Register doesn't connect user.



1. Register (Request)

- username : Name that identifies user
- password : User's Password

TYPE = 1	username	password
----------	----------	----------

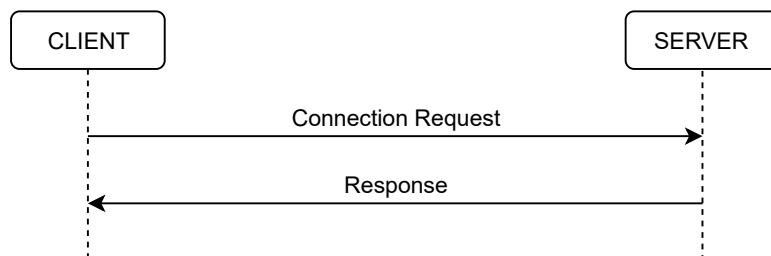
0. Response

- successFlag : If the operation was successful or not
- message : Additional information

TYPE = 0	successFlag	message
----------	-------------	---------

ii. Client logs in

- Server needs to make sure username and password are correct.
- If successful, server sends the port for connection.
- If unsuccessful, server sends an error message.



2. Connection (Request)

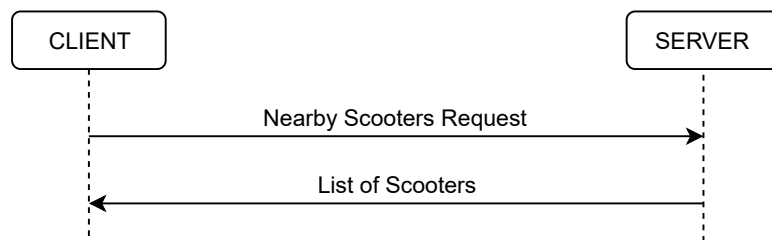
- username : Name that identifies user
- password : User's Password

TYPE = 2	username	password
----------	----------	----------

iii. Client logs out

iv. Client asks to see all scooters nearby

- Server sends a list of scooters and their locations



3. Nearby Scooters (Request)

- location : where the user is

TYPE = 3	x	y
----------	---	---

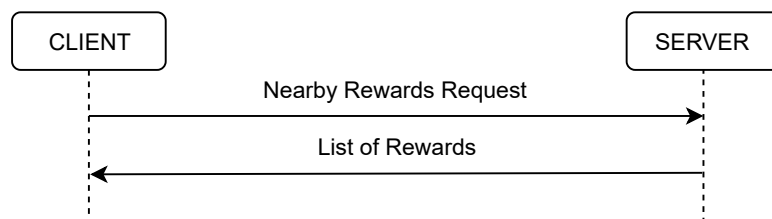
4. List of Scooters (Response)

- numberOfScooters : How many scooters the list has
- location : Where the scooter is

TYPE = 4	numberOfScooters	x	y	...
----------	------------------	---	---	-----

v. Client asks to see all rewards nearby

- Server sends a list of rewards nearby



5. Nearby Rewards (Request)

- location : where the user is

TYPE = 5	x	y
----------	---	---

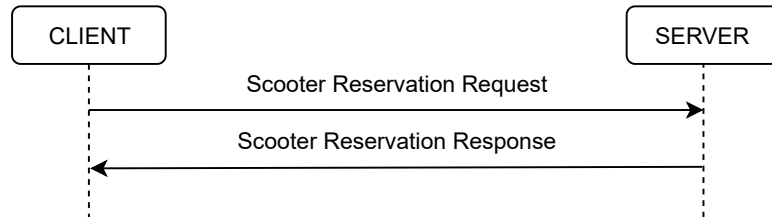
6. List of Rewards (Response)

- numberOfScooters : How many rewards the list has
- originLocation : Where to start trip
- destLocation : Where to end trip

TYPE = 6	numberOfRewards	x	y	x	y	...
----------	-----------------	---	---	---	---	-----

vi. Client asks for scooter reservation

- Clients asks to reserve a scooter
- Server creates a reservation for a scooter, adds it to the list of reservations and sends the reservation code to the client.
- If client doesnt use the reservation code in X time, the server will discard the reservation.



7. Scooter Reservation (Request)

- location : where the user is

TYPE = 7	x	y
----------	---	---

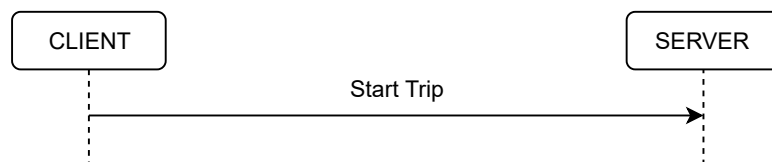
8. Scooter Reservation (Response)

- reservationCode : Code that user will use to start the trip(can be UNSUCCESSFUL)
- location : Where the scooter is.

TYPE = 8	reservationCode	X	Y
----------	-----------------	---	---

vii. Client starts trip

- Client will use the code to start the trip



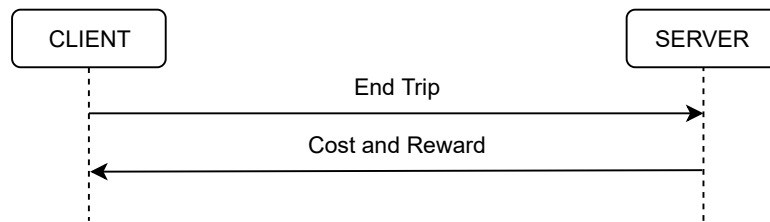
9. Start Trip

- location : where the user is

TYPE = 9	x	y
----------	---	---

viii. Client ends trip

- Client sends reservation code and the server will calculate total distance and total time, generating the cost and reward of the trip.
- Server takes reservation out of list of reservations and updates a new reward list.



10. End Trip

- `reservationCode` : id of reservation where all the information is saved.
- `location` : Final location of client.

TYPE = 10	reservationCode	x	y
-----------	-----------------	---	---

11. Cost and Reward (Response)

- `cost` : How much the client will pay
- `reward` : How much the client saved??

TYPE = 11	cost	reward
-----------	------	--------

ix. Client turns on notifications for rewards nearby

x. Client turns off notifications for rewards nearby