**Rendelések kezelése:** pipa

Szükséges új osztályok (szerverhez + amúgy) – már implementálva, ha valami rossz vagy máshogy kell módosítsd. Pirossal az új szükséges függvények a szervernél.

1. Order: ebben van minden adat egy darab rendelésről
   1. 5 új státusz kell ami leírja az állapotot, most jelenleg: 0 = új rendelés, 1 = fogadott rendelés (kaja már készül), 2 = kiszállításra kész (már csak kikell rendelni futárnak), 3 = Kiszállítás alatt, 4 = Kiszállítva
   2. plusz a rendelés dátum – idejét is elkellene tárolni
2. **OrderList**: a szerver egy ilyen OrderList típust ad vissza, ami a rendelések listája (orderList = List<Order>) – mint a FoodList-eknél csináltuk.
   1. Hívása pl: getOrders(restID) ahol restID az adott étterem id. pipa

Ami kell még fv: (nem kell visszatérési érték) pipa

* amiben a rendelés állapotát módosítom pl.: updateOrderState(OrderID, newState), ahol orderID a rendelés ID-ja, newState pedig egy int az új állapotról.

**Futárok kezelése:**

Szükséges új osztályok:

1. DeliveryBoy: ebben van minden adat egy futárról (most jelenleg az id-ja meg a neve meg a hozzá tartozó rendelések listája)
2. **DeliveryBoyList**: a szerver ilyen típust ad vissza ami a futárok listája (deliveryBoyList = List<DeliveryBoy>). Hívása pl: getDeliveryBoys()

Szükséges függvények szerveren: (nem kell visszatérési érték)

* rendelés hozzáadása futárhoz: pl: addOrderToDeliveryBoy(boyID, OrderID) – boyID a futár ID, orderID meg a rendelés ID.
* rendelés visszavonása futártól / amikor a futár utasítja vissza a rendelést arra is jó:
  + removeOrderFromDeliveryBoy(boyID, orderID)
* Új tábla (deliveryboyID – orderID)

Megdumálni:

* Új rendelésről értesítés? (most jelenleg másik thread figyel 10 mp-ként) pipa
* Kiszállításási címet beírni az orderek kezelésénél (amikor futárhoz van rendelve) pipa
* Futárok jelentkeznek arra, hogy melyik étteremnek szállítanak ki?
* A fentiek alapján nem lenne szükséges hogy az Order eltárolja a deliveryPersonID-t (legalábbis nálam), mert adott futárnál van az összes rendelés listája amivel dolga volt
* Várható kiszállítási idő megadása? Hol? Hogyan?
* Időre: tartózkodási hely, elkészülési idő, futár kiérkezésének, összes rendelés darabszámának figyelembe vétele
* Beokoskodni a rendszerbe hogy figyeljük a rendelés idejét és ha új érkezik, akkor napszaktól függően lesz (pl.: alap 40 perc, (11:30 – 13:30 és 18:00-20:00): + 15 perc)

Több kliens tud e szerverre csatlakozni?

Nyitvatartási idő étteremnek?

Étterem: TODO

* szabályozni a rendelések fogadását
* akciók kezelése
* adott időszakra milyen megrendelések voltak – és mik a költségek (esetleg bevételek)
* Megjeleníteni a **szállítási címet**
  + vagy a getOrder() szerveres fv adja vissza a customer mellé a szállítási címet (ez jobbnak tűnik)
  + vagy kell új fv amiben visszakapok customer id (username) által egy addresst
* Személyes átvétel?

**Költségek? : futárok fizuja?**

Legyen mondjuk úgy, hogy minden rendelésre (ami kiszállításos) 1000 ft-ot fizetünk az online ételrendelés miatt

**Akciók kezelése?**

külön attribútum ami jelzi h akciós? vagy külön ár – illetve akciós ár tárolva.

Plusz attribútum legyen: price mellett egy discountPrice. Alapesetben **1**, ha pedig kivan töltve akkor tudjuk h discounted.

setDiscount(FoodID, double percentage) – foodID alapján megváltoztatja az étel árát

* plusz attribútum: discountPercentage double (0.2 – csökkentés, 1.2 – növelés)
* Marci: vendég kliensen ár megjelenítése: price\*discountPercent (ár\*0.8)

**Tervezett szállítás!!**

Beokoskodni a rendszerbe, hogy figyeljük a rendelés idejét és ha új érkezik, akkor napszaktól függően lesz (pl.: alap 40 perc, (11:30 – 13:30 és 18:00-20:00): + 15 perc)

* + új attribútum kell -> tervezett szállítás Order-be is illetve az étteremnek is.
  + registerRestaurant fv. kiegészítése deliveryTime-al

setApproximateDeliveryTime(OrderID, RestID)

SZERVER: csinálja az idő állítást a bejövő OrderID és RestaurantID alapján.

Vendég kliens, beállítja az alap kiszállítási időt az étterem adata alapján (+ figyeli a napszakot és annak megfelelően hozzáad még ha kell). Tehát a paraméter: jelenlegi idő + az étterem által megadott perc (+ az időszaki plusz idő)

Étterem kliensnél a futárhoz való rendelésnél is meghívódik ott beállítunk kb +10 percet a jelenlegihez

MARCI

Étel elérhetőségét is figyeld amikor megjeleníted a vendég kliensén

Étel rendelésnél megadni a jelenlegi időt.

Mivel rendelni két féle képpen lehet, regisztrált és nem regisztrált felhasználóként, ezért erre kell egy megoldás. Vázolom az én ötletem.

**Regisztrált user** esetén a fgvhez kell :

* -étterem ID (ahonnan rendel)
* -rendelő userneve
* -kaják ID-ja vesszővel elválasztva pl "5,3,2,8"
* -fizetendő összeg

**Regisztrálatlan user** esetén, szerintem legegyszerűbb, ha ugye minden, amit felsoroltam kell, a username viszont "" lesz. Mivel nem regisztrált, ezért még a fenti adatokon kívül kell ilyen, hogy

* név (vezeték keresztnév), cím (irányítószám, város, line1 (Stadion utca 8.), line2 (2. emelet 3 ajtó)), telefonszám és akkor azokat is hozzáteszed még
* VENDEG KLIENS küldjön egy üres username-t!!!
* SZERVER: kezeli a regisztrálatlan user hiányos adatait
* Étterem: username = guestX (ne jelenítse meg)

Étterem:

vezetéknél, keresztnév, telefonszám, cím 🡪 futárhoz rendelésnél

**Étel módosítása:**

updateFood(Food food) – már létező food-ot kap meg módosított adatokkal. Adatbázisban kell módosítani adott foodID nál az adatokat

Order még nem lett kiegészítve a User paraméterrel.

Szerver akkor is 1.0-t ad vissza adatbázisból amikor már értéke van.

**Kategória módosítása**:

* Opció az étterem kategória nevének a módosítására.
* Függvény: modifyCategory(categoryID, newCategoryName, resturantID) 🡪 megváltoztatja a categoryID-hoz tartozó categoryName-et a paraméterben kapottra