**KiwiClub webalkalmazás**

**Fejlesztők:**

Solymosi Richárd

Szabó Tamás

Lauber Attila

Tartalom

[A webalkalmazás specifikációja 3](#_Toc122099578)

[Domain osztályok: 3](#_Toc122099579)

[Service osztályok: 7](#_Toc122099580)

[Perzisztencia réteg: 8](#_Toc122099581)

[Kontrollerek: 8](#_Toc122099582)

[Adat transzfer objektumok: 9](#_Toc122099583)

[View réteg: 10](#_Toc122099584)

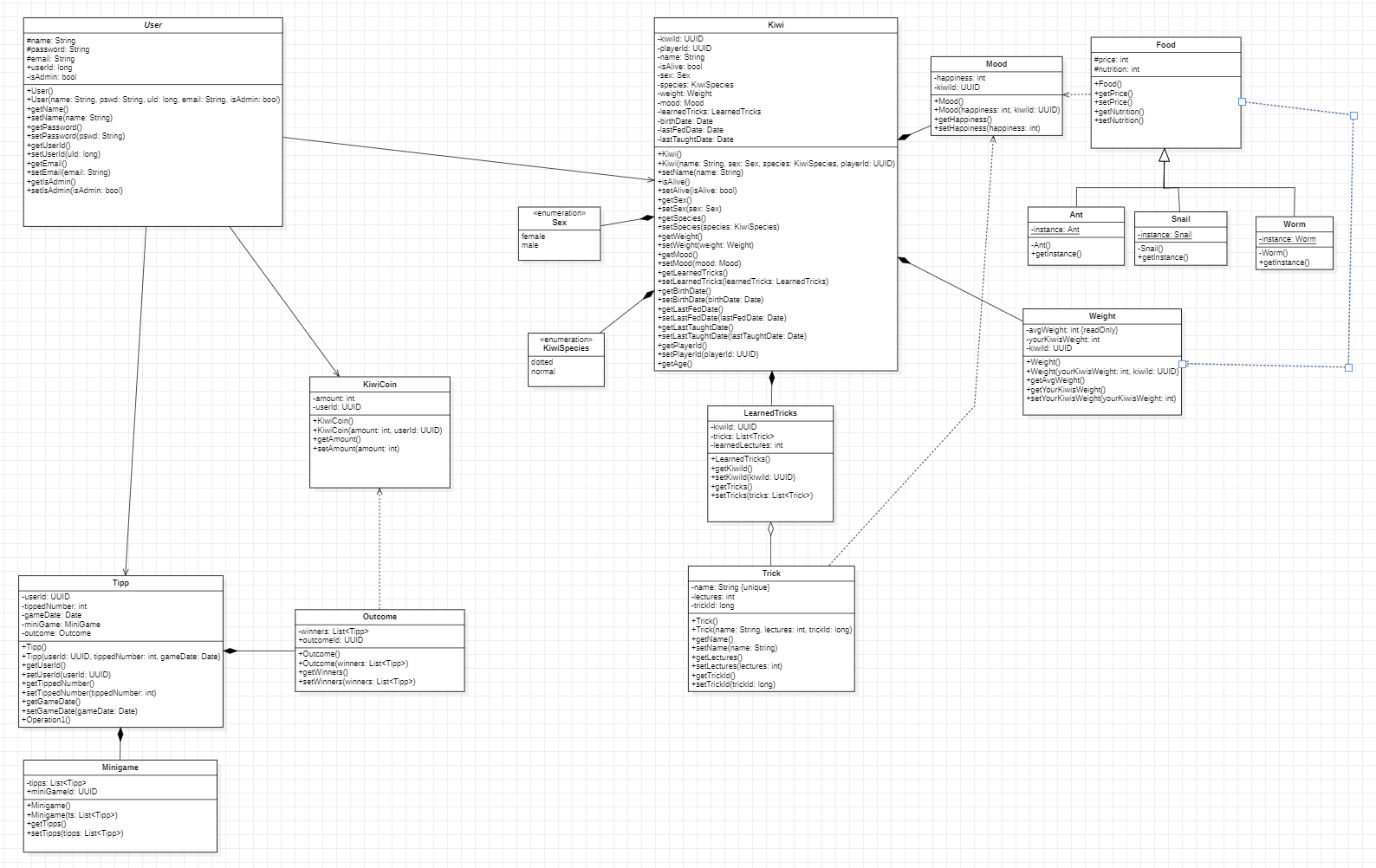
Az alkalmazás összegzése

Az applikáció lényege, hogy a felhasználó a webböngészőn keresztül gondozhatja a saját kiwijét. A játékos a regisztráció során személyre szabhatja, és nevet adhat a kisállatának. Az oldalakon lehetőség van a kiwi etetésére, itatására, valamin tanítására. Ügyelni kell arra, hogy jókedvű, és egészséges legyen a madarunk. A házikedvenc ellátása pénzbe kerül, így egy bizonyos mennyiségű összeg minden játékos rendelkezésére áll, azonban minijáték formájában van lehetőség extra pénz szerzésére is.

A webalkalmazás specifikációja

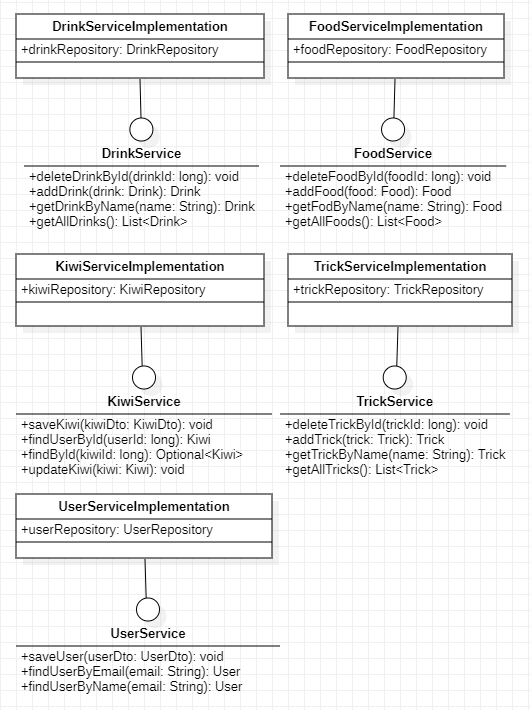
# A webalkalmazás specifikációja

## Domain osztályok:

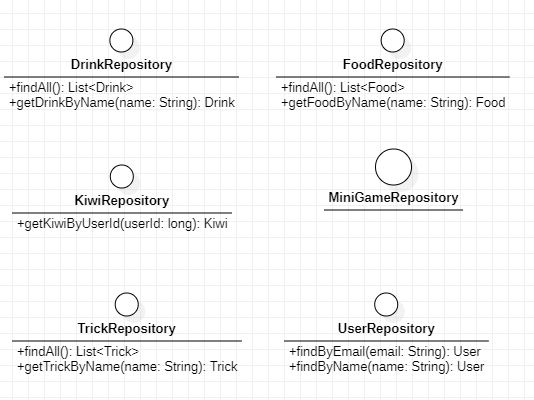


* User osztály
  + osztály, amelyben a játékos adatait tároljuk
  + mezők:
    - #name: a felhasználó neve
    - #password: a felhasználó jelszava
    - #email: a felhasználó email címe
    - +userId: a felhasználó egyedi azonosítója
    - -isAdmin: visszaadja, hogy az adott felhasználó admin-e
  + metódusok
    - minden mezőhöz setter,getter
  + konstruktorok
    - +User()
    - +User(name: String, password: String, userId: long, email: String, isAdmin: bool)
* Kiwi osztály
  + A játékos kiwije
  + mezők
    - -learnedTricks: A kiwi megtanult trükkjeit tartalmazó lista
    - -name: A kiwi neve
    - -sex: A kiwi neme
    - -species: A kiwi fajtája
    - -weight: A kiwi súlya, amely a tápláltságtól függ
    - -mood: A kiwi hangulata, amely a tápláltságtól, és a napi tanulástól függ
    - -isAlive: A kiwi életben van-e
    - -kiwiId: A kiwi egyedi azonosítója
    - -playerid: A kiwi gazdájának egyedi azonosítója
    - -lastFedDate: Date a kiwi legutolsó etetésének dátuma
    - -lastTaughtDate: Date a kiwi legutolsó tanításának dátuma
    - -birthDate: A kiwi létrehozásának dátuma
  + metódusok
    - minden mezőhöz setter, getter
  + konstruktorok
    - +Kiwi()
    - +Kiwi(name: String, sex: Sex, type: KiwiType, kiwiId: int)
* LearnedTricks osztály
  + Egy adott id-jú kiwi megtanult trükkjeit leíró osztály
  + mezők
    - -kiwiId: A kiwi egyedi azonosítója
    - -tricks: A kiwiId id-jú kisállat megtanult trükkjeit tartalmazza
  + metódusok
    - minden mezőhöz setter, getter
  + konstruktorok
    - +LearnedTricks()
* Trick osztály
  + Trükk, amit a kiwi megtanulhat
  + mezők
    - -name: A trükk neve
    - -lectures: Ennyi leckéből áll a trükk
    - -trickId: A trükk egyedi azonosítója
  + metódusok
    - minden mezőhöz setter, getter
  + konstruktorok
    - +Trick()
* Sex enum
  + A kiwi neme, a felületen vizuálisan megjelenik
* KiwiType enum
  + A kiwi fajtája, a felületen vizuálisan megjelenik
* Mood osztály
  + A kiwi kedvét mutató osztály, amely befolyásolja a napi tanulást, és azt az etetés
  + mezők
    - -happines: A kiwi kedvét reprezentálja (1-100 közötti érték)
    - -kiwiId: A kiwi egyedi azonosítója
  + metódusok
    - A happines mezőhöz setter, getter
  + konstruktorok
    - +Mood()
    - +Mood(happines: int)
* Food osztály
  + absztrakt osztály amelyből a különféle táplálék osztályok származnak
  + mezők
    - #price: Az étel ára
    - #nutrition: az étel tárpértéke
  + metódusok
    - get-set metódusok a price és a nutrition mezőkhöz
  + konstruktorok
    - +Food()
* Ant osztály
  + A Food-ból származó singleton osztály
  + mezők
    - -instance az osztály egyetlen példányát tartalmazza, mivel az osztály singleton
  + metódusok
    - +Instance() az osztály egyszeri példányosításáról gondoskodik
  + konstruktorok
    - -Ant()
* Snail osztály
  + A Food-ból származó singleton osztály
  + mezők
    - -instance az osztály egyetlen példányát tartalmazza, mivel az osztály singleton
  + metódusok
    - +Instance() az osztály egyszeri példányosításáról gondoskodik
  + konstruktorok
    - -Snail()
* Worm osztály
  + A Food-ból származó singleton osztály
  + mezők
    - -instance az osztály egyetlen példányát tartalmazza, mivel az osztály singleton
  + metódusok
    - +Instance() az osztály egyszeri példányosításáról gondoskodik
  + konstruktorok
    - -Snail()
* Weight osztály
  + A Kiwi osztályban jelenik meg mint adattag, minden kiwinek van különsúlya, amit százalékos formában lesz értelmezve, ezért az átlag súly adattag
  + mezők
    - -avgWeight: egy átlag kiwi testsúlyát írja le (beégetett érték)
    - -yourKiwisWeight: a játékos kiwijének aktuális súlya
    - -kiwiId: A kiwi egyedi azonosítója
  + metódusok
    - get metódusok az adattagokhoz
    - setYourKiwisWeight(yourKiwisWeight: int)
* KiwiCoin osztály
  + Játékbeli pénz, amelynek mennyiségét befolyásolja az ételvásárlás, a minijáték
  + mezők
    - -amount: A pénz mennyisége
    - -userId: A felhasználó egyedi azonosítója
  + metódusok
    - az amount mezőhöz getter,setter
  + konstruktorok
    - +KiwiCoin()
    - +KiwiCoin(amount: int)
* MiniGame osztály
  + A számkitalálós játékot reprezentálja
  + mezők
    - -tipps: A különböző játékosoktól beérkezett tippek listája
    - +miniGameId: A minijáték egyedi azonosítója
  + metódusok
    - get-set metódusok a tipps-hez
  + konstruktorok
    - Minigame()
    - Minigame(ts:List<Tipp>)
* Outcome osztály
  + A minijáték nyerteseit tartalmazza, valamint befolyásolja a nyertes felhasználók pénzét
  + mezők
    - -winners: A minijáték nyertesei
    - +outcomeId: A kimenetel egyedi azonosítója
  + metódusok
    - get-set metódusok a winners-höz
  + konstruktorok
    - +Outcome()
    - +Outcome(ws:List<Tipp>)
* Tipp osztály
  + Egy játékostól beérkező tipp, amelyet a MiniGame osztály tárol el
  + mezők
    - -userId: A játékos azonosítója akitől a tipp érkezett
    - -tippedNumber: A tippelt szám
    - -gameDate: a játék dátuma
  + metódusok
    - get-set metódusok
  + konstruktorok
    - +Tipp()
    - +Tipp(uid:int,tn: int)

## Service osztályok:



Perzisztencia réteg:

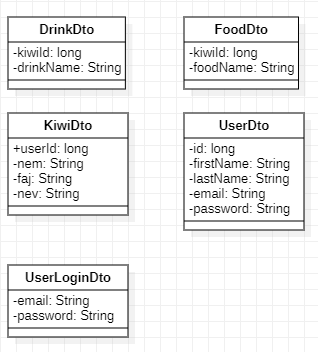


* DrinkRepository
  + Az adható italok listáját tároljuk el
* FoodRepositry
  + Az adható ételek listáját tároljuk el.
* KiwiRepository
  + A kiwik adatait tároljuk el.
* MiniGameRepository
  + Magát a minijátékot, a tippek listáját, és a kimenetelt tároljuk el.
* TrickRepository
  + A tanulható trükkök listáját tároljuk el.
* UserRepository
  + a felhasználók adatait tároljuk el.

## Kontrollerek:

* KiwiController:
  + Ez a kontroller felelős a kiwi létrehozásáért, etetésért, itatásáért, és tanításáért.
* TrickController:
  + Ez a kontroller felelős a Trükkök CRUD műveleteiért.
* UserController:
  + Ez a kontroller felelős a regisztrációért, és a loginért.

## Adat transzfer objektumok:



## View réteg:

|  |  |
| --- | --- |
| Oldalak, amiket bárki láthat bejelentkezés előtt: | |
| . ábra Bejelentkező oldal | . ábra Regisztrációs oldal |

|  |
| --- |
| Oldalak, amiket játékosok látnak: |
|  |