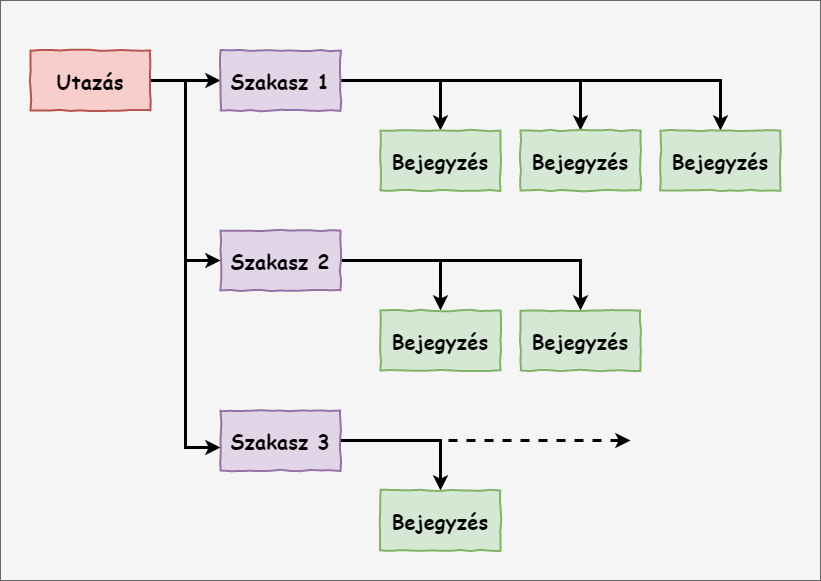
# DiceTravel Home felhasználói kézikönyv

## A DiceTravel koncepciója

A DiceTravel alapvetően egy naplózó alkalmazás, mely utazásokkal kapcsolatos eseményeket tárol, rendszerez és jelenít meg a felhasználó számára. A DiceTravel továbbá egy platformot nyújt, hogy a felhasználók megoszthassák egymással az élményeiket.

Minden utazást kisebb részekre lehet osztani, ezeket szakaszoknak nevezzük. Egy esemény, melyről bejegyzés készül, mindig egy adott szakasz alatt történik. Ezért az adott bejegyzés az adott szakasz alatt kerül rögzítésre.

Egy szakasz befejezése egy új szakasz indítását vagy az utazás végét indukálja. Mivel ez egy logoló, és nem pedig tervező alkalmazás, nem lehetséges egyszerre több utazáson, szakaszban részt venni. Ettől függetlenül egy felhasználónak tetszőleges számú lezárt utazása lehet, egy utazásnak tetszőleges számú szakasza.

A bejegyzéseink, szakaszaink és utazásaink láthatóságának három szintje van: Privát, Barátok, Publikus.

Baráti kapcsolatokat jelölés-visszaigazolás módszerrel lehet létesíteni, ami azt jelenti, hogy tetszőleges felhasználónak küldhetünk barátjelölést, viszont a barátság csak abban az esetben kerül aktiválásra, ha ezt a célfelhasználó elfogadja.

A felület egyik legfontosabb része, amit szeretnék már itt kiemelni: a Folyam.  
A Folyamban fognak megjelenni a különböző keresésekkel kapcsolatos találatok. Például a felhasználó utazásai, más felhasználók utazásai, vagy egy felhasználó barátai.

### Kulcsszavak

* **Utazás (Journey):** A DiceTravel utazással kapcsolatos legnagyobb logikai egysége, mely alatt a Szakaszok találhatóak. Egy kezdőponttal rendelkezik, valamint egy elnevezéssel.
* **Szakasz (Trip):** Az utazás elemei. Minden szakasz egy végponttal rendelkezik. A kezdőpont az előző szakasz végpontja határozza meg, vagy ha még nincs teljesített szakasz, akkor az Utazás kezdőpontja lesz.
* **Bejegyzés (Entry):** A DiceTravel utazással kapcsolatos legkisebb logikai egysége. Minden Bejegyzés egy szöveges bejegyzésbő, ami maximum 1024 karakter hosszú lehet, és/vagy egy képből áll. Utóbbi maximum 16 MB méretű lehet.
* **Folyam (Flow):** A különböző utazással/barátokkal kapcsolatos szűrések nézetei. Egy felhasználó Utazás folyama például az összes általa készített Utazást listázza ki és jeleníti meg. A következő folyamok vannak például a DiceTravelben: Utazás, Történet, Aktuális, Barát.
* **Barát (Friend):** Olyan felhasználó, akivel láthatjuk egymás csak barátoknak szánt bejegyzésit is.

Az alkalmazás adatai egy MySQL adatbázisban vannak tárolva.

## A DiceTravel felépítése

### Osztálytípusok

Szeretném általánosságban ismertetni, hogy a Dice Travel hogyan épül fel, milyen osztálytípusokkal doglozik.

#### Entitások

Az entitások az adatbázissal való kommunikációra lesznek kifejlesztve, valamint az adott tábla egy adott sorának az programbeli leképzései lesznek. A hozzá tartozó metódusokkal könnyen képesek leszünk az alapvető CRUD műveletek végrehajtására, valamint a fontos lekérdezések indítására.

A koncepció alapján az alkalmazás az alábbi entitásokkal rendelkezik:

|  |  |
| --- | --- |
| * **Utazás** * **Szakasz** * **Bejegyzés** * **Felhasználó** * **Barátság** | Ezek az entitások lesznek az alkalmazás adatbázisában tárolva, valamint ezek lesznek a programunk főbb osztályai, amik a megfelelő adatok manipulálására lesznek használva. Ezek mind az *Entity* absztrakt osztályból fognak származni.  Ezek egy közös *EntityClasses* mappába lesznek szervezve. |

#### Formok

A felhasználó különböző űrlapokon keresztül tud majd létrehozni/módosítani/törölni bizonyos entitásokat az adatbázisból. Ezek szintén külön a *Forms* mappába lesznek szervezve, azon belül is entitásonkénti almappába. Valójában ezek az űrlapok fognak lehetőséget adni az adott entitás metódusainak felügyelt hívásaira.

Az alkalmazásnak a *MainForm* űrlap osztálya fog keretet adni, ez szintén a Forms mappában fog szerepelni, közvetlenül az entitás almappák mellett.

#### Kontrollok

A kontroll egy entitásnak a folyamban való megjelenése. Segítségével a felhasználó az adott entitással kapcsolatos műveleteket végezhet. Például törölheti az entitást, vagy kontextustól függően megváltoztathatja a folyamot. Az űrlapokhoz hasonlóan itt is a megfelelő entitások ellenőrzött hívásai fognak történni.

A kontrollok a Controls mappába lesznek szervezve.

#### Kiegészítő osztályok

A Util mappán belül a különböző kiegészítő osztályok szerepelnek. Ebből három van:

* *ActiveUserStore*: Ebben a statikus osztályban van letárolva az éppen bejelentkezett felhasználó, valamint ez az osztály felel a ki és bejelentkeztetésért.
* *FlowElementProvider*: Ez az osztály felel a folyam összeállításáért/tárolásáért/frissítéséért.
* *ImageHandler*: Ez a statikus osztály képes egy képet bináris formára hozni, hogy azt fel lehessen tölteni az adatbázisba.

## Az osztályok részletezése