**I.Database(tables):**

1.User:

* userID (int, nn, pk,ai)
* userName (varchar, nn)
* password (String, nn)
* score (int,nn)
* walidPassword (boolean, nn)
* role (ENUM, nn)

2.Location:

* locationID (int, nn, pk, ai)
* locationName (varchar, nn)
* loctionAddress (varchar, nn)
* rentFeeProHour (int, nn)

3.Game:

* gameID (int, nn, pk, ai)
* firstUserID (int, nn) (fk\_userTable - firstUserID - userID)
* secondUserID (int, nn) (fk\_userTable - secondUserID - userID)
* gameLocationID (int, nn) (fk\_locationTable – gameLocationID - locationID)
* firstUserScore (int, nn)
* secondUserScore (int, nn)

**II.Model:**

1.User:

* int userID
* String userName
* String password
* boolean validPassword
* Enum role
* int score

2.Location:

* int locationID
* String locationName
* String locationAddress
* int rentFeeProHour

3.Game:

* int gameID
* int firstUserID
* int firstUserScore
* int secondUserID
* int secondUserScore
* int gameLocationID

**III.Java program:**

1. Log page
   1. betöltődik a login oldal
   2. //localhost:8080/log
   3. login.html
2. Login page:
   1. itt várom a felhasználó nevet és jelszót
   2. @PostMapping, @RequestParam - String username, String password
   3. //localhost:8080/login?username=x?password=y
   4. login.html
3. Firstlogin page
   1. első bejelentkezéskor ide irányítjuk a felhasználót, és csak a jelszó megváltoztatása után tovább az index oldalra
   2. @PostMapping, @RequestParam – String username(ezt nem kérem be újra, automatikusan küldöm tovább a korábban beírtat), String password
   3. firstlogin.html
4. Index page:
   1. itt láthatóak a mérkőzések adatai:
      * //localhost:8080/index
      * @GetMapping, nincs bemenő adat,
      * Model-be belerakom a visszaérkező DTO listát
   2. keresés lehetőség a nevekre:
      * //localhost:8080/index/searchuser?username=x
      * @GetMapping
      * @RequestParam – String userName
      * Model-be belerakom a visszaérkező DTO listát
   3. keresés lehetőség a location-re
      * //localhost:8080/index/searchlocation?locationname=x
      * @GetMapping
      * @RequestParam – String locationName
      * Model-be belerakom a visszaérkező DTO listát
   4. itt lesz egy lehetőség, hogy az admin átléphet a /admin oldalra.
      * megnézem,hogy adminként szerepel e a felhasználó az adatbázisban,amennyiben igen, tovább mehet az admin.html.re, egyébként nem.  
        (Ötletem ehhez az lenne, hogy bejelentkezés után a userID-t is visszaküldöm a dto-val, majd mikor az admin oldalra akarok átlépni az ID alapján csinálok egy lekérdezést és megnézem hogy az ID-hoz tartozó role az admin e vagy sem, és ez alapján küldöm az index.html/admin.html oldalra)
      * @Getmapping, @RequestParam int userID
      * //localhost:8080/admin?userid=x
      * admin.html
   5. index.html
5. Admin page:
   1. felhasználó hozzáadása lehetőség(/admin/reguser)
      * //localhost:8080/admin/adduser?userame=x&userrole=y
      * @PostMapping
      * @RequestParam - String userName, Enum userRole
   2. helyszín regisztrálása(/admin/regplace)
      * //localhost:8080/admin/regplace?locationname=x&locationaddress=y&rentfeeprohour=z
      * @PostMapping
      * @RequestParam – String locationName, String locationAddress, int rentFeeProHour
   3. mérkőzések regisztrálása(/admin/reggame)
      * //localhost:8080/admin/reggame?firstuserid=x&seconduserid=y&gamelocationid=z
      * @PostMapping
      * @RequestParam – int firstUserID, int secondUserID, int gameLocationID
   4. admin.html

**IV. Adatbázis műveletek:**

1. Felhasználó mentése (User regisztrációhoz)
2. Felhasználó módosítása (pl.: jelszó módosítása)
3. Felhasználó kiolvasása (bejelentkezéshez)
4. Helyszín mentése (Place regisztrálásához)
5. Mérkőzés mentése (Game regisztrálásához)