**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. sikeres belépés esetén:
      * első bejelentkezésnél firstlogin.html
      * ha nem először jelentkezik be index.html
   5. sikertelen belépés esetén újra login.html
   6. User DTO
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. sikeres jelszó változtatás után index.html
   4. User DTO
4. Index page:
   1. itt láthatóak a mérkőzések adatai:
      * //localhost:8080/index
      * @GetMapping, nincs bemenő adat,
      * Game DTO lista
   2. keresés lehetőség a nevekre:
      * //localhost:8080/index/searchuser?username=x
      * @GetMapping
      * @RequestParam – String userName
      * User DTO lista
   3. keresés lehetőség a location-re
      * //localhost:8080/index/searchlocation?locationname=x
      * @GetMapping
      * @RequestParam – String locationName
      * Location DTO lista
   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.
      * @Getmapping, @RequestParam int userID
      * //localhost:8080/admin?userid=x
      * admin a felhasználó akkor átirányítom az admin.html-re
      * nem admin felhasználót visszaírányítom az index.html-re
      * User DTO
5. Admin page:
   1. felhasználó hozzáadása lehetőség(/admin/reguser)
      * //localhost:8080/admin/adduser?userame=x&userrole=y
      * @PostMapping
      * @RequestParam - String userName, Enum userRole
      * User DTO
   2. helyszín regisztrálása(/admin/regplace)
      * //localhost:8080/admin/regplace?locationname=x&locationaddress=y&rentfeeprohour=z
      * @PostMapping
      * @RequestParam – String locationName, String locationAddress, int rentFeeProHour
      * Location DTO
   3. mérkőzések regisztrálása(/admin/reggame)
      * //localhost:8080/admin/reggame?firstuserid=x&seconduserid=y&gamelocationid=z
      * @PostMapping
      * @RequestParam – int firstUserID, int secondUserID, int gameLocationID
      * Game DTO
   4. Sikeres felhasználó/helyszín/mérkőzés regisztrálása esetén továbbra is az admin.html töltődik be. (vissza lépés az index oldalra extra opción keresztül)

**IV. Adatbázis műveletek:**

1. Felhasználó mentése (User regisztrációhoz)
   * + public saveUser(User user)
2. Felhasználó módosítása (pl.: jelszó módosítása)
   * + public void updateUser(User user)
3. Felhasználó kiolvasása (név alapján)
   * + public User getUserByName(String name)
4. Felhasználó kiolvasása (ID alapján)
   * + public getUserByID(int userID)
5. Felhasználók kilistázása
   * + public List<User> getUsers()
6. Helyszín mentése (Place regisztrálásához)
   * + public void saveLocation(Location location)
7. Helyszín kiolvasása (név alapján)
   * + public Location getLocationByName(String name)
8. Helyszínek kilistázása
   * + public List<Location> getLocations()
9. Mérkőzés mentése (Game regisztrálásához)
   * + public saveGame(Game game)
10. Mérkőzés kiolvasása(ID alapján)
    * + public Game getGameByID(int gameID)
11. Mérkőzések kilistázása
    * + public List<Game> getGames()