**I.Database(tables):**

1.User:

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

2.Location:

* locationID (int, nn, pk, ai)
* locationName (varchar, nn)
* loctionAddress (varchar, nn)
* rentFeePerHour (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
* boolean loggedIN

2.Location:

* int locationID
* String locationName
* String locationAddress
* int rentFeePerHour

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
   3. @RequestParam - String username, String password
   4. //localhost:8080/login
   5. sikeres belépés esetén:
      * első bejelentkezésnél firstlogin.html (UserDTO)
      * ha nem először jelentkezik be index.html (GameDTOList)
   6. sikertelen belépés esetén újra login.html
3. Firstlogin page
   1. első bejelentkezéskor ide irányítjuk a felhasználót, és csak a jelszó megváltoztatása után tovább az index oldalra
   2. //localhost:8080/login/firstlogin
   3. @PostMapping
   4. @RequestParam – int userID(ezt nem kérem be újra, automatikusan küldöm tovább a korábban beírtat), String password
   5. sikeres jelszó változtatás után index.html (GameDTOList)
   6. sikertelen jelszó változtatás esetén újra firstlogin.html (UserDTO)
4. Index page:
   1. itt láthatóak a mérkőzések adatai
   2. keresés lehetőség a nevekre:
      * //localhost:8080/index/searchuser?username=x&userid=y
      * @GetMapping
      * @RequestParam - String userName
      * @RequestParam – int userID
      * GameDTOList
      * index.html
   3. keresés lehetőség a location-re
      * //localhost:8080/index/searchlocation?locationname=x&userid=y
      * @GetMapping
      * @RequestParam - String locationName
      * @RequestParam - int userID
      * GameDTOList
      * index.html
   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(AdminDTO)
      * nem admin felhasználót visszaírányítom az index.html-re(GameDTOList)
5. Admin page:
   1. felhasználó hozzáadása lehetőség(/admin/reguser)
      * //localhost:8080/admin/reguser?userame=x&userrole=y&userid=z
      * @PostMapping
      * @RequestParam - String userName
      * @RequestParam - String userRole
      * @RequestParam – int userID
      * admin.html-re visszük
      * AdminDTO
   2. helyszín regisztrálása(/admin/reglocation)
      * //localhost:8080/admin/reglocation?locationname=x&locationaddress=y&rentfeeperhour=z&userid=v
      * @PostMapping
      * @RequestParam – String locationName
      * @ String locationAddress
      * @ int rentFeePerHour
      * @ int userID
      * admin.html-re visszük
      * AdminDTO
   3. mérkőzések regisztrálása(/admin/reggame)
      * //localhost:8080/admin/reggame?firstuserid=x&seconduserid=y&gamelocationid=z&fisrtuserscore=v&seconduserscore=b&userid=n
      * @PostMapping
      * @RequestParam - int firstUserID
      * @RequestParam - int secondUserID
      * @RequestParam - int gameLocationID
      * @RequestParam - int firstUserScore
      * @RequestParam - int secondUserScore
      * @RequestParam - int userID
      * admin.html-re visszük
      * AdminDTO
6. DTO-k?
   1. UserDTO
      * int userID
      * String userName
      * boolean validPassword
   2. GameDTOList
      * List<GameDTO > gameDTOs
      * int UserID
   3. GameDTO
      * int gameID
      * UserDTO firstUserDTO
      * int firstUserScore
      * UserDTO secondUserDTO
      * int secondUserScore
      * LocationDTO locationDTO
   4. LocationDTO
      * int locationID
      * String locationName
      * String locationAddress
      * int rentFeePerHour
   5. AdminDTO
      * List<UserDTO> userDTOList
      * List<LocationDTO> locationDTOList
      * UserDTO admin

**IV. Adatbázis műveletek:**

1. Felhasználó mentése (User regisztrációhoz)
   * + public void 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()