

*<<DatabaseController>>*

##### Treasure #####

```
+ getAllTreasures(onlyActive : boolean) : List<Treasure>
+ getTreasure(treasureId : int) : Treasure
+ getTreasures(location : GeoLocation, radius : double, onlyActive : bool) : List<Treasure>
+ saveTreasure(treasure : Treasure) : int
    # returns id of saved treasure, -1 otherwise
+ deleteTreasure(id : int) : bool
+ deleteAllTreasures() : bool
+ getLocationData(location : GeoLocation, radius : double, onlyActive : bool) : List<Treasure>
    # returns list of Treasures where only id and Location is used (rest is null)
+ getAllLocationData(onlyActive : boolean)
    # returns list of Treasures where only id and Location is used (rest is null)
+ getQuiz(treasureId : int) : Quiz
    # returns a quiz object for the given Treasure id, -1 otherwise
    # needed ? since its possible to get a whole Treasure with 'getTreasure(...)'
```

##### User #####

```
+ addUser(user : User) : int
    # returns id of created User, -1 otherwise
+ checkUserLogin(user : User) : bool
    # returns true on success, false otherwise (in 'user' only passwordHash and name is available)
    # maybe not needed since this could be done with 'getUserProfile(...)' in the Core-model
+ deleteUser(id : int) : bool
+ getUser(id : int) : User
    # returns user and all user connected data
+ getUser(name : String) : User
+ getUserProfile(id : int) : User
    # returns only the basic information(name, email, rank, email)
+ getUserProfile(name : String) : User
    # returns only the basic information(id, email, rank, email)
+ getUserInventory(uld : int) : Inventory
+ getTreasureHistory(uld : int) : List<Treasure>
    # returns a list of treasures, the treasures only need to be filled partly...
+ getHighscoreList(minRange : int, maxRange : int) : HighscoreList
```