Part 2 – Adding AppUser and Details entities

1. Create a package for your entity models
2. Create classes AppUser and Details according to *Table 1*
3. Turn Details into an entity
4. Turn AppUser into an entity
5. Set up the OneToOne relationship
6. Create interfaces and implementing classes according to *Table 2*
7. (If you have time) Test implementations of AppUserDAO and DetailsDAO.

AppUser requirements:

- appUserId generated with the identity strategy
- username need to be unique

Details requirements:

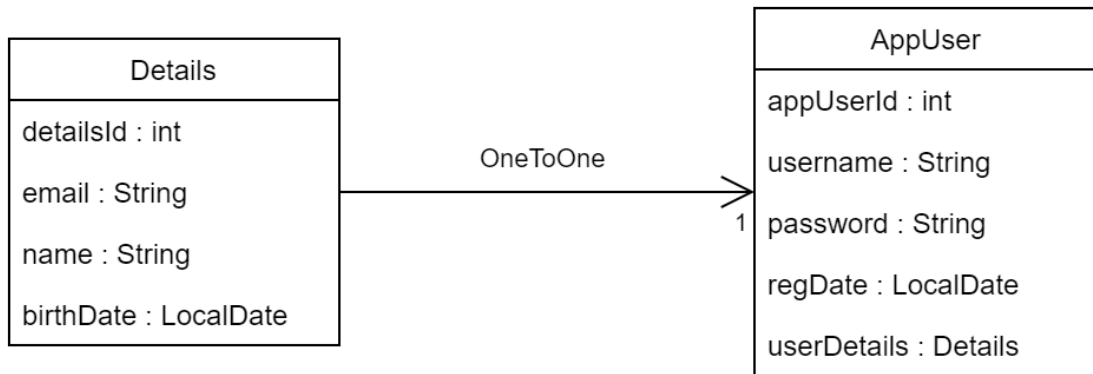- detailsId generated with identity strategy
- email need to be unique

Table 1

Details

detailsId : int

email : String

name : String

birthDate : LocalDate

OneToOne →

1

AppUser

appUserId : int

username : String

password : String

regDate : LocalDate

userDetails : Details

Table 2

AppUserDAO

findById(int) : AppUser

findAll() : Collection<AppUser>

create(AppUser) : AppUser

update(AppUser) : AppUser

delete(int) : void

package: data

DetailsDAO

findById(int) : Details

findAll() : Collection<Details>

create(Details) : Details

update(Details) : Details

delete(int) : void

AppUserDAORepository

DetailsDAORepository