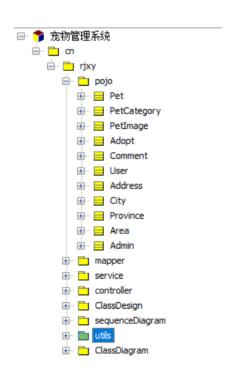
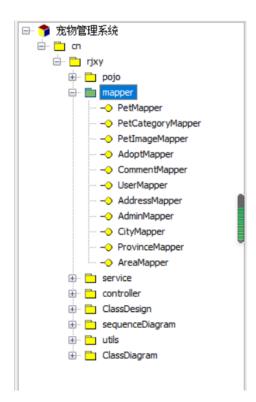
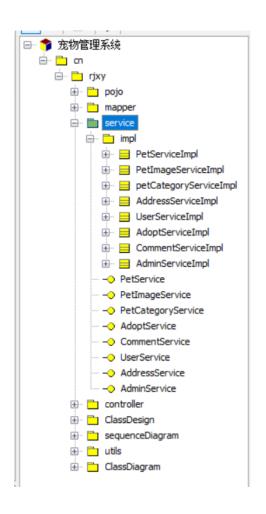
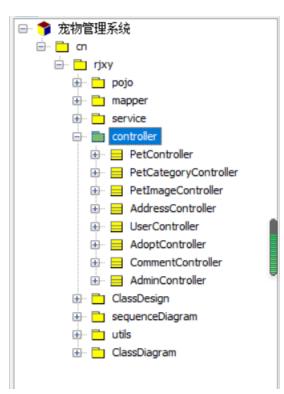
类图设计文档

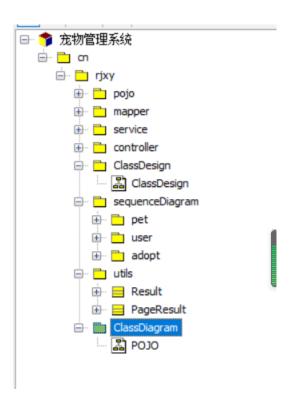
1. 概要:











2. POJO 设计:

PetCategory

+id: int +categoryCode: String +categoryName: String +adminCode: String

Pet

Hid: int
+code: String
+name: String
+birthday: Date
+Age: int
+Sex: Int
+Describe: String
+categoryCode: String
+adminCode: String
+petTmages: List<PetImage>
+petCategory: PetCategory
+idDelete: int

PetImage +id: int +petCode: String +petImage: String

Adopt

+id: int +time: Date +state: int +petCode: String +userCode: String +adminCode: String +adoptCode: String +user: User +pet: Pet

User

User

+id: Int
+code: String
+username: String
+password: String
+sex: int
+phone: String
+email: String
+idCard: String
+image: String
+image: String
+image: String
+time: Date
+state: int
+adminCode: String
+addresstd: int
+addresstd: int
+addresstd: int

Admin

+id: int +code: String +username: String +password: String

Address

+id: int +fullAddress: String +district: String +areaCode: String +cityCode: String +provinceCode: String +userCode: String

Province +id: int +code: String +name: String

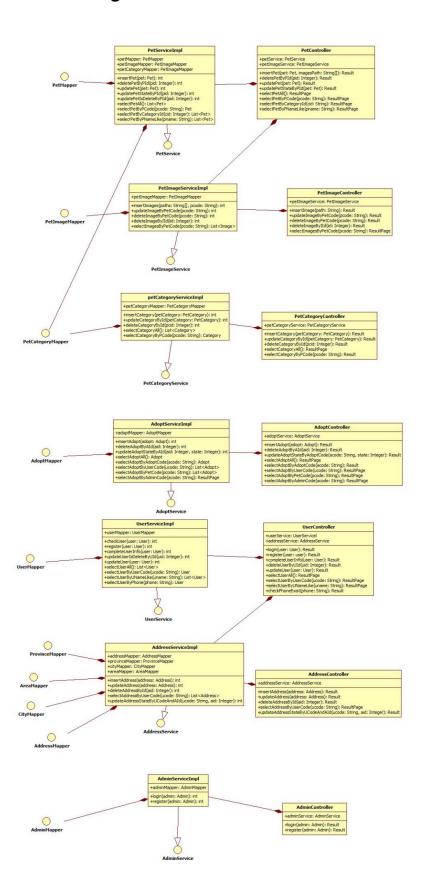
City +id: int +code: String +name: String +provinceCode: String

Area

+id: int +code: String +name: String +cityCode: Stirng

+state: int +msg: String +data: Object PageResult +Result(state, msg) +Result(state, msg, data) +Result(state, msg, data) +getState(): Int +setState(state: int): void +getMsg): String +setMag(msg: String): void +getData[): Object +setData(dataLObject): void +toString(): String +pageInfo: PageInfo +PageResult(state, msg) +PageResult(state, msg, data) +toString(): String

3. ClassDesign 设计:

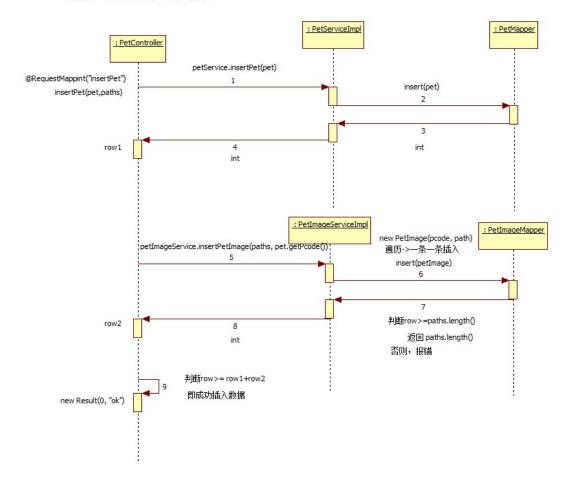


3.时序图:

3.1 Pet 业务:

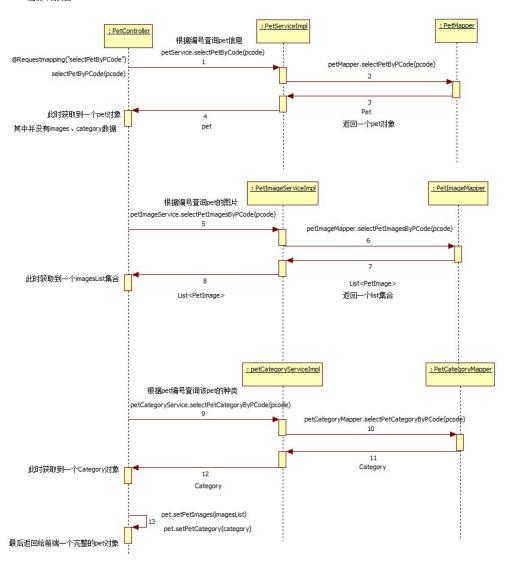
1. insertPet:新增宠物,操作两个表 t_pet,p_pet_image

新增宠物,操作两个表t_pet, p_pet_image



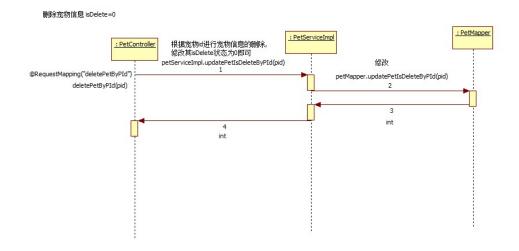
2. selectPetByCode:宠物详情界面

宠物详情页面



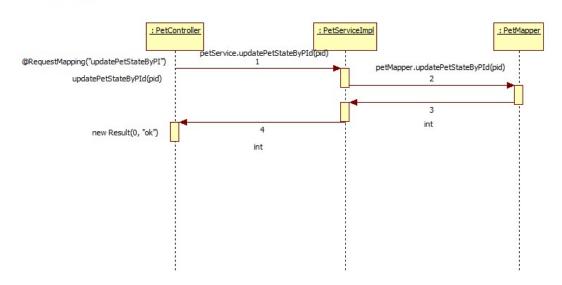
此外,查询全部宠物、根据名字查询宠物,过程都与之类似,此处不在重复就是多次调用,进行多次查询,最后返回List<Pet>即可

3. deletePetById:删除宠物信息 isDelete=0



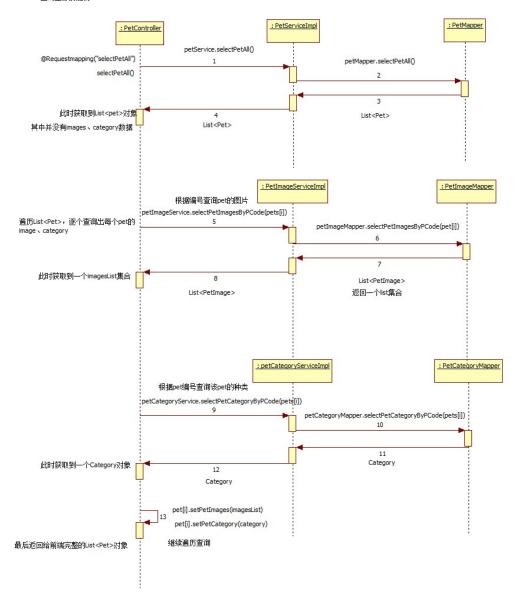
4. updatePetStateByPld:修改宠物的状态,操作表 $t_pet_state->0$ 表示被领养,1 表示没有被领养

修改宠物的状态,操作表t_pet pet_state->0表示被领养,1表示没有被领养



5. seletePetAll:查询所有的宠物

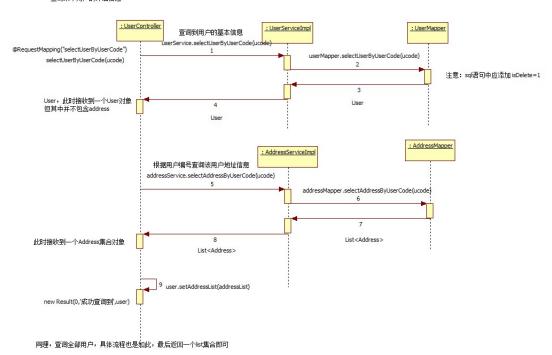
查询全部的宠物



3.2 User 业务:

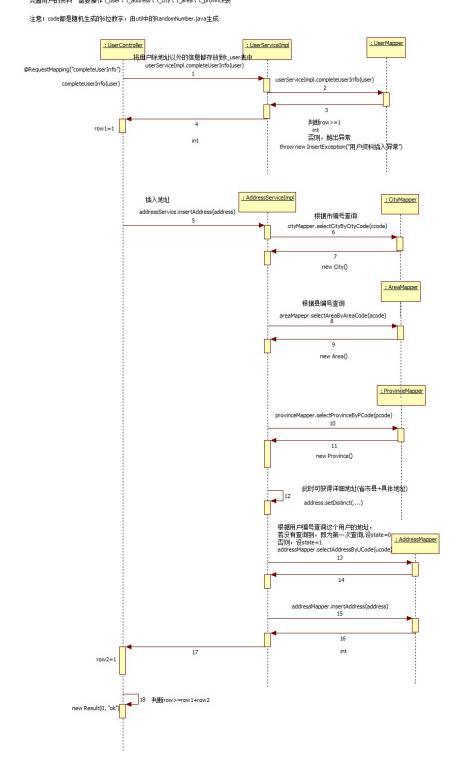
1. selectUserByUCode:查询某个用户的详细信息

查询某个用户的详细信息



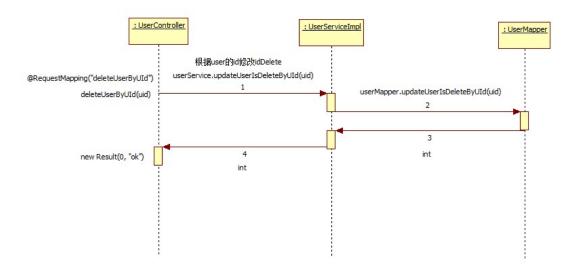
2. completeUserInfo:完善用户的资料 需要操作 t_user、t_address、t_city、t_area、t_province 表 注意: code 都是随机生成的 6 位数字,由 util 中的 RandomNumber.java 生成

注意: code 都走随机生成的 6 位数字,由 util 中的 RandomNumber.java 生成 完顯用的資料 需要操作 tuser、taddress、t.city、t_area、t_province表



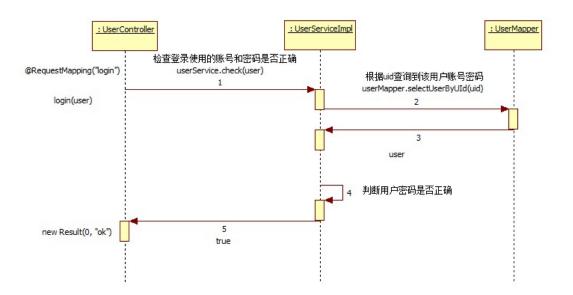
3. deleteUserByUld:删除用户信息 t_user 仅修改 isDelete=0 即可

删除用户信息 t_user 仅修改 isDelete=0即可



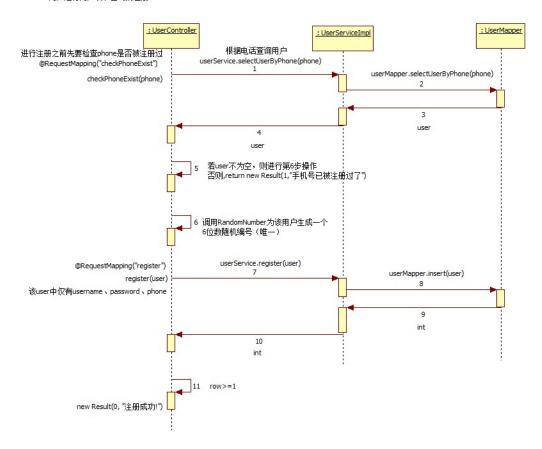
4. login:用户登录,操作 t_user

用户登陆,操作t_user



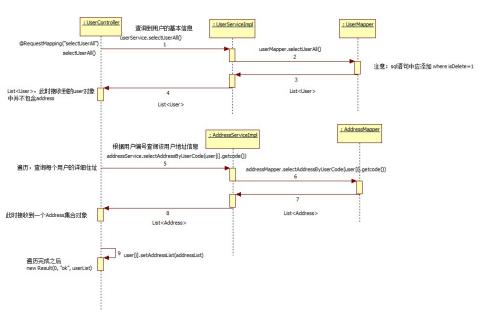
5. register:用户进行用户名和密码的注册

用户进行用户名和密码的注册



6. seleteUserAll:查询所有的用户

查询全部用户



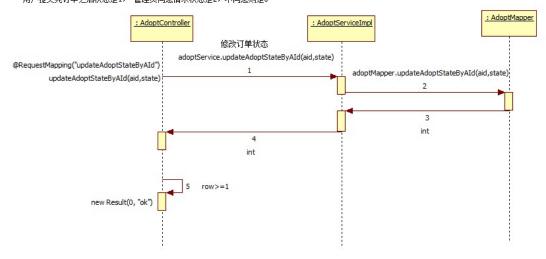
3.3 adopt 业务:

1. insertAdopt:用户进行申请领养宠物

用户进行申请领养宠物 根据用户编号进行查询, 查看用户信息是否完整 : AdoptController : UserMapper vice.selectUserByUserCode(ucode) @RequestMapping("insertAdopt") insertAdopt(adopt) User 检查user中详细数据是否填写完整, true or false true or false true->继续下一步 false->跳转至用户详情页进行信息补充 : AdoptMapper : AdoptServiceImpl 进行订单的插入 adoptService.insertAdopt(adopt) 7 adoptMapper.insertMapper(adopt) 10 new Result(0, "ok")

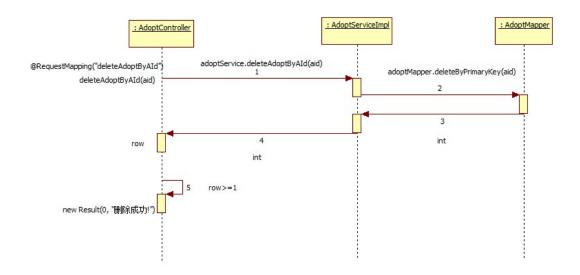
2. updateAdoptStateByAld:修改订单的状态 state->是否同意被领养, 0 表示不同意, 1 表示还在审核, 2 表示同意, 用户提交完订单之后状态为 1, 管理员同意请求状态为 2, 否则为 0

修改订单的状态 state->是否同意被领养,0表示不同意,1是还在审核,2是同意 用户提交完订单之后状态是1,管理员同意请求状态是2,不同意则是0



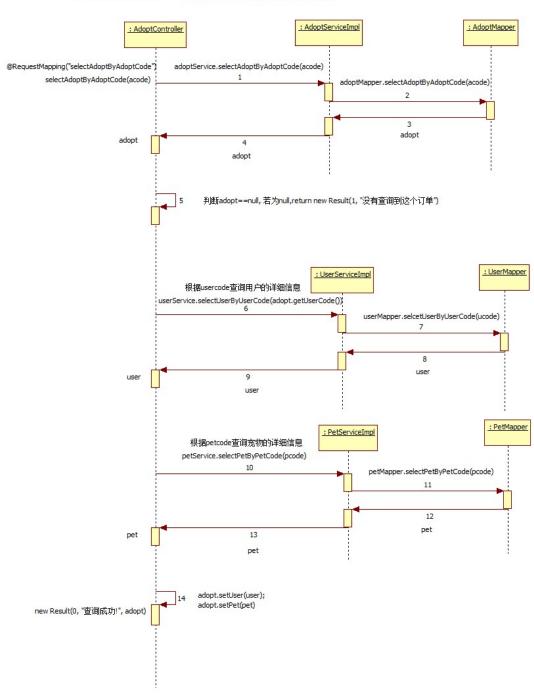
3. deleteAdopeByAld:删除订单

删除订单



4. selectAdoptByAdoptCode:根据订单编号查询订单信息 t_ adopt, t_user,t_pet_image,t_address

根据订单编号查询订单信息 t_adopt, t_user,t_pet, t_address, t_pet_image



5. selectAdoptAll:查询全部订单信息 t_adopt,t_user,t_address,t_pet_image

