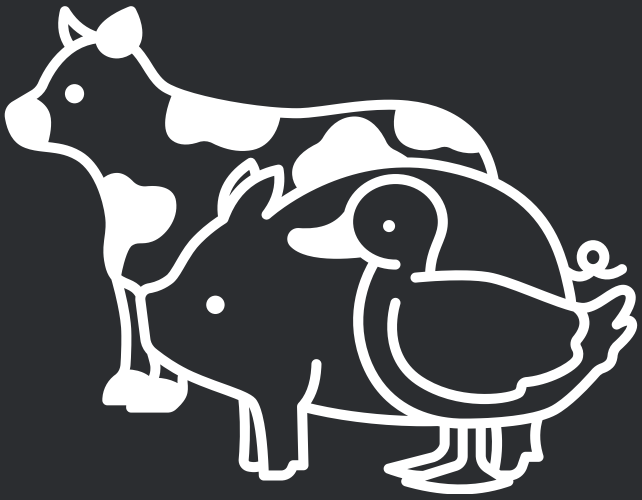


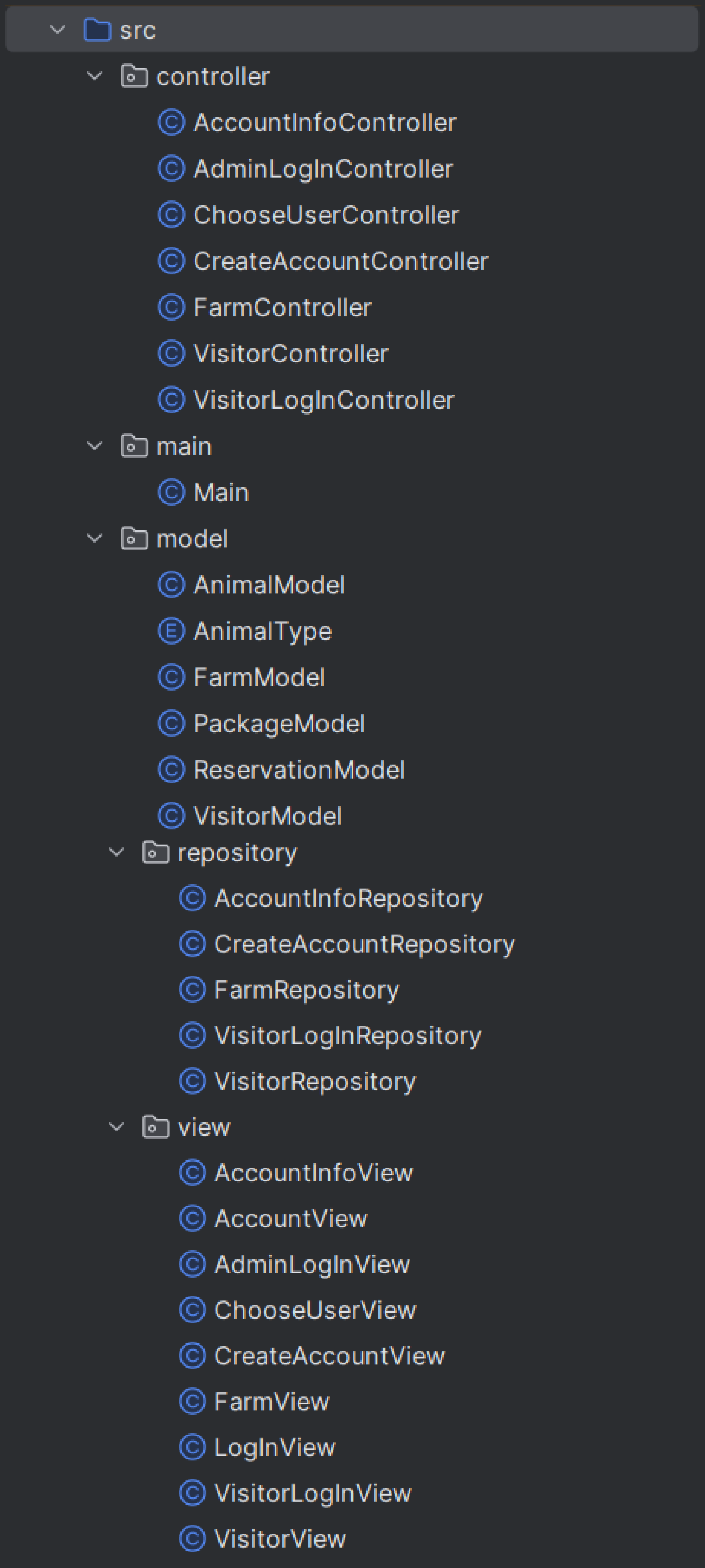
OOP DATABASE INTERFACE PROJECT DOCUMENTATION UNCLE BOB’S FARM

Zotescu Ruth

REQUIREMENTS

- Compilation 1p
- MVC 1p (pg. 4)
- Inheritance (Override & Overload) 0.5p (pg. 5)
- Enumerations 0.5p (pg. 6)
- Desktop Application 0.5p (pg. 7-8)
- Min 5 Views 1p (pg. 7-8)
- 2 Types of user 1p (pg. 4/7-8)
- Exception Handling 0.5p (pg. 9)
- Code Style 1p
- Database Connection 1p (pg. 10)
- CRUD on entities 1p (pg. 11)
- Documentation 1p

PROJECT STRUCTURE



```
src
  controller
    Ⓒ AccountInfoController
    Ⓒ AdminLogInController
    Ⓒ ChooseUserController
    Ⓒ CreateAccountController
    Ⓒ FarmController
    Ⓒ VisitorController
    Ⓒ VisitorLogInController
  main
    Ⓒ Main
  model
    Ⓒ AnimalModel
    Ⓔ AnimalType
    Ⓒ FarmModel
    Ⓒ PackageModel
    Ⓒ ReservationModel
    Ⓒ VisitorModel
   repository
     Ⓒ AccountInfoRepository
     Ⓒ CreateAccountRepository
     Ⓒ FarmRepository
     Ⓒ VisitorLogInRepository
     Ⓒ VisitorRepository
   view
     Ⓒ AccountInfoView
     Ⓒ AccountView
     Ⓒ AdminLogInView
     Ⓒ ChooseUserView
     Ⓒ CreateAccountView
     Ⓒ FarmView
     Ⓒ LogInView
     Ⓒ VisitorLogInView
     Ⓒ VisitorView
```

MVC

Choose User

- Ⓒ ChooseUserView
- Ⓒ ChooseUserController

Admin Log In

- Ⓒ AdminLogInView
- Ⓒ AdminLogInController

Farm

- Ⓒ FarmView
- Ⓒ FarmController
- Ⓒ FarmRepository
- Ⓒ AnimalModel
- Ⓒ FarmModel
- Ⓔ AnimalType

ADMIN USER

Visitor Log In

VISITOR USER

- Ⓒ VisitorLogInView
- Ⓒ VisitorLogInController
- Ⓒ VisitorLogInRepository
- Ⓒ VisitorModel

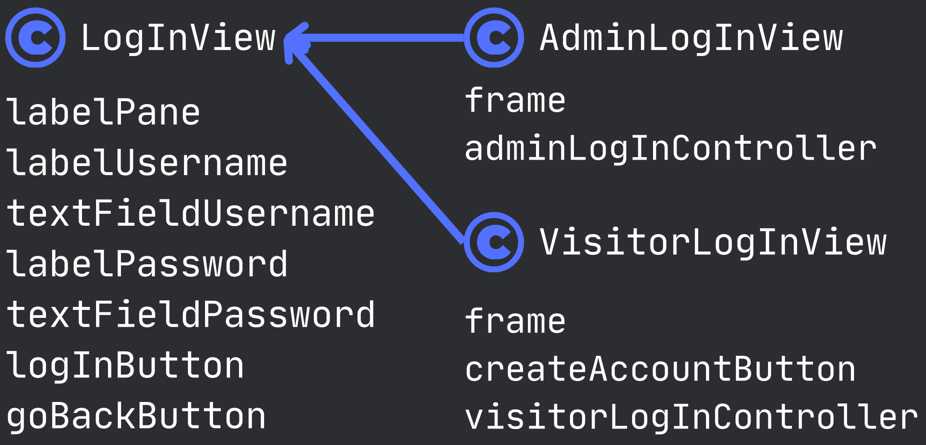
Create Account

- Ⓒ CreateAccountView
- Ⓒ CreateAccountController
- Ⓒ CreateAccountRepository
- Ⓒ VisitorModel

Visitor

- Ⓒ VisitorView
- Ⓒ VisitorController
- Ⓒ VisitorRepository
- Ⓒ AccountInfoView
- Ⓒ VisitorlModel
- Ⓒ ReservationModel
- Ⓒ PackageModel

INHERITENCE



LogInView

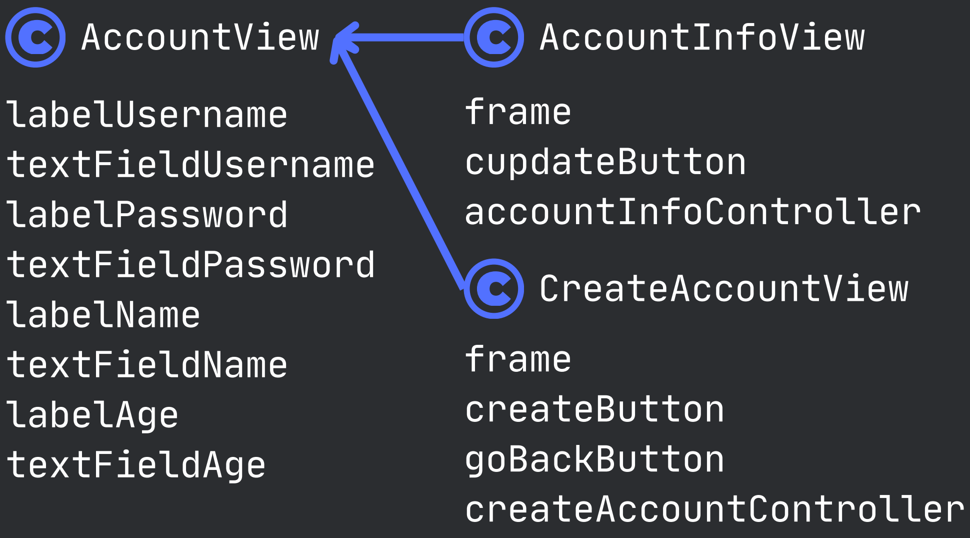
labelPane
labelUsername
textFieldUsername
labelPassword
textFieldPassword
logInButton
goBackButton

AdminLogInView

frame
adminLogInController

VisitorLogInView

frame
createAccountButton
visitorLogInController



AccountView

labelUsername
textFieldUsername
labelPassword
textFieldPassword
labelName
textFieldName
labelAge
textFieldAge

AccountInfoView

frame
cupdateButton
accountInfoController

CreateAccountView

frame
createButton
goBackButton
createAccountController

Overload

FarmView

```
displayData(List<AnimalModel> animals)
displayData(List<AnimalModel> animals, String searchText)
```

Override

FarmView

```
table.addMouseListener(new java.awt.event.MouseAdapter() {
   @Override
   public void mouseClicked(java.awt.event.MouseEvent evt) {
   ...
}
});
```

ENUMARATION

Ⓔ AnimalType

```
ANIMAL,
SHEEP,
CHICKEN,
GOOSE,
PIG,
GOAT,
HORSE,
DUCK,
PONY,
COW
```

USAGE

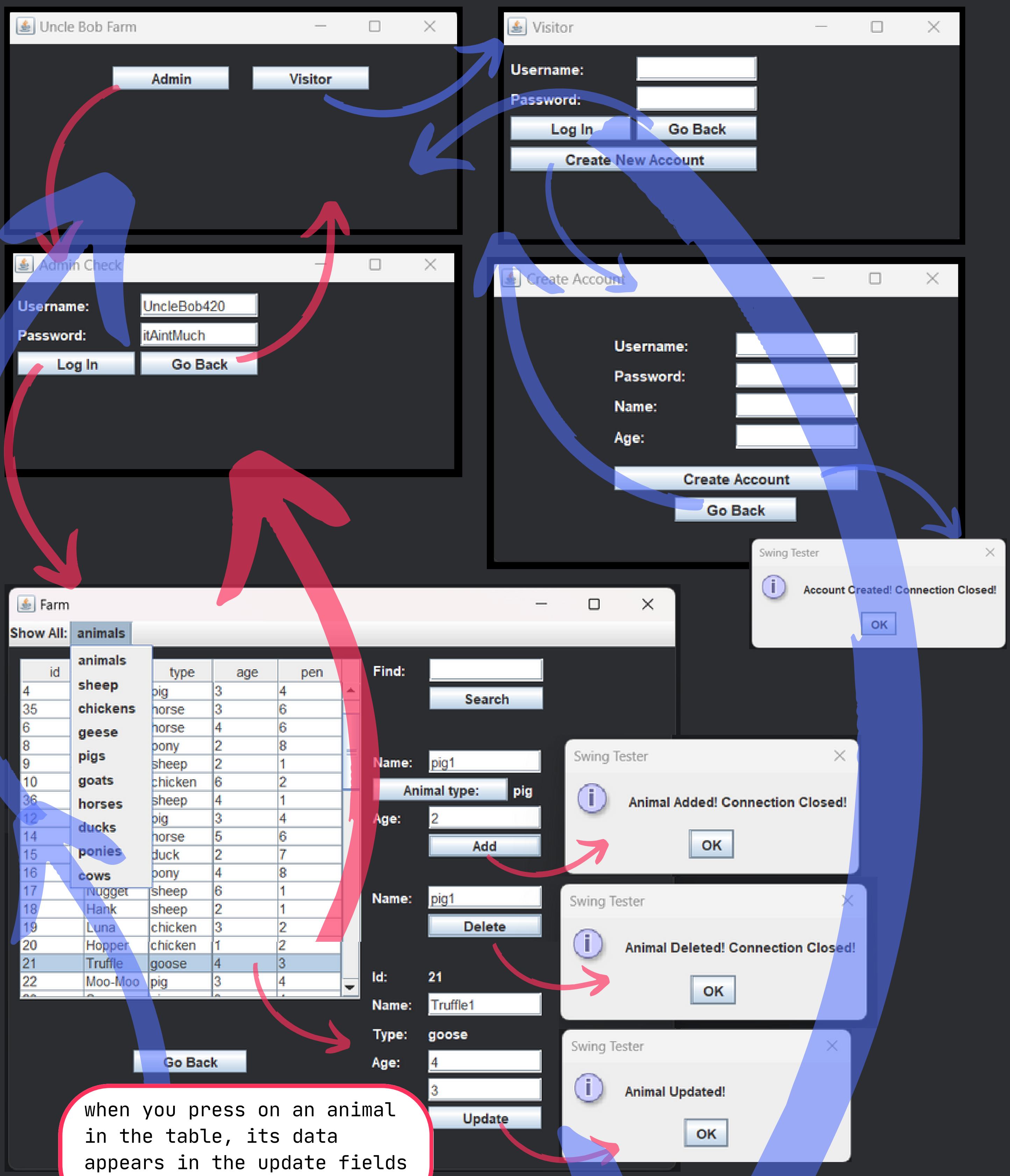
Ⓒ FarmView

```
animals.addActionListener(e -> farmController.animalButtonClicked(AnimalType.ANIMAL));
sheep.addActionListener(e -> farmController.animalButtonClicked(AnimalType.SHEEP));
chickens.addActionListener(e -> farmController.animalButtonClicked(AnimalType.CHICKEN));
geese.addActionListener(e -> farmController.animalButtonClicked(AnimalType.GOOSE));
....
```

Ⓒ FarmController

```
 public void animalButtonClicked(AnimalType animalType) {
            ...
    }
```

Desktop Application



Username:
Password:

Username: UncleBob420
Password: itAintMuch

Username:
Password:
Name:
Age:

Account Created! Connection Closed!

Show All: animals

animals
sheep
chickens
geese
pigs
goats
horses
ducks
ponies
cows

id		type	age	pen
4		pig	3	4
35		horse	3	6
6		horse	4	6
8		pony	2	8
9		sheep	2	1
10		chicken	6	2
36		sheep	4	1
12		pig	3	4
14		horse	5	6
15		duck	2	7
16		pony	4	8
17	Nugget	sheep	6	1
18	Hank	sheep	2	1
19	Luna	chicken	3	2
20	Hopper	chicken	1	2
21	Truffle	goose	4	3
22	Moo-Moo	pig	3	4

Find:

Name: pig1
Animal type: pig
Age: 2

Animal Added! Connection Closed!

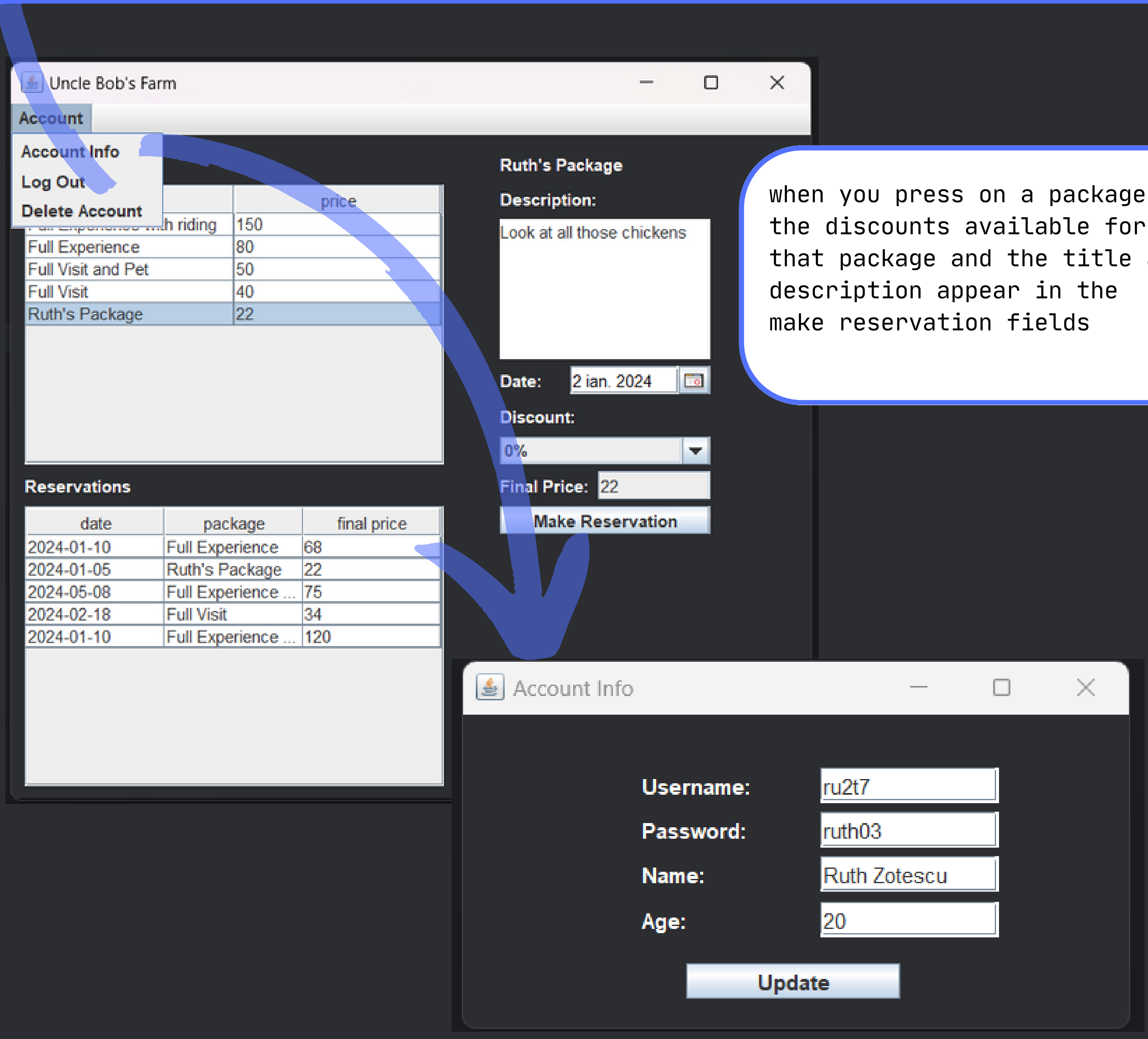
Name: pig1

Animal Deleted! Connection Closed!

Id: 21
Name: Truffle1
Type: goose
Age: 4
3

Animal Updated!

when you press on an animal in the table, its data appears in the update fields so you can modify it



Uncle Bob's Farm

	price
Full Experience with riding	150
Full Experience	80
Full Visit and Pet	50
Full Visit	40
Ruth's Package	22

Ruth's Package

Description:

Look at all those chickens

Date: 2 ian. 2024

Discount: 0%

Final Price: 22

Reservations

date	package	final price
2024-01-10	Full Experience	68
2024-01-05	Ruth's Package	22
2024-05-08	Full Experience ...	75
2024-02-18	Full Visit	34
2024-01-10	Full Experience ...	120

Account Info

Username: ru2t7

Password: ruth03

Name: Ruth Zotescu

Age: 20

when you press on a package, the discounts available for that package and the title and description appear in the make reservation fields

Exception Handeling Example

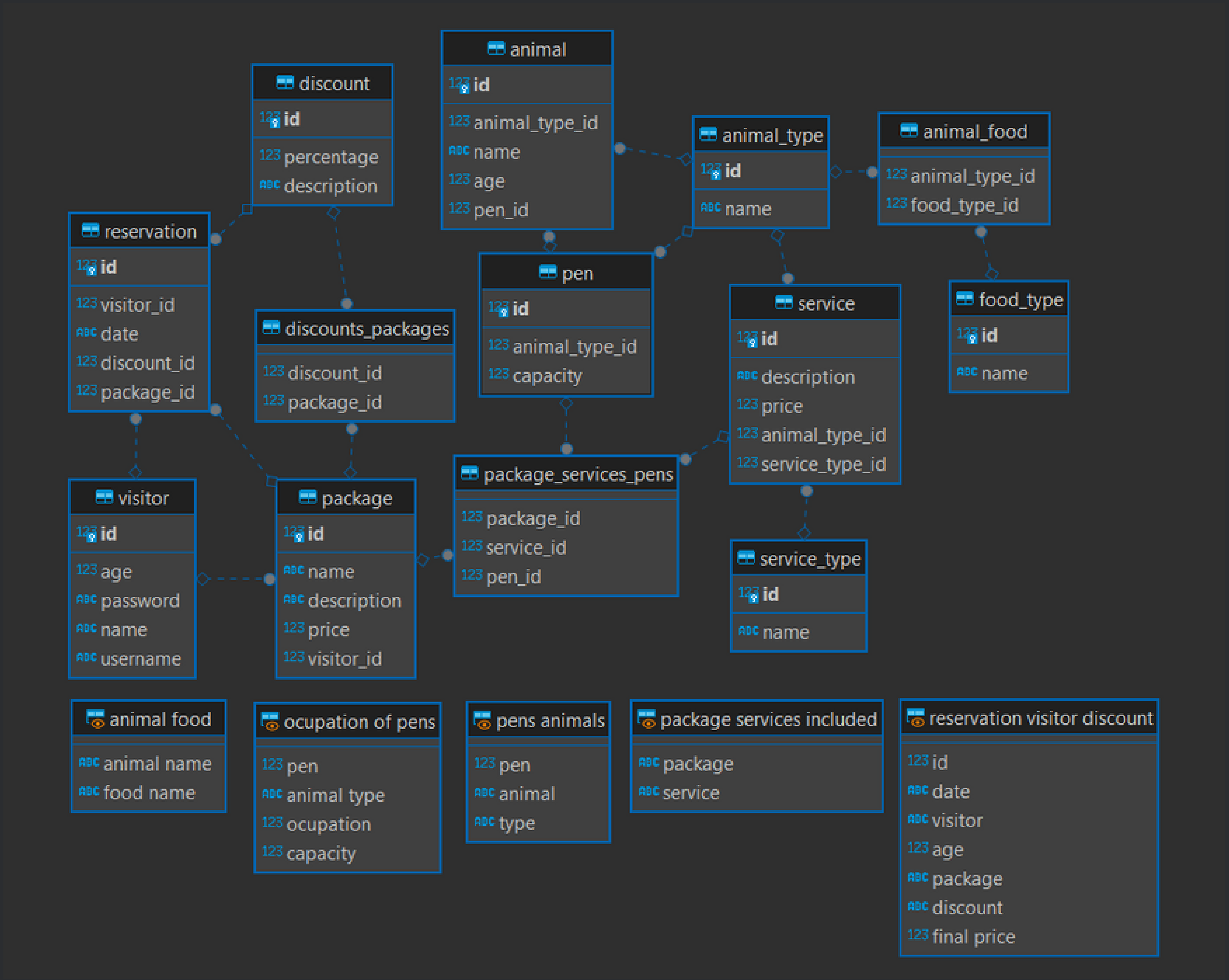
AdminLogInController

```
 public void logInButtonClicked() {

  try {
    String username = this.adminLogInView.getUsername();
    String password = this.adminLogInView.getPassword();
    boolean checkPassed = checkUsernameAndPassword(username, password);
    String message = checkPassed ? "Welcome Uncle Bob!" :
                                   "Username or Password Incorrect";
    if (checkPassed) this.farmView();
      this.adminLogInView.showMessage(message, 1);
  } catch (Exception e) {
      this.adminLogInView.showMessage("Input error!", 0);
  }

}
```

DATABASE



DATABASE CONNECTION

```
String url = "jdbc:postgresql://localhost:5432/unclebobfarm";
String dbUsername = "postgres";
String dbPassword = "root";
```

CRUD

Create

Ⓒ CreateAccountRepository

- createAccount(VisitorModel visitor)

Ⓒ FarmRepository

- createAnimal(AnimalModel animal)

Ⓒ VisitorRepository

- createReservation(ReservationModel reservation)

Read

Ⓒ CreateAccountRepository

- getNextIndex()

Ⓒ FarmRepository

- getPenIds(int animalTypeId)
- isThereSpace(int penId)
- getNextIndex()
- showAllAnimals()
- getData(int animalTypeId)

Ⓒ VisitorRepository

- getNextIndex()
- getPackagesData(VisitorModel visitor)
- getReservationsData(VisitorModel visitor)
- getPackage(String name)

Update

Ⓒ AccountInfoRepository

- updateVisitor(VisitorModel visitor)

Ⓒ FarmRepository

- updateAnimal(AnimalModel animal)

Delete

Ⓒ VisitorRepository

- deleteVisitor(VisitorModel visitor)

Ⓒ FarmRepository

- deleteAnimal(String name)