

GRASP JUSTIFICATION FOR A PARTICULAR CLASS

CLASS: PetClinic

OPERATIONS

- **Operations**

1. Import()

- The responsibility of the PetClinic class is to act as an Information Expert and Creator to the DAO objects as well as provide Indirection to them. Having the Import function allows to maintain high cohesion as the PetClinic can read in and update its DOAs as needed

2. Export()

- The responsibility of the PetClinic class is to act as an Information Expert and Creator to the DAO objects as well as provide Indirection to them. Having the Export function allows to maintain high cohesion as the PetClinic can write to a file to later update its DAOs as needed

3. isEmpty()

- The responsibility of the PetClinic class is to act as an Information Expert and Creator to the DAO objects as well as provide Indirection to them. The PetClinic Class must know if it's EmployeeDAO is empty ensure the DAO never tries to refer to a null object.

4. findPet()

- The findPet method allows for indirection from outside controllers by searching through the PetDAO class by going through the PetClinic class first before doing a DML function inside the PetDAO class. This also supports High cohesion between the PetClinic and its petInventory member variable while also supporting low cohesion as the PetDAO will do the desired DML and not the PetClinic Class

5. findCustomer()

- The findCustomer method allows for indirection from outside controllers by searching through the CustomerDAO class by going through the PetClinic class first before doing a DML function inside the CustomerDAO class. This also supports High cohesion between the PetClinic and its custDatabase member variable while also supporting low cohesion as the CustomerDAO will do the desired DML and not the PetClinic Class

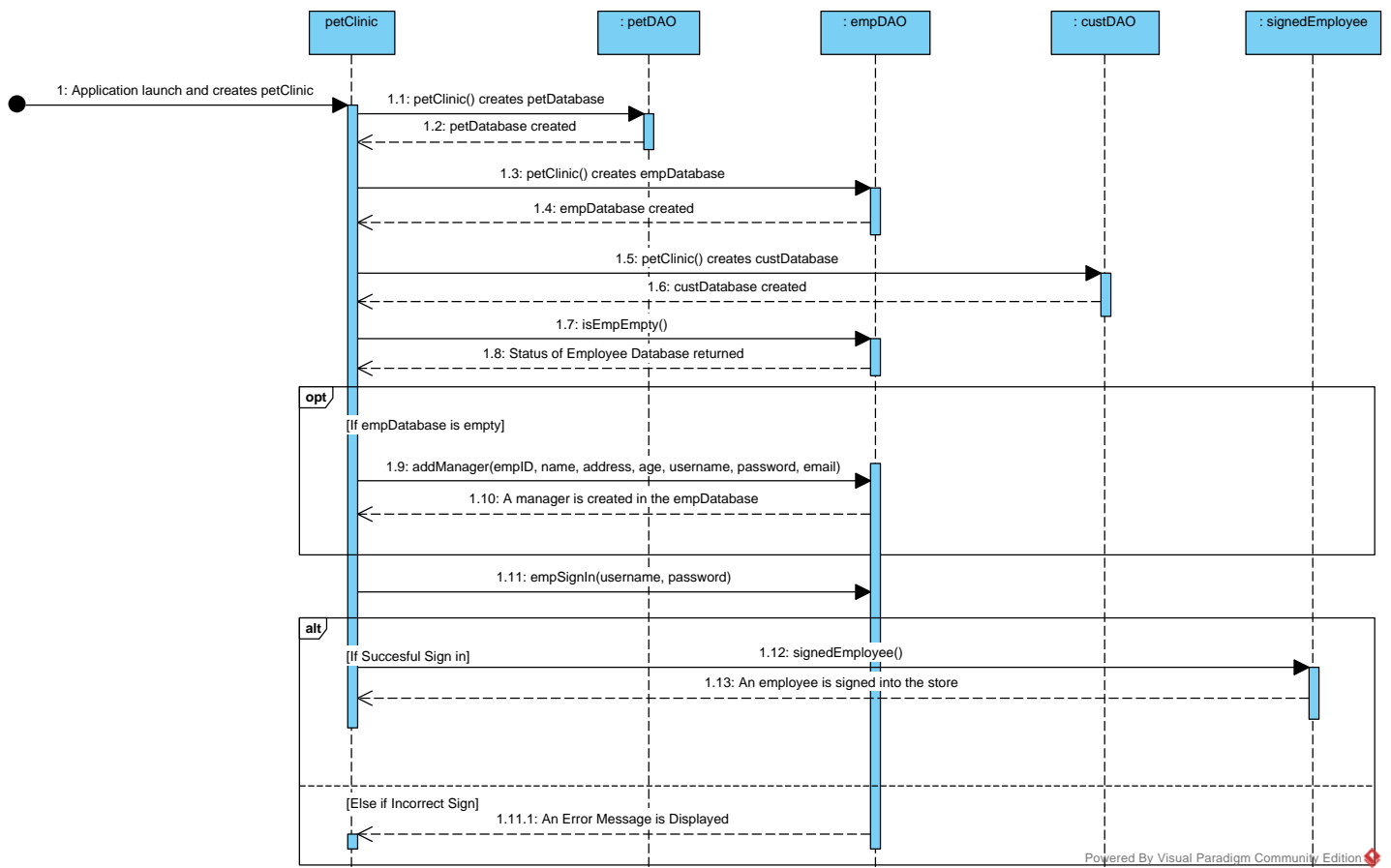
6. findEmployee()

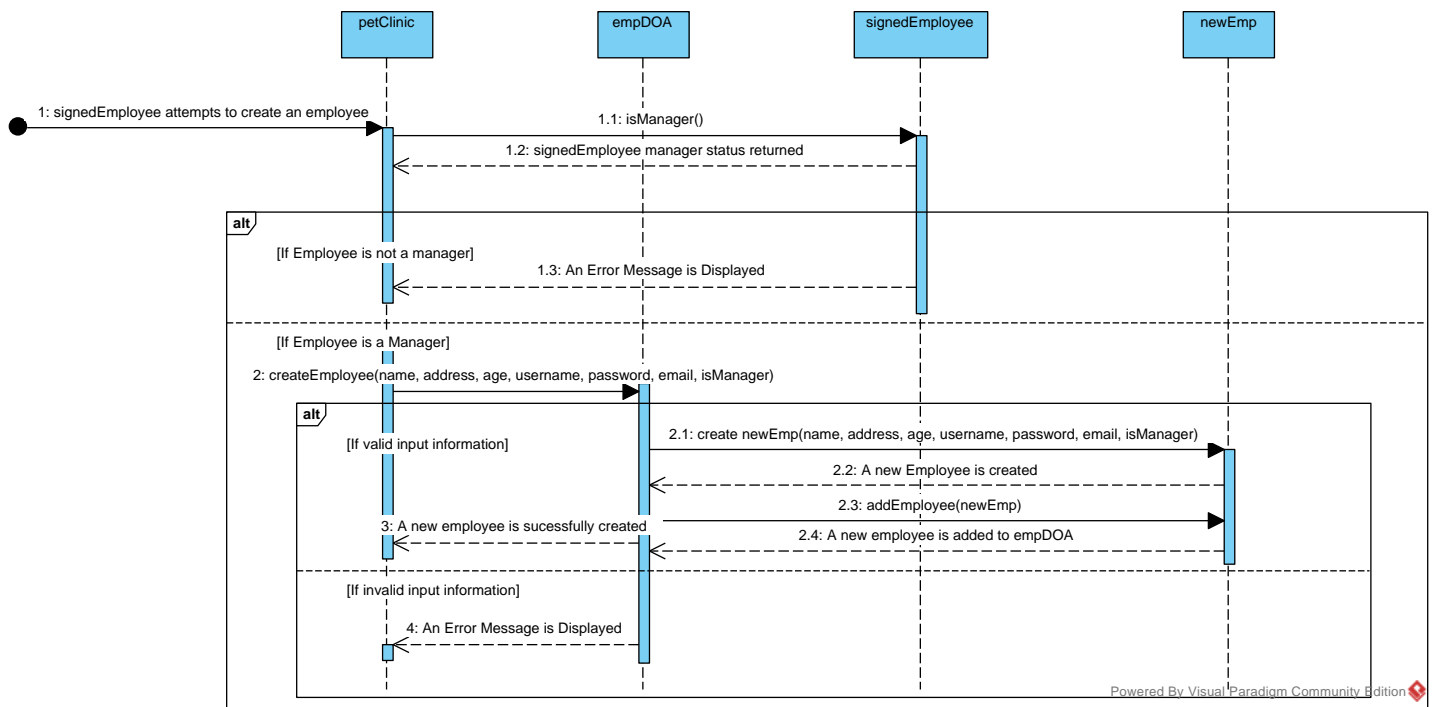
- The findEmployee method allows for indirection from outside controllers by searching through the EmployeeDAO class by going through the PetClinic class first before doing a DML function inside the EmployeeDAO class. This also supports High cohesion between the

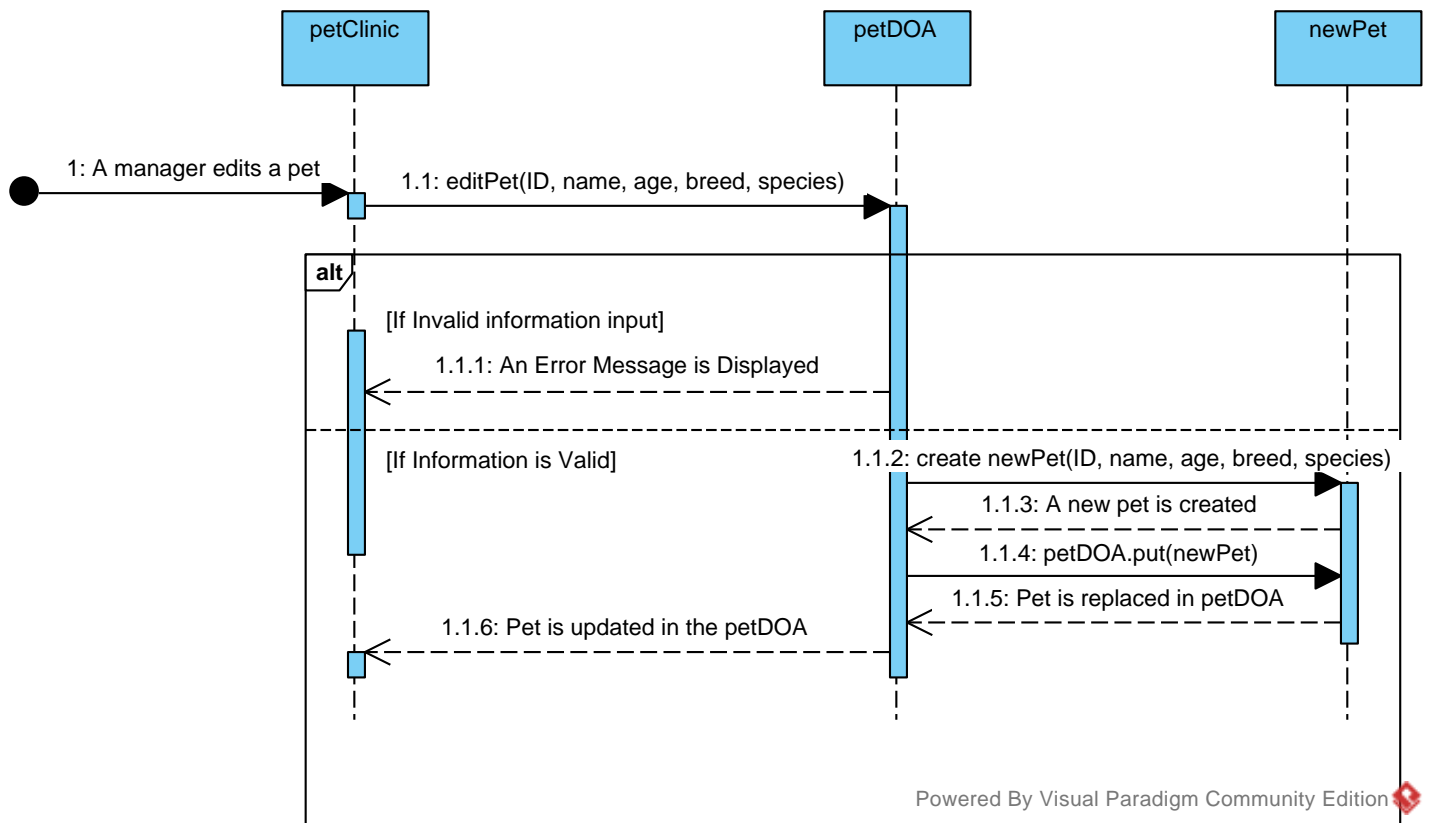
PetClinic and its employDatabase member variable while also supporting low cohesion as the EmployeeDAO will do the desired DML and not the PetClinic Class

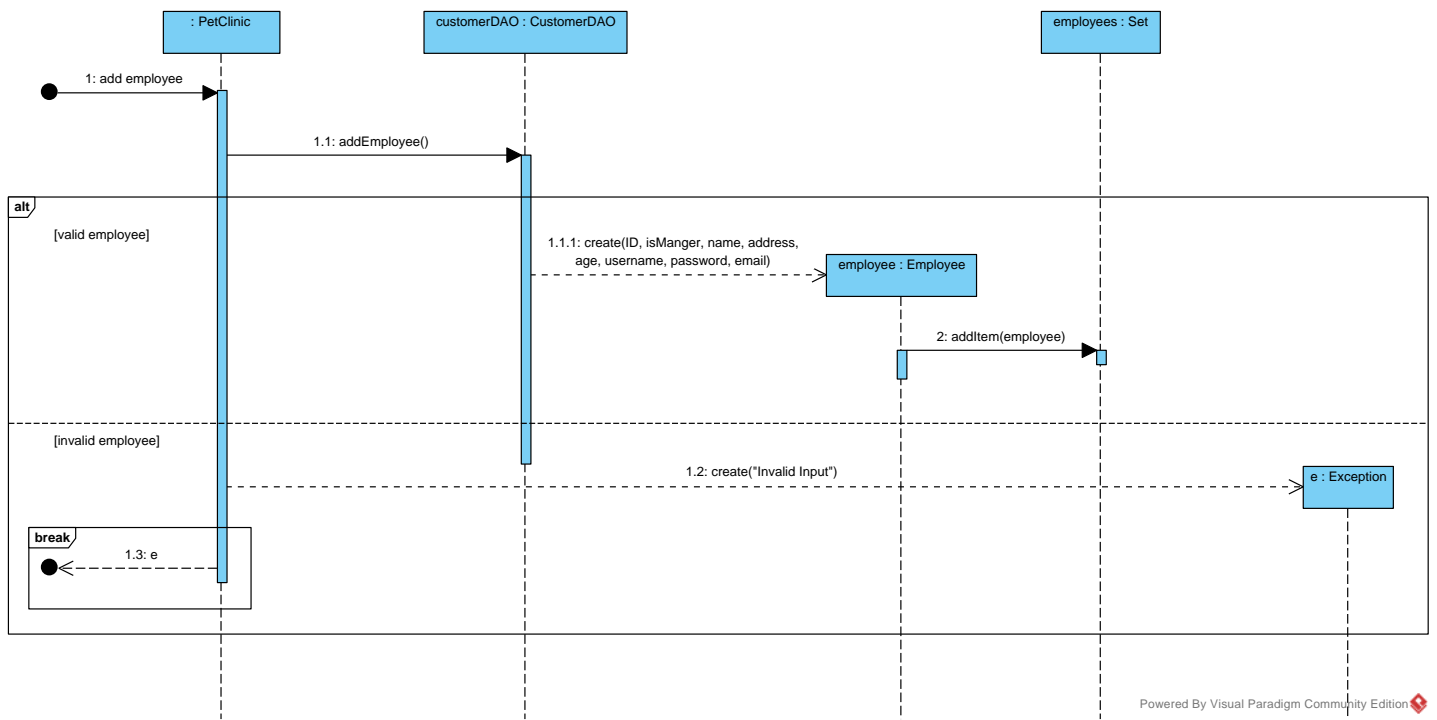
7. Adopt()

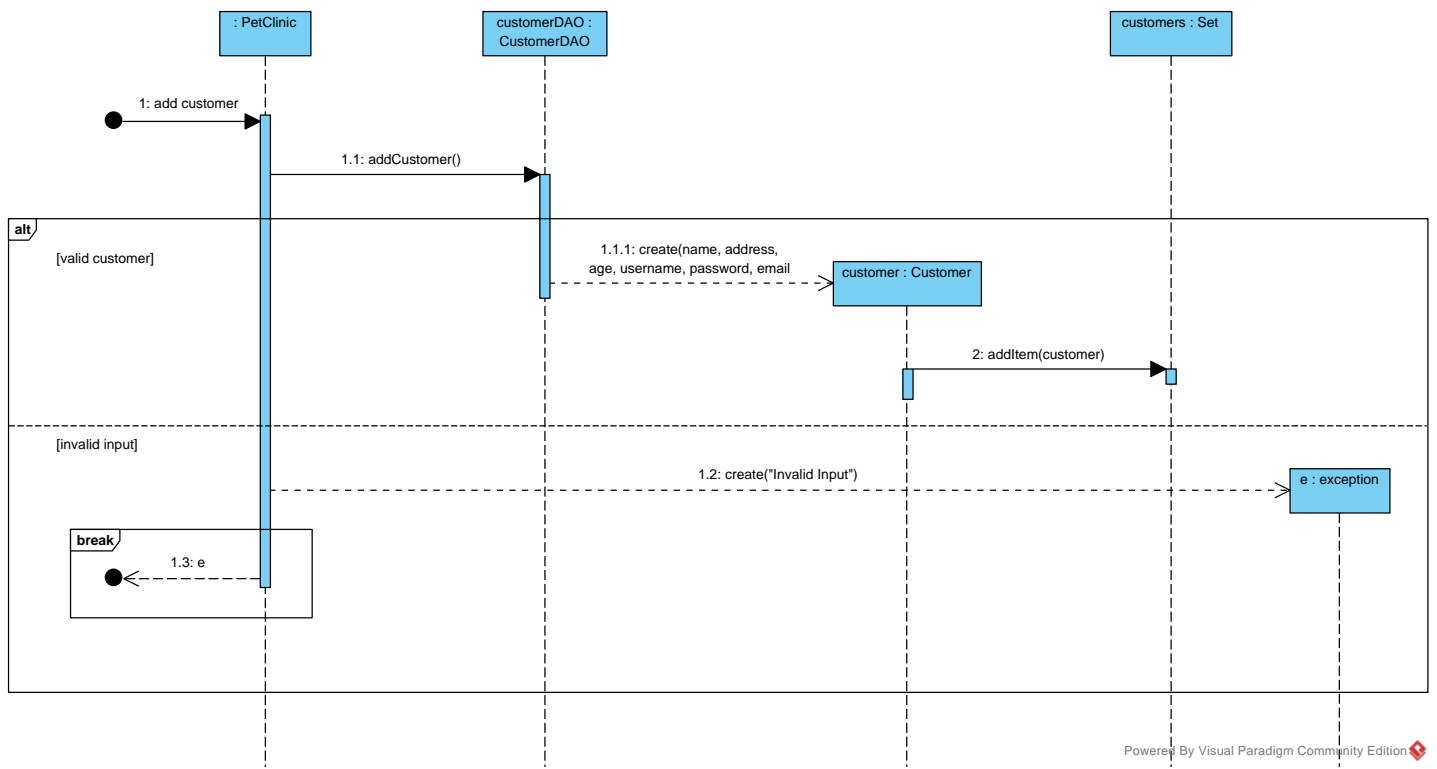
- The Adopt method allows for a pure fabrication approach when manipulating the PetClinic Class's member variables. It allows for Indirection for the Sale class by only being able to create a receipt. By using Indirection and Pure Fabrication in this way, we can maintain a lower coupling for the PetClinic class and also a high cohesion between the PetClinic Class and its DAO member variables

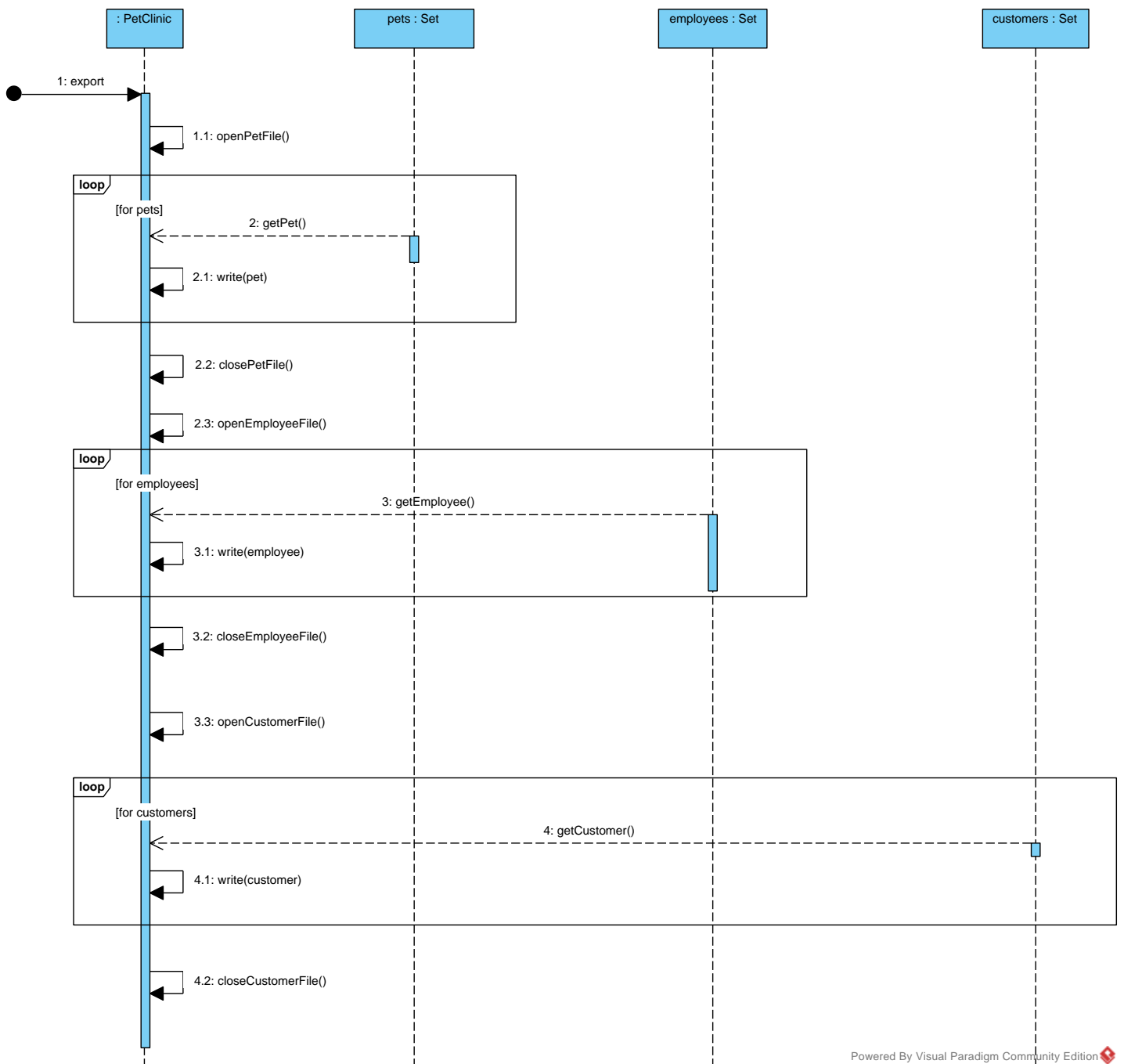


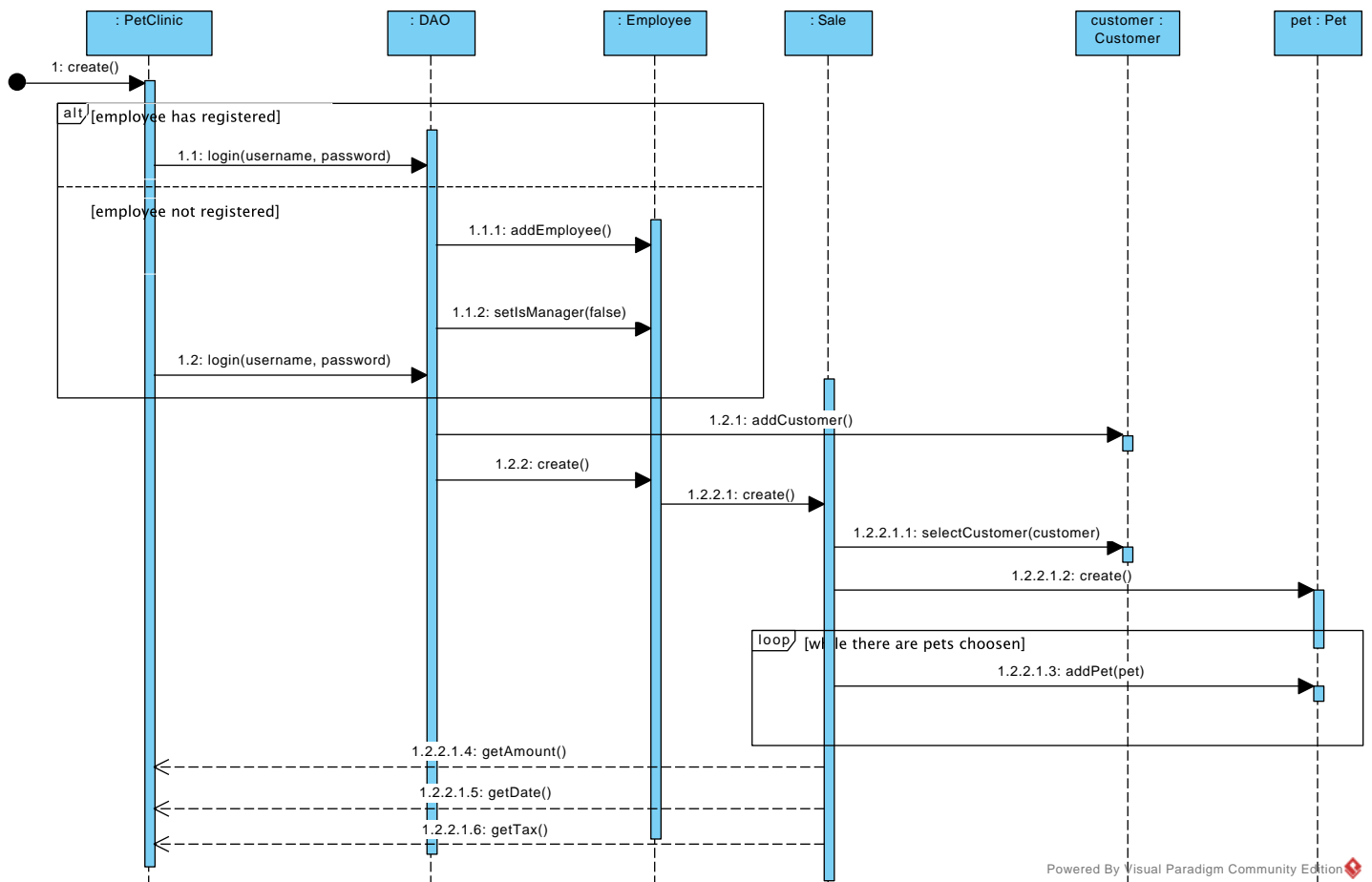


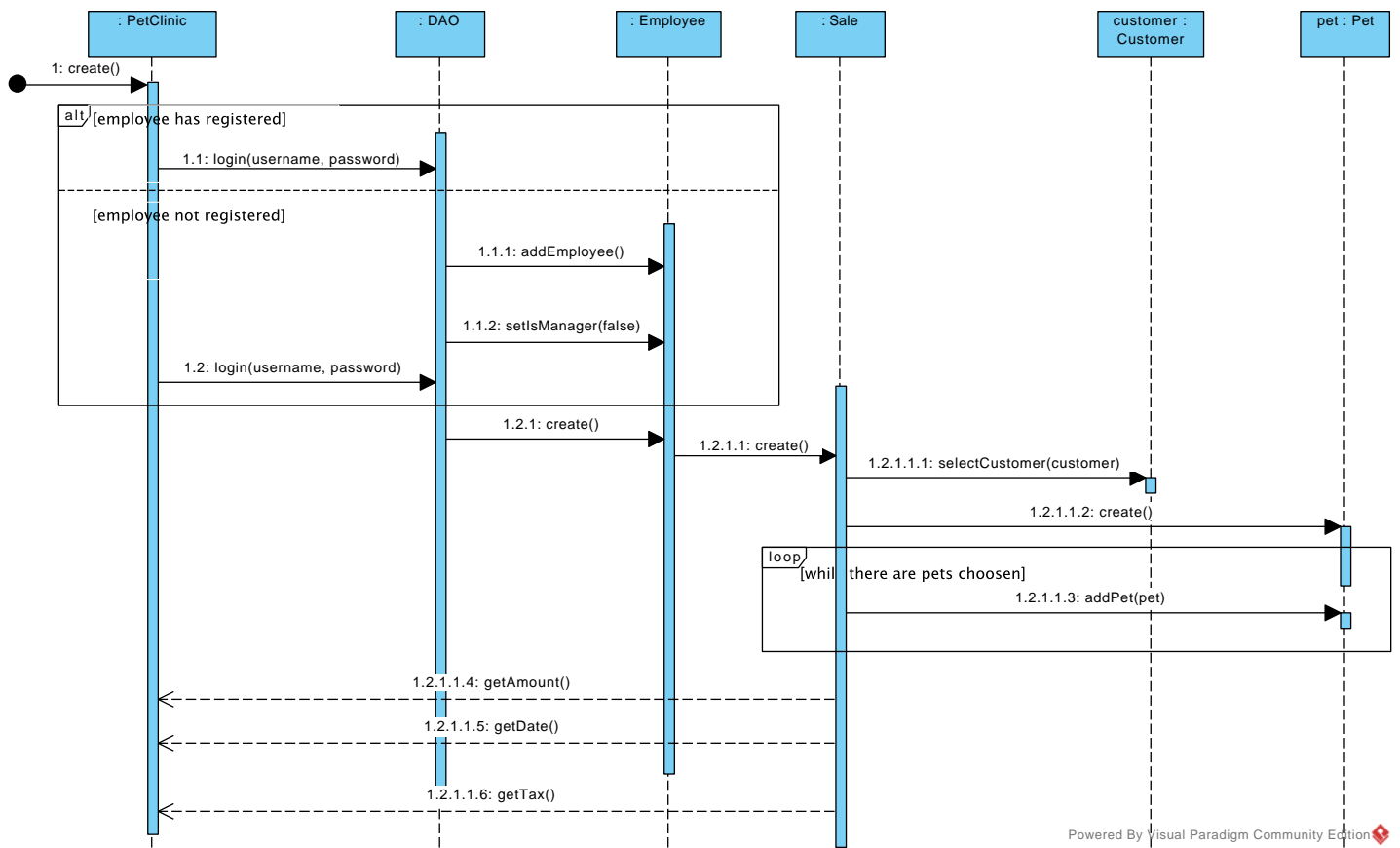


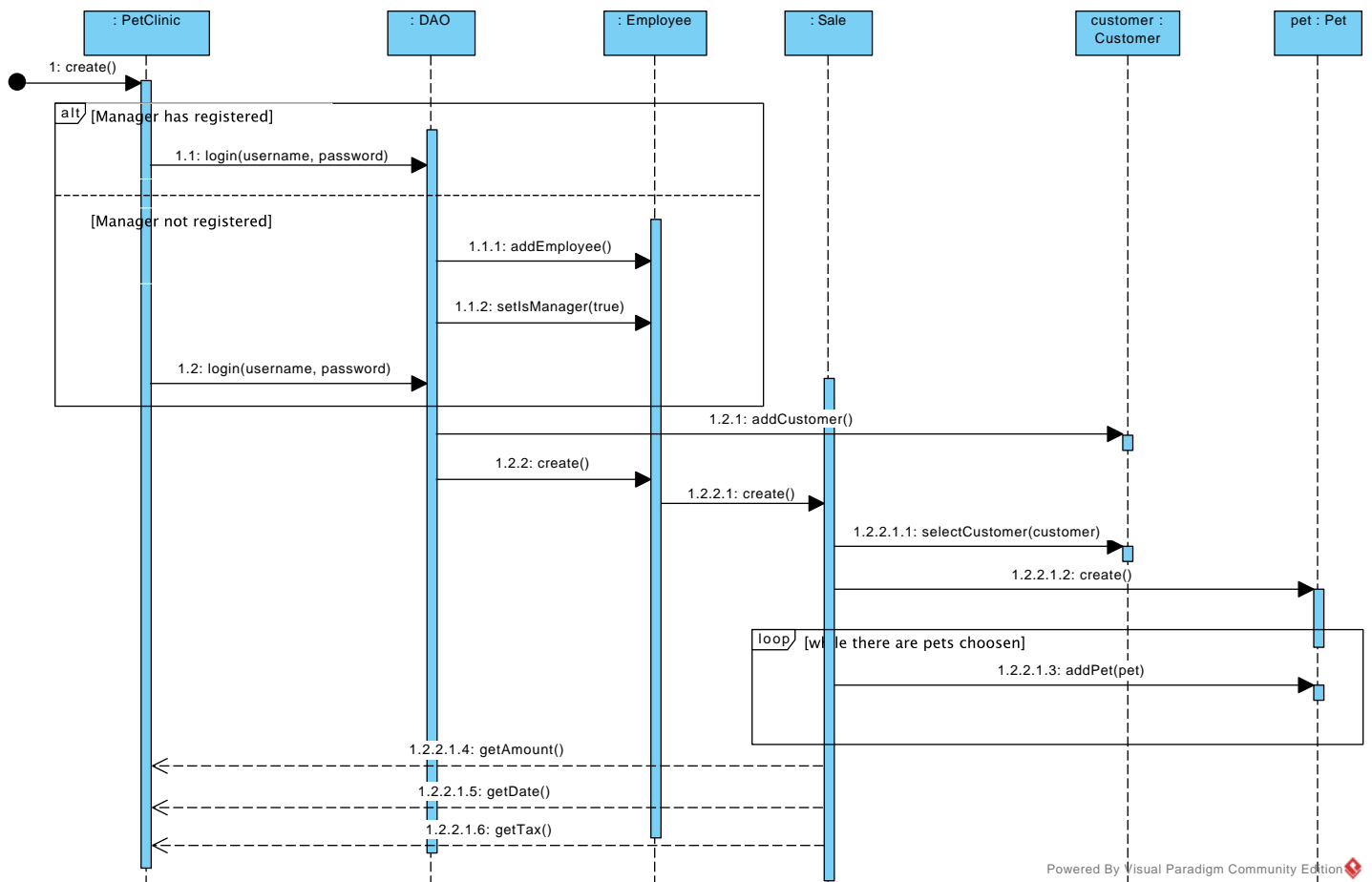




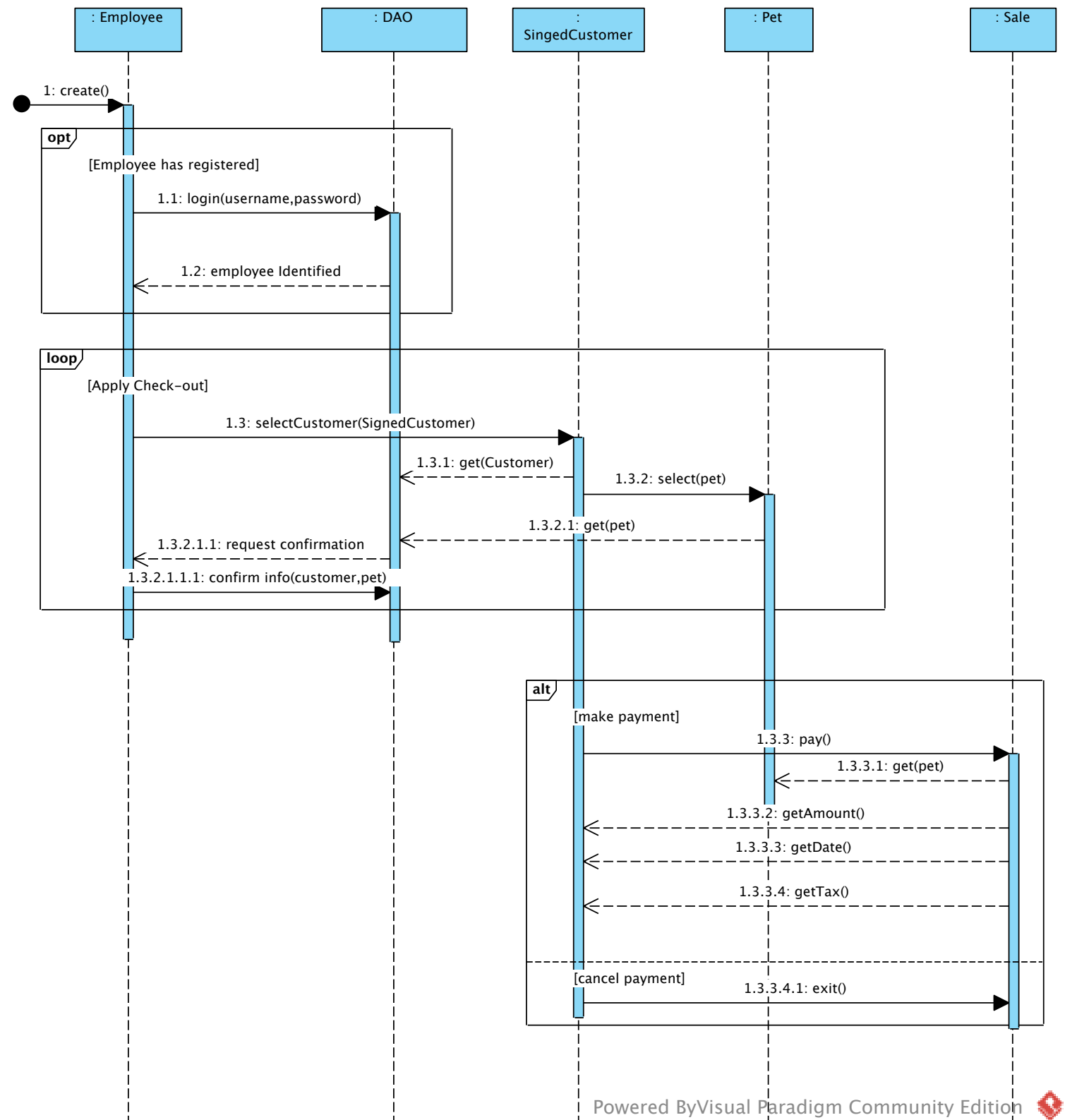




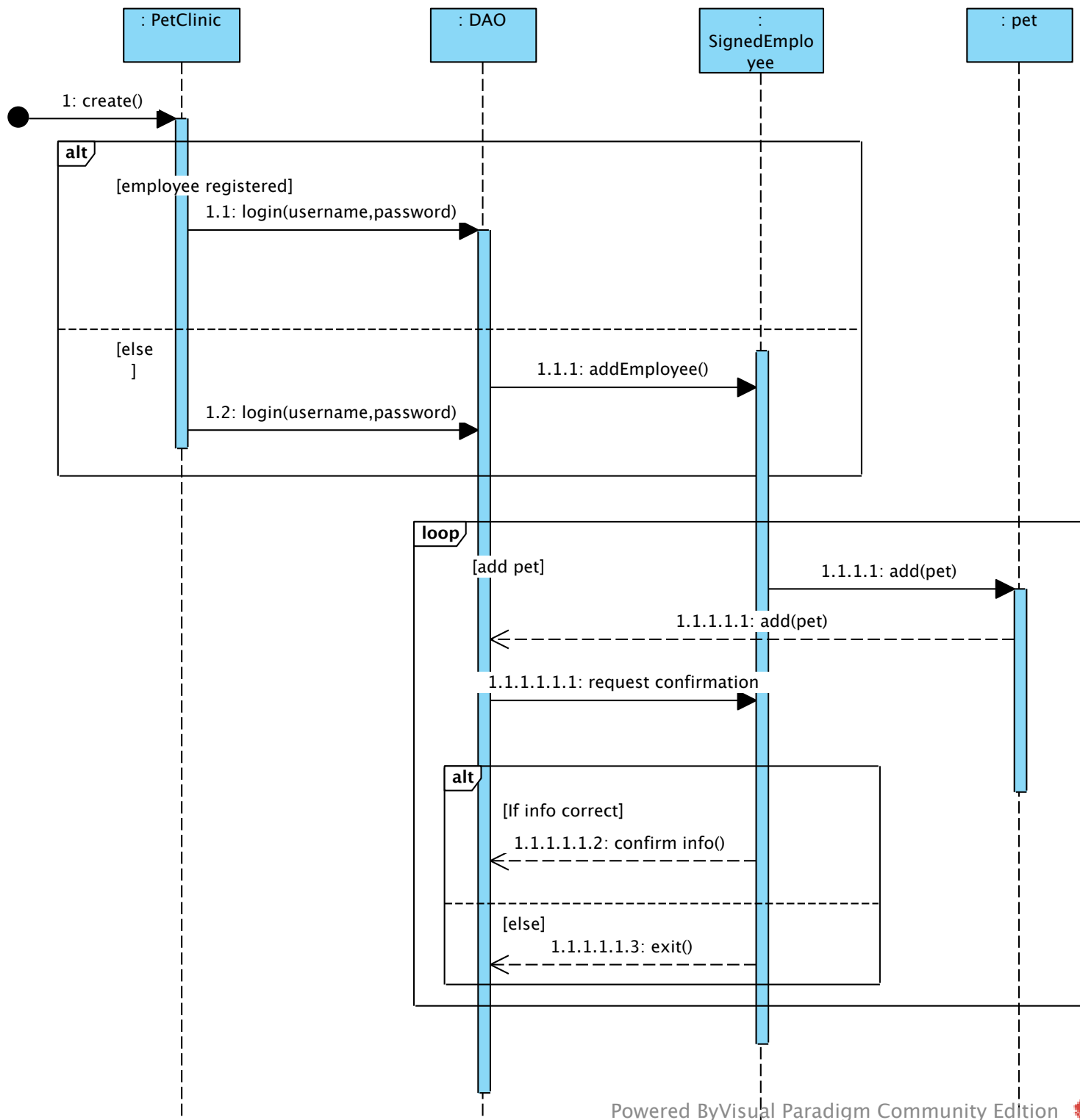




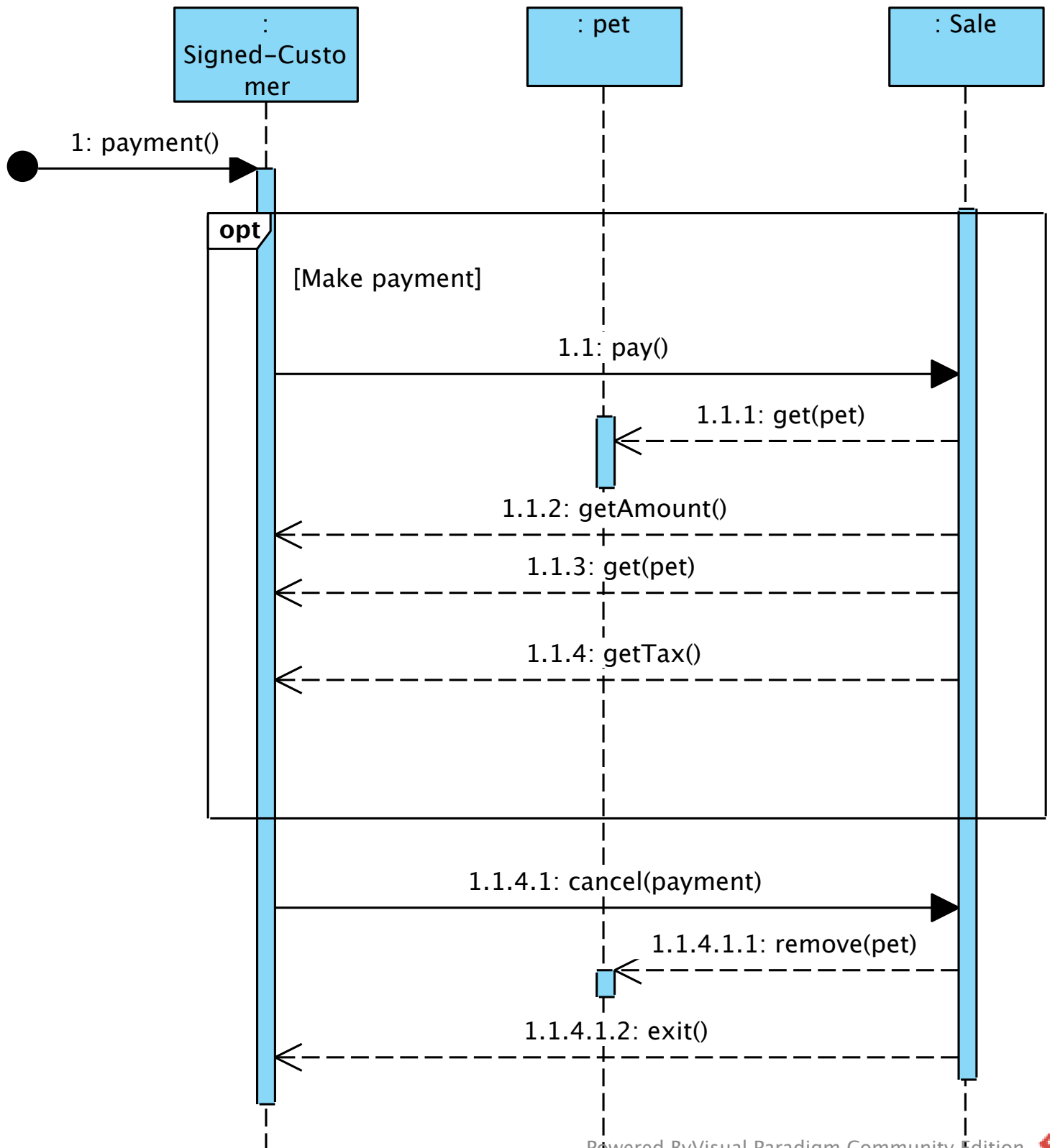
sd [SD-UC-010]

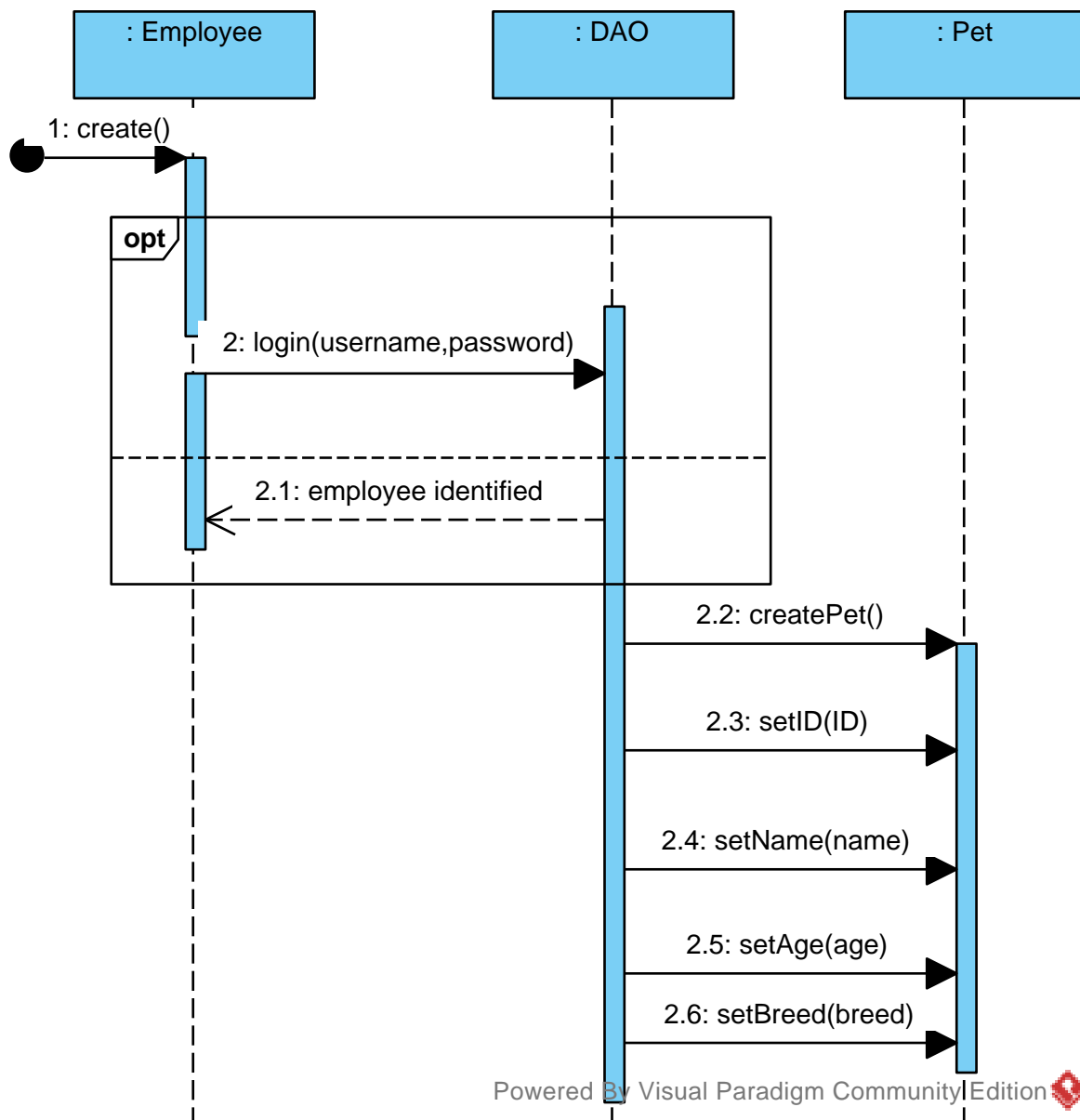


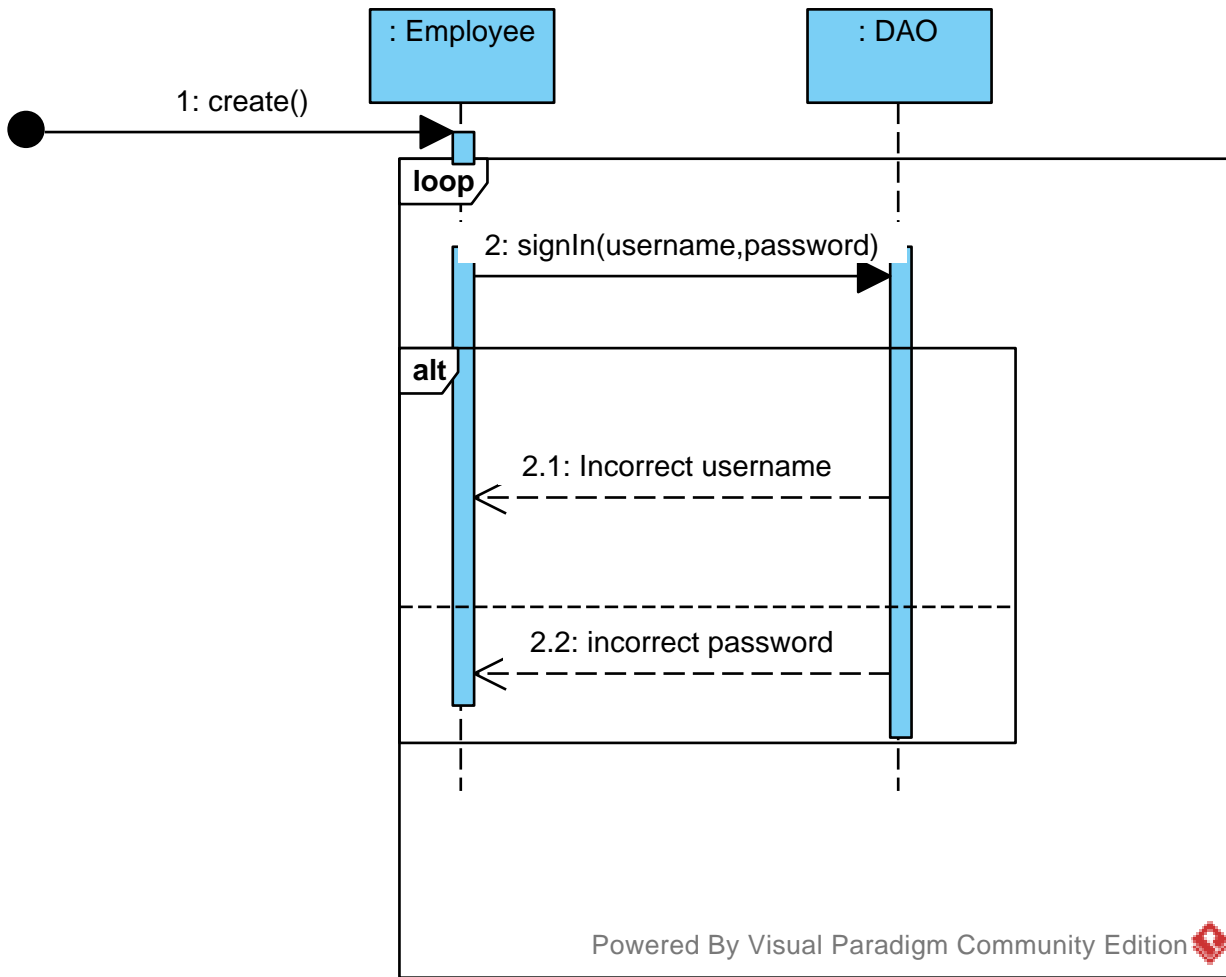
sd [SD-UC-011]

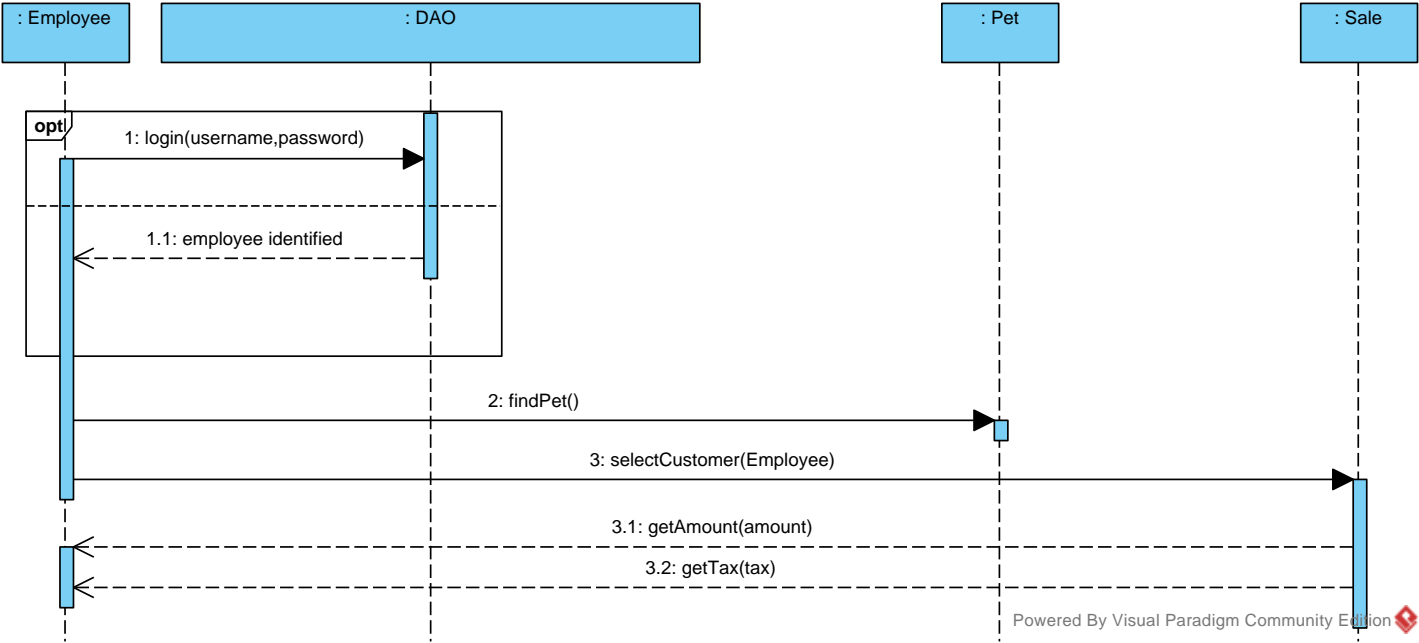


sd [SD-UC-012]









Test Coverage Plan

1. Main Menu Testing

- a. Import an existing file.
- b. Import a nonexistent file.
- c. Import a null value.
- d. Export to an existing file.
- e. Export to a new file.
- f. Export to a null value.

2. Home Page Testing

- a. Login with an existing user.
- b. Login with an invalid username and/or password.
- c. Login with null values.
- d. Attempt to open a database/inventory without logging in.
- e. Open the pet inventory.
- f. Open the employee database.
- g. Open the customer database.

3. Pet Inventory Testing

- a. Add a new, valid pet to the inventory.
- b. Attempt to add an invalid/null pet to the inventory.
- c. Edit an existing pet, giving valid edits.
- d. Edit an existing pet, giving invalid/null edits.
- e. Remove a pet from the inventory.
- f. Cancel the removal of a pet from the inventory.
- g. Save the edits.
- h. Cancel all the edits.

4. Employee Database Testing

- a. Add a new, valid employee to the database.
- b. Attempt to add an invalid/null employee to the database.
- c. Edit an existing employee, giving valid edits.
- d. Edit an existing employee, giving invalid edits.
- e. Remove an employee from the database.

- f. Cancel the removal of an employee.
- g. Save the edits.
- h. Cancel the edits.

5. Customer Database Testing

- a. Add a new, valid customer to the database.
- b. Attempt to add an invalid/null customer to the database.
- c. Edit an existing customer's information.
- d. Attempt to edit an existing customer with invalid/null edits.
- e. Remove a customer from the database.
- f. Cancel the removal of a customer.
- g. Save the edits.
- h. Cancel the edits.

6. Sale Testing

- a. Perform a sale with a customer adopting one or more pets.
- b. Perform a sale with an employee adopting one or more pets.
- c. Perform a sale with a manager adopting one or more pets.

CSI 3471 Team project

STAY_NEGATIVE

Dante Hart

Time burned: 40 hours

Available time: Every day from 4pm to 11pm

Garrett Parker

Time burned: 35 hours

Available time: 4 pm to 6 pm Monday to Saturday.

Zane Pitzer

Time burned: 32 hours

Available time: Every day from 4pm to 9pm

Yihan Zhang

Time burned: 35 hours

Available time: Every day from 12pm to 9pm

Ruiqi Zhao

Time burned: 35 hours

Available time: 4 pm to 10 pm Monday to Saturday.

Jingke Shi

Time burned: 35 hours

Available time: Every day from 4pm to 9pm

Untitled Gantt Project

2022330

<http://>

Project manager

Dante Hart

Project dates

202228 - 2022419

Completion

91%

Tasks

24

Resources

6

Tasks

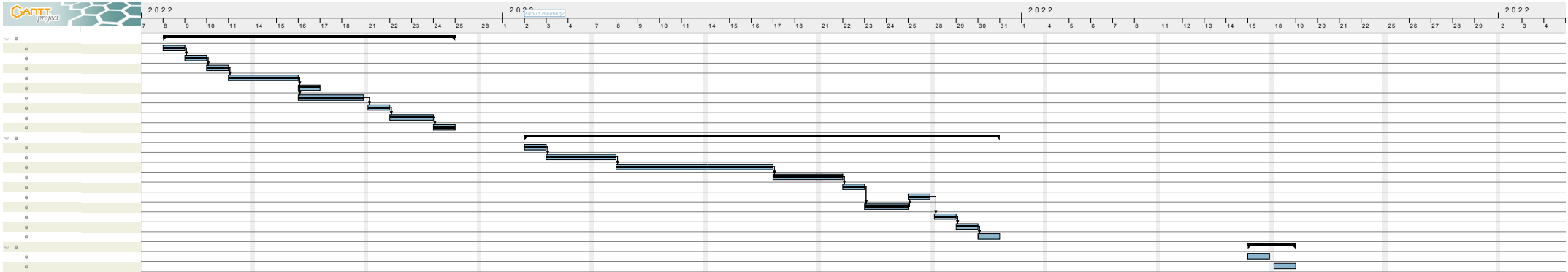
2

Name	Begin date	End date
Iteration 1	2022/2/8	2022/2/24
Group meeting 1	2022/2/8	2022/2/8
Project visition	2022/2/9	2022/2/9
requirements	2022/2/10	2022/2/10
Use cases	2022/2/11	2022/2/15
traceability matrix	2022/2/16	2022/2/16
domain model	2022/2/16	2022/2/18
User Interface wireframes	2022/2/21	2022/2/21
Gantt DDiagram	2022/2/22	2022/2/23
Presentation	2022/2/24	2022/2/24
Iteration 2	2022/3/2	2022/3/30
Group meeting2	2022/3/2	2022/3/2
Design class	2022/3/3	2022/3/7
Demo design	2022/3/8	2022/3/16
GRASP	2022/3/17	2022/3/21
Test coverage plan	2022/3/22	2022/3/22
Issur tracking system	2022/3/25	2022/3/25
Gantt update	2022/3/23	2022/3/24
Report pdf	2022/3/28	2022/3/28
Group meeting3	2022/3/29	2022/3/29
Presentation	2022/3/30	2022/3/30
Iteration 3	2022/4/15	2022/4/18
Group meeting4	2022/4/15	2022/4/15
Final Presentation	2022/4/18	2022/4/18

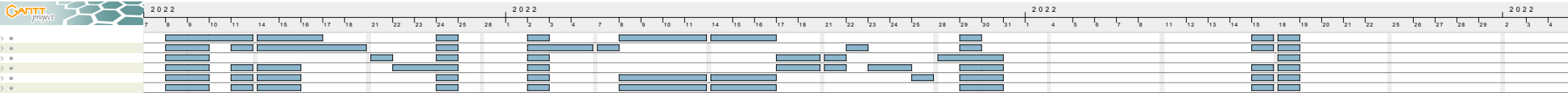
Resources

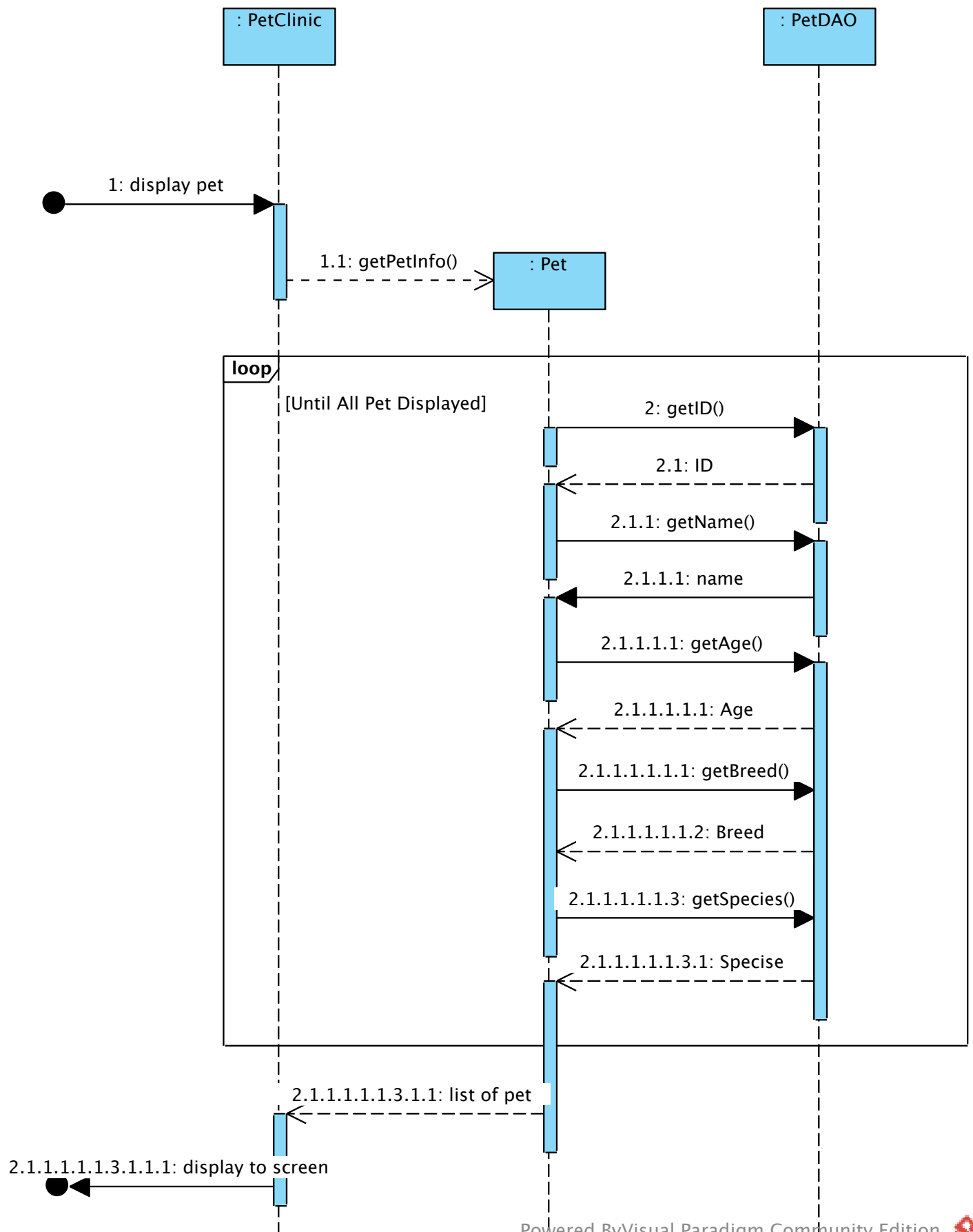
Name	Default role
Dante Hart	project manager
Garrett Parker	developer
Yihan Zhang	graphic designer
Jingke Shi	doc writer
Ruiqi Zhao	tester
Zane Pitzer	analysis

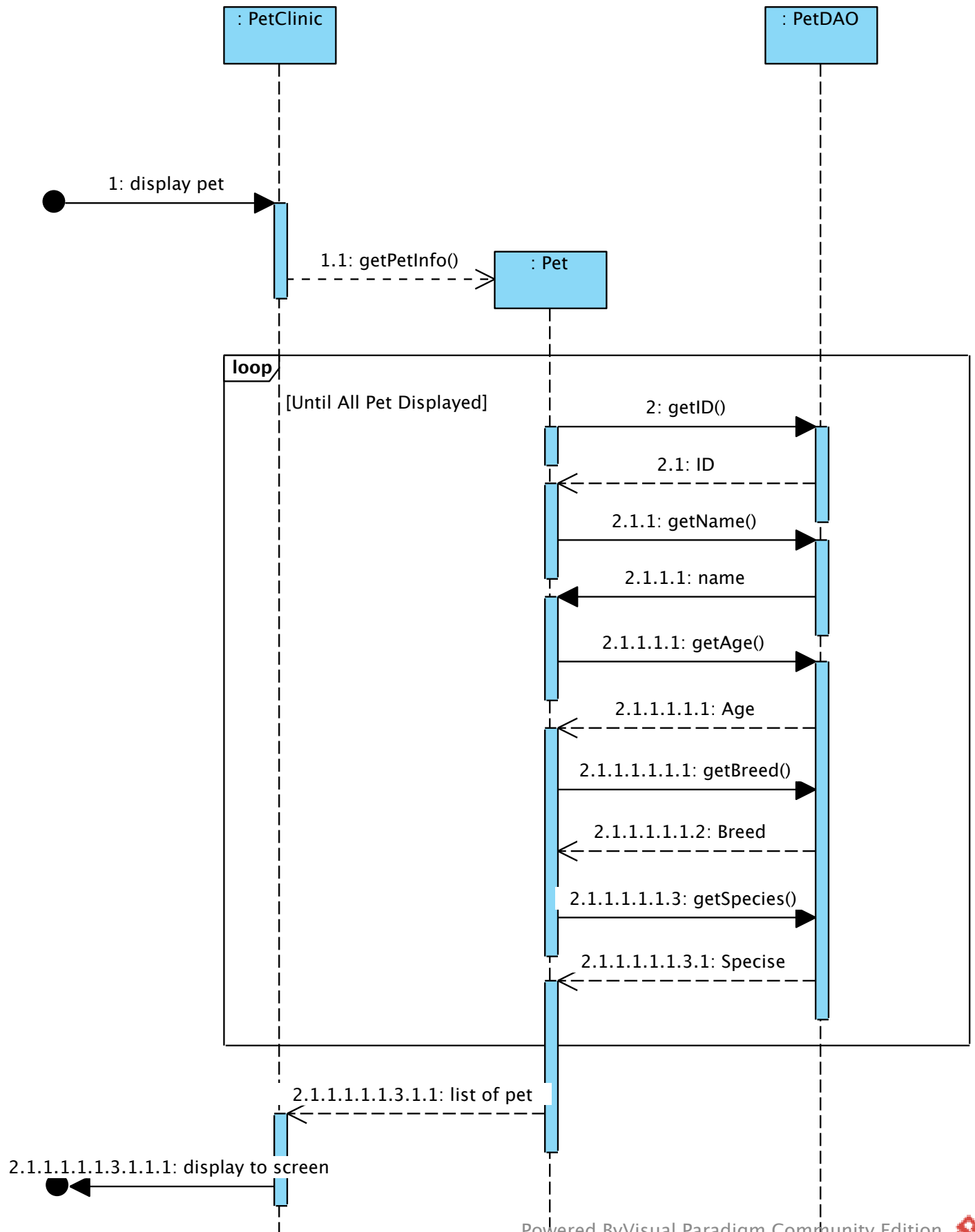
Gantt Chart



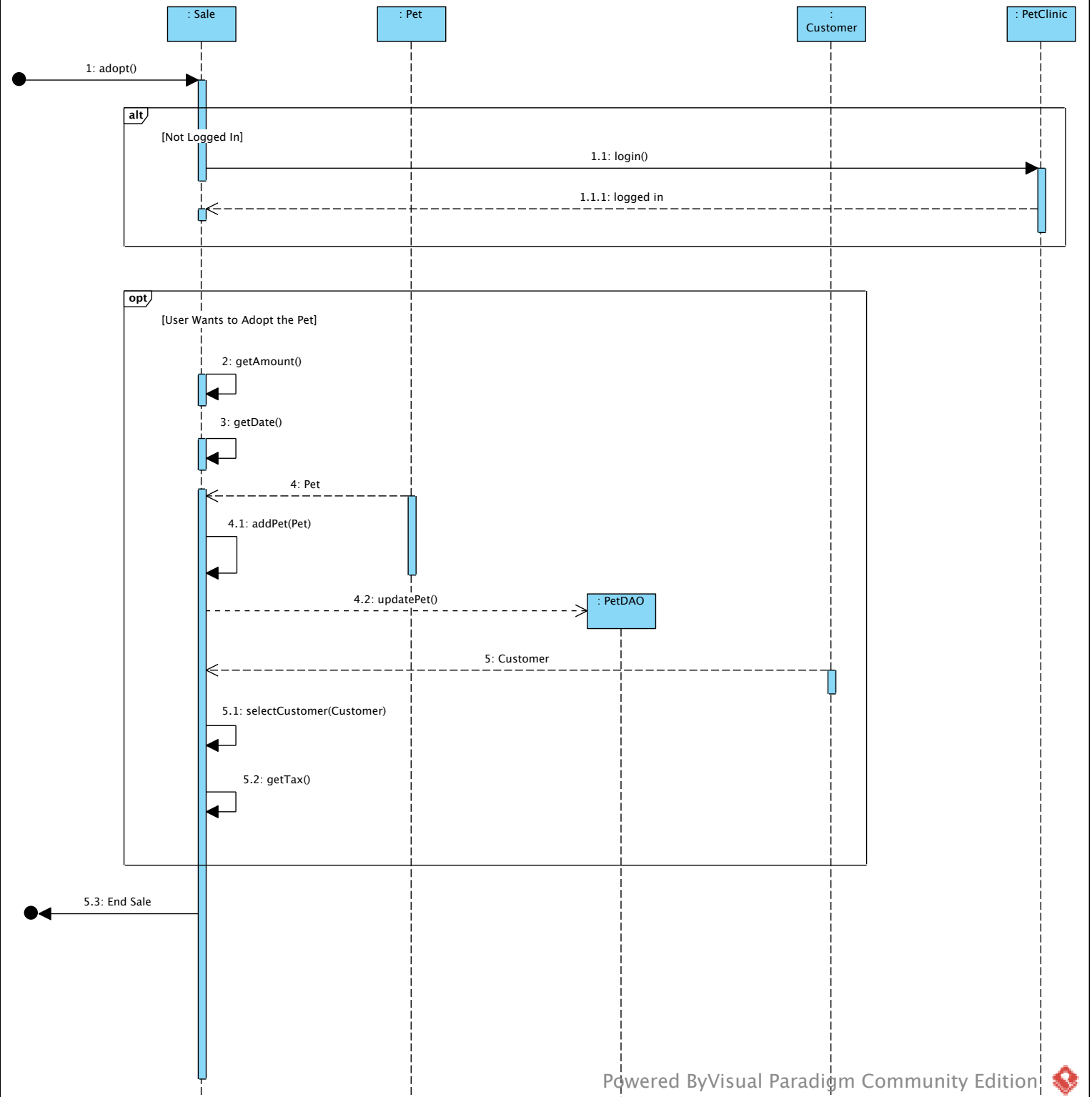
Resources Chart







sd [SD UC15]



Issue Tracking is done via

https://github.com/KillerRaptor247/STAY_Negative/issues