

# Pet Adoption Clinic

## Project vision

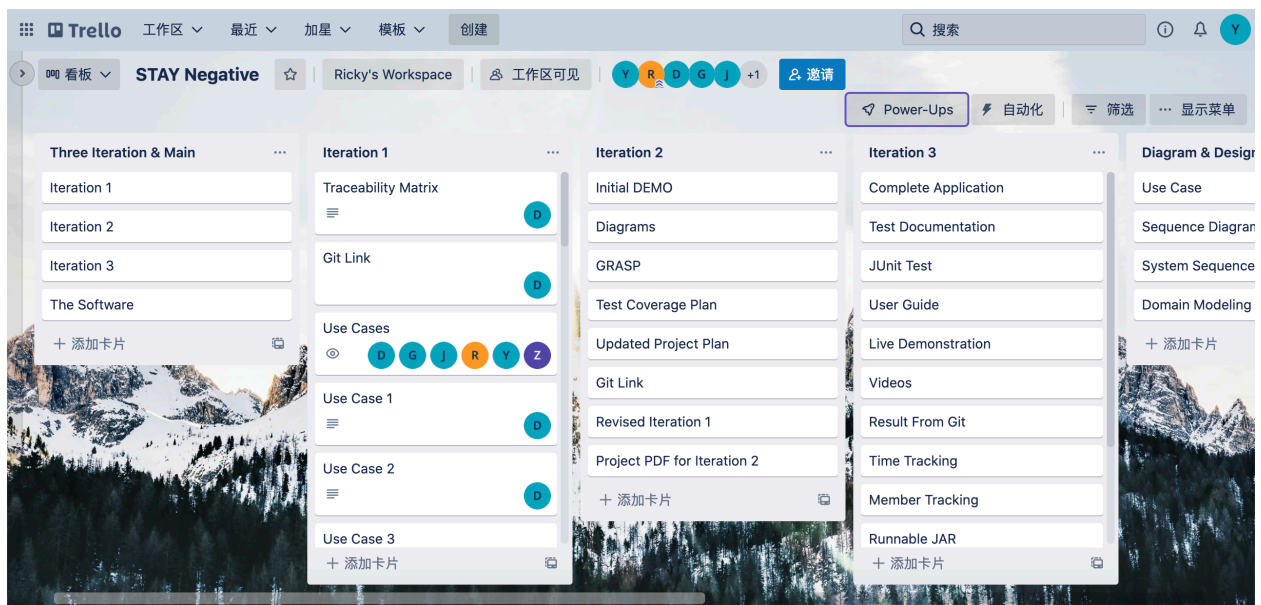
- Pet Adoption Clinic Application
- Day-to-Day Activities
- Psuedo-Database DML Transaction framework for Pets, Employees, and Customers using
- CSV files for storage
- Checkout Form
- Register Form
- Sign in Form
- Main Menu Form
- Display Form

## Team members

Dante Hart, Garret Parker, Ruiqi Zhao, Yihan Zhang, Claire Shi, Zane Pitzer

## Trello

<https://trello.com/b/v8jQjrE/stay-negative>



## REQUIREMENTS & USE CASES

1. A Pet class needs to be created and implemented
2. A Pet Catalog class needs to be created and implemented
3. A Dog class derived from Pet needs to be created and implemented
4. A Cat class derived from Pet needs to be created and implemented
5. A Customer Class needs to be created and implemented
6. A Employee Class derived from Customer needs to be created and implemented
7. There needs to be a Database class that can be derived from to better implement the database such as PETCATALOG, PETINVENTORY, CUSTOMERDATABASE, EMPLOYEEDATABASE, PETCATALOG
8. There needs to be an CustomerDatabase class to store Customers
9. There needs to be an EmployeeDatabase class to store Employees
10. An Employee or manager should be able to checkout a customer
11. A manager/admin needs to be identified in some form
12. A Login screen needs to be created
13. A login screen needs to know when there are no employees to prompt a creation
14. The System needs to be able to create an employee
15. A display menu needs to be implemented
16. A display menu needs to be able to display the inventory and catalog
17. The system needs to be able to read in a csv file of the current inventory and catalog
18. The system needs to be able to write to a csv file the current inventory and catalog before closing
19. A manager/admin needs to be able to edit the catalog and inventory databases as needed.
20. The system needs to be able to update the inventory when a pet is added or checked out
21. A display menu needs to be able to check out a customer
22. A customer needs to be able to be registered
23. A customer needs to be able to log in into the system and the system needs to be able to remember/store previous customers
24. Each pet needs to have a unique id

### USE CASES: (Make slides on these)

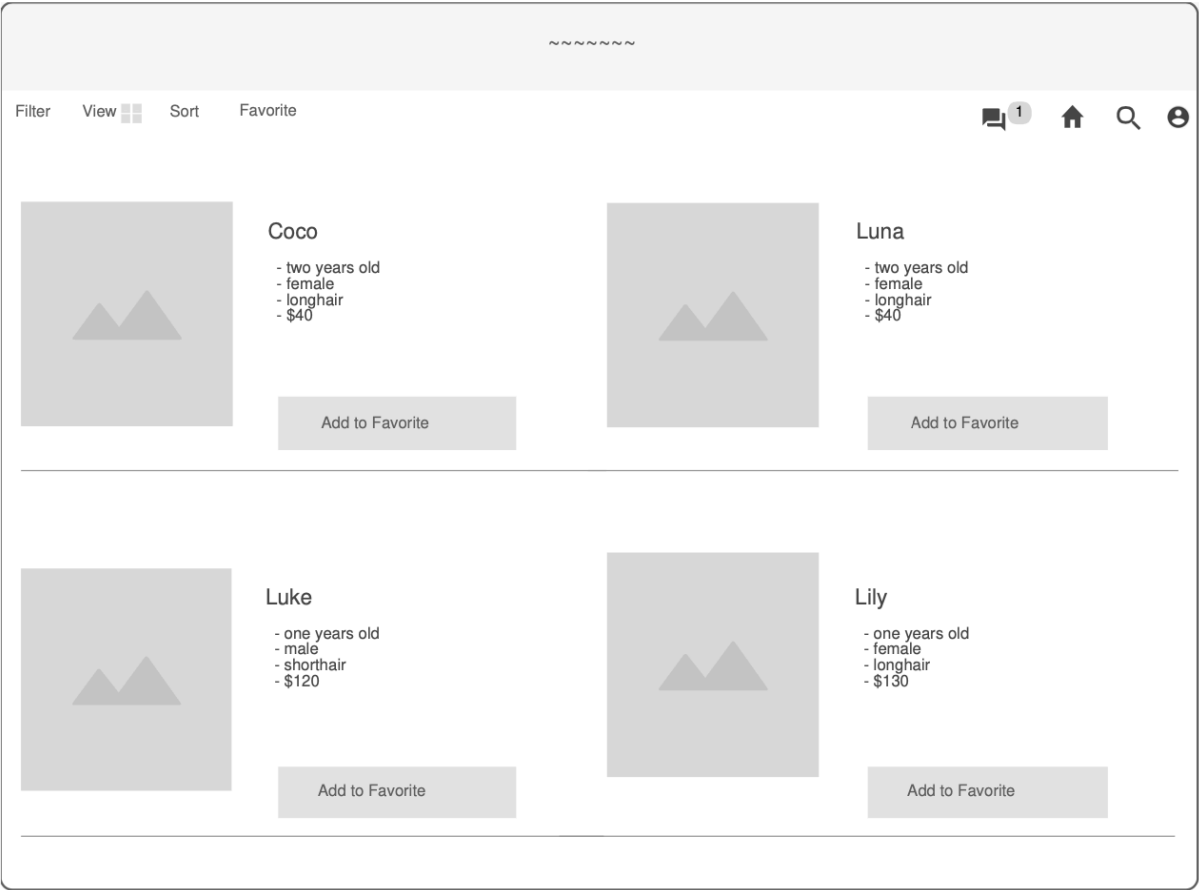
1. First Boot-up of System(No Employees are known) Manager Creation
2. Manager Creates an Employee
3. Manager edits a Pet in the inventory
4. Manager edits a Pet in the catalog
5. Manager edits an Employee in the Database
6. Manager edits a Customer in the Database
7. Employee Signs in and checks out a Customer (not registered)
8. Employee Signs in and checks out a Customer (registered)
9. Manager Signs in and checks out A customer (not registered)
10. Manager Signs in and checks out A Customer (registered)
11. A pet is added to the catalog by the manager
12. A customer changes their mind and decides to not purchase a pet mid-transaction

13. An employee desires to display the inventory to a customer
14. A manager desires to display the inventory to a customer
15. An employee desires to adopt a pet of their own
16. A manager desire to adopt a pet of their own
17. An employee/manager signs in but gives incorrect information
18. A Customer tries to sign in but gives incorrect information
19. The system application is closed
20. A pet is checked into the clinic by a manager/admin

<https://docs.google.com/spreadsheets/d/1smMC4bsRIYySg8JQKhltAFDM7UAiuPWjri5WoL92GD0/edit#gid=1398917863>

[illegible]

# User interface wireframes



# Untitled Gantt Project

2022223

<http://>

Project manager

Dante Hart

Project dates

202228 - 202225

Completion

100%

Tasks

10

Resources

6

Tasks

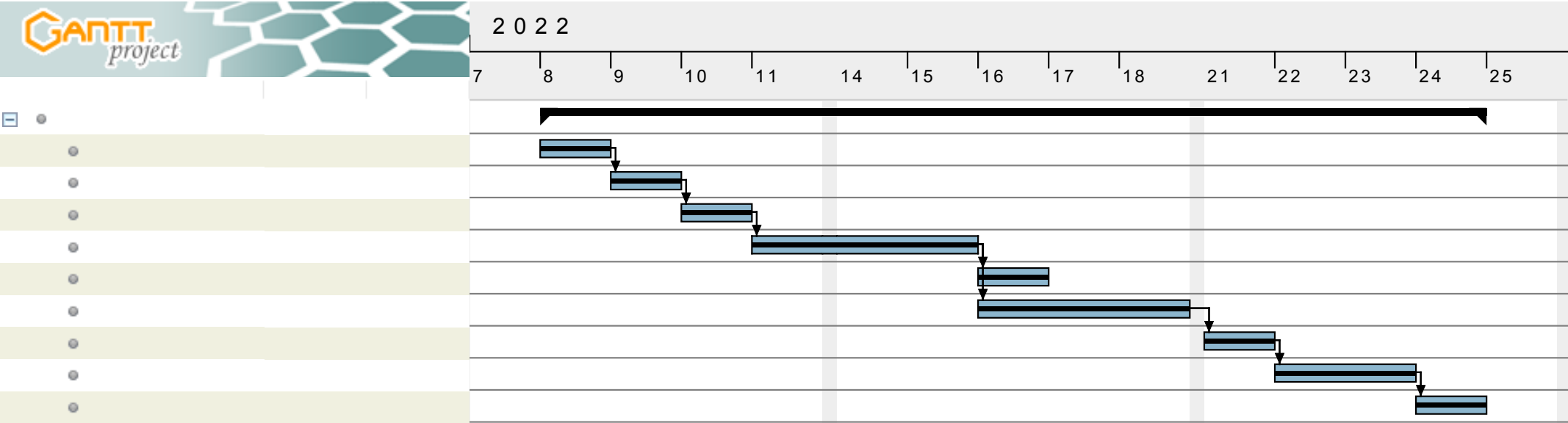
Name	Begin date	End date
Iteration 1	2022/2/8	2022/2/24
Group meeting 1	2022/2/8	2022/2/8
Project vision	2022/2/9	2022/2/9
requirements	2022/2/10	2022/2/10
Use cases	2022/2/11	2022/2/15
traceability matrix	2022/2/16	2022/2/16
domain model	2022/2/16	2022/2/18
User Interface wireframes	2022/2/21	2022/2/21
Gantt DDiagram	2022/2/22	2022/2/23
Presentation	2022/2/24	2022/2/24

Resources

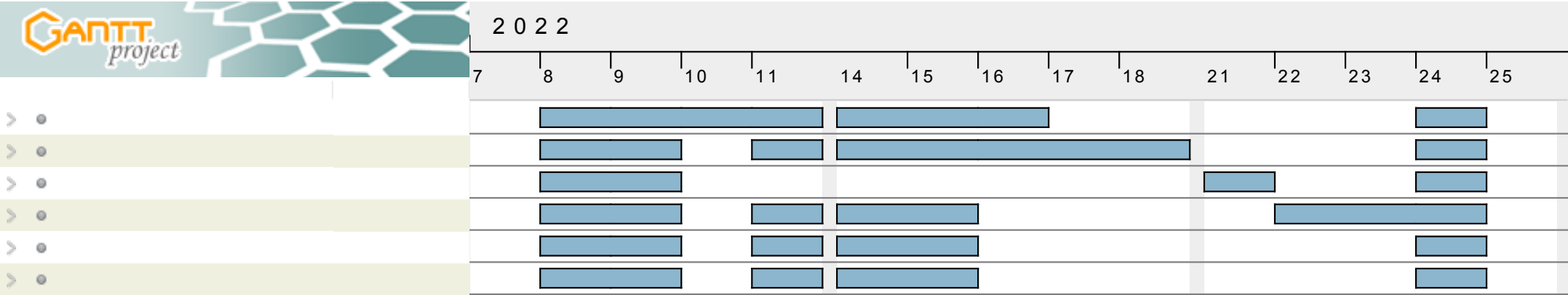
Name	Default role
Dante Hart	project manager
Garrett Parker	developer
Yihan Zhang	graphic designer
Jingke Shi	doc writer
Ruiqi Zhao	tester
Zane Pitzer	analysis

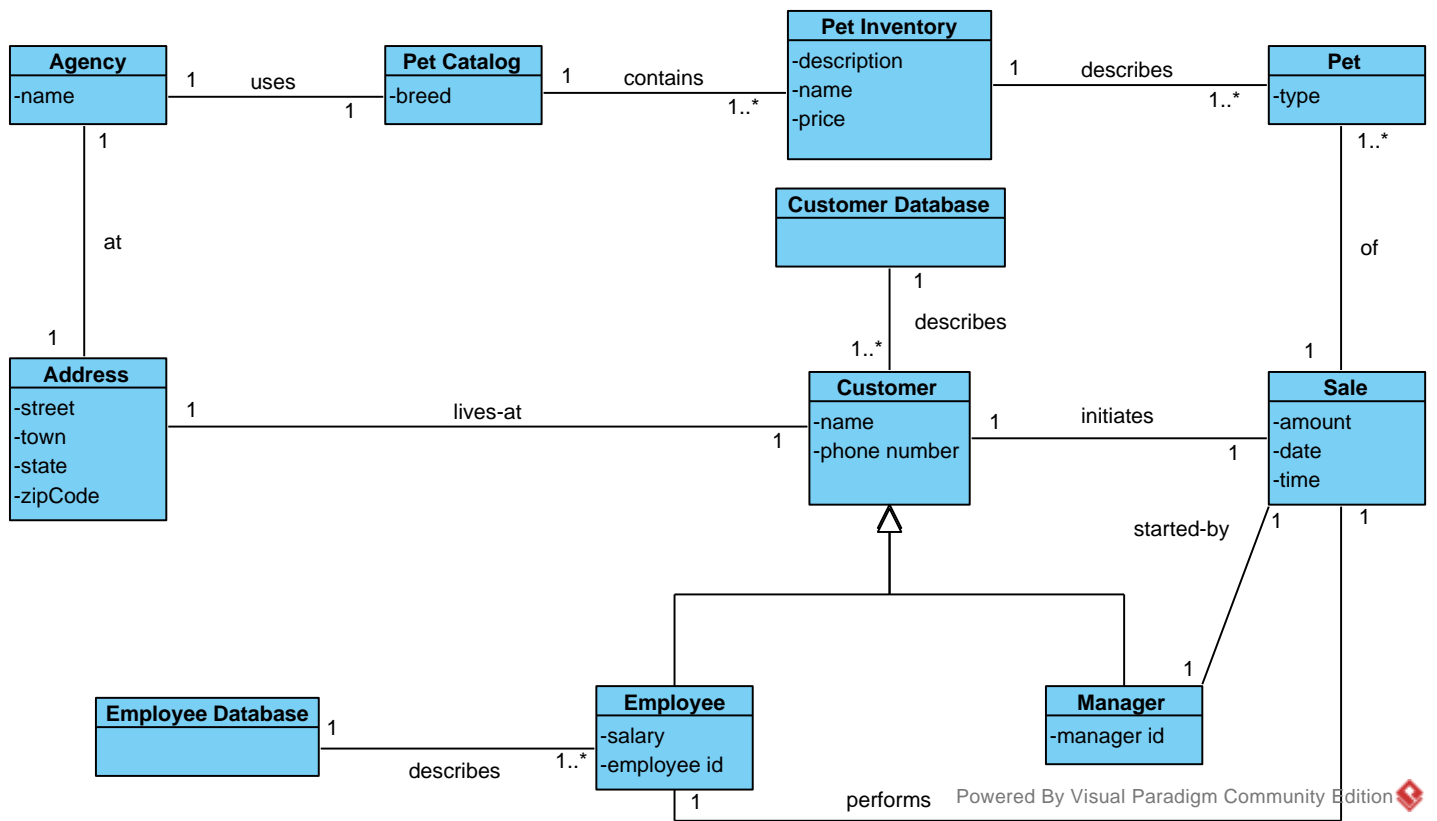


Gantt Chart

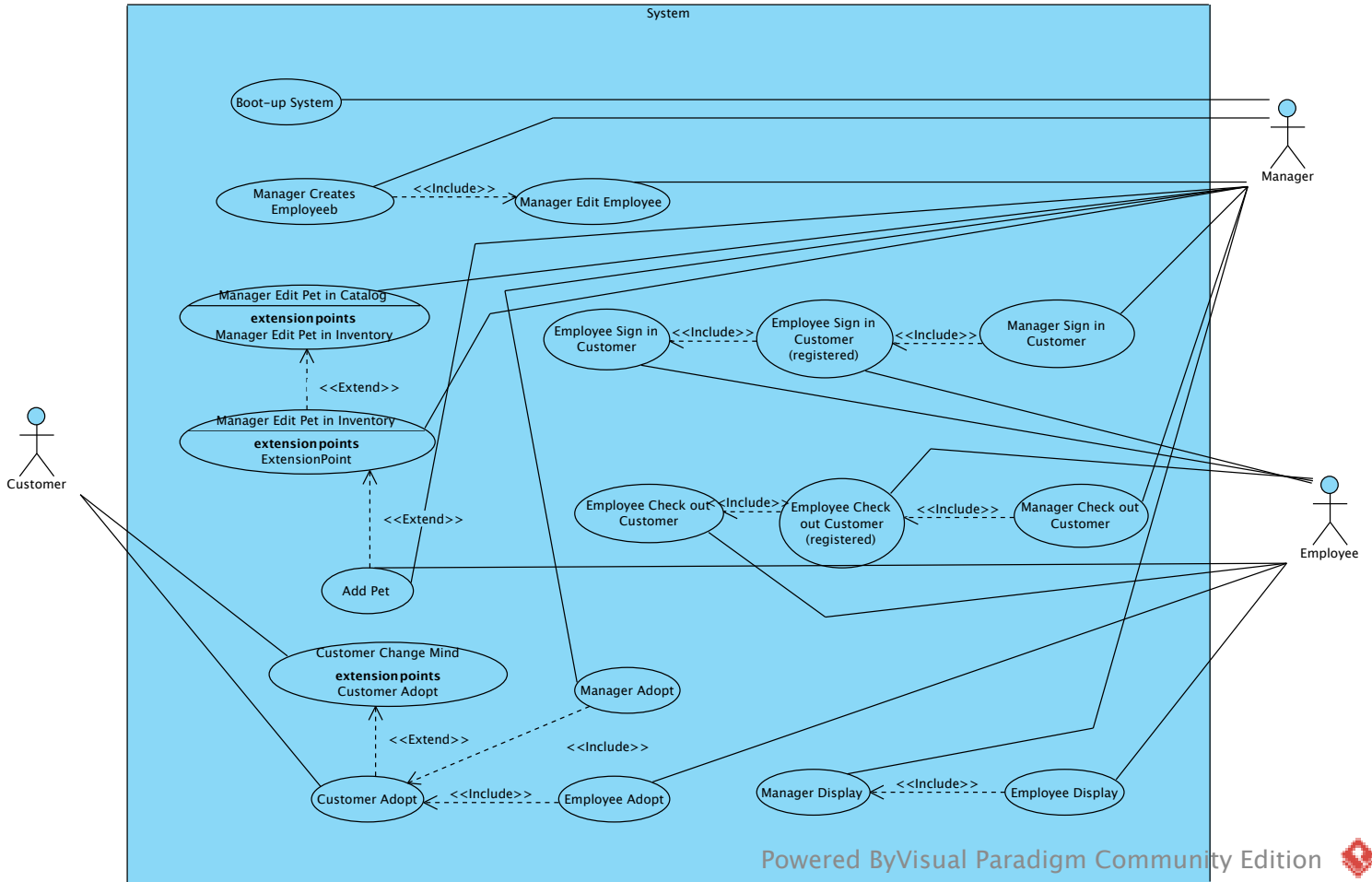


Resources Chart





uc [Use Case Diagram1]



## **Timecards**

CSI 3471 Team project

STAY\_NEGATIVE

### **Dante Hart**

Time burned: 12 hours

Available time: Every day from 4pm to 11pm

### **Garrett Parker**

Time burned: 10 hours

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

### **Zane Pitzer**

Time burned: 9.5 hours

Available time: Every day from 4pm to 9pm

### **Yihan Zhang**

Time burned: 10 hours

Available time: Every day from 12pm to 9pm

### **Ruiqi Zhao**

Time burned: 10 hours

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

### **Jingke Shi**

Time burned: 10 hours

Available time: Every day from 4pm to 9pm

use cases 1-3 Dante Hart

ID:	UC 001
Title:	First Boot-up of System
Description:	A first clean start-up of a system where no Employees are known.
Primary Actor:	Manager/Employee
Preconditions:	The system does not have any Employees in its database
Postconditions:	An Employee is created for the system to continue normal functions
Main Success Scenario:	<div>1. A user starts the application</div> <div>2. The System checks the employee database and detects there are no employees</div> <div>3. Instead of prompting a sign-in, The system prompts for a Manager registration.</div> <div>4. The Manager creates an Employee with the title of Manager in the Employee database.</div> <div>5. The register form closes and the login form appears for the Manager to login</div>
Extensions:	<div>2.a If there are employees found, the login form appears instead</div> <div>3.a If the Manager closes the form without a complete registration, and error message will be displayed and the application will close.</div>
Frequency of Use:	Should be a single occurrence on a successful use. Could be continuous based on user refusal to successfully create an employee for whatever reason
Status:	In Development
Owner:	Dante Hart
Priority:	MEDIUM

ID:	UC 002
Title:	Manager creates an Employee
Description:	A manager is creating another employee for the database

<b>Primary Actor:</b>	Manager
<b>Preconditions:</b>	A Manager has signed into the application and is on their main menu screen
<b>Postconditions:</b>	An Employee has been successfully created
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. A manager clicks on a button that takes them to the create employee form</li><li>2. An employee puts in information for an employee into the system and presses a button to commit the changes</li><li>3. Upon valid information, an Employee is added into the database</li><li>4. The form closes back to the main menu and the Employee Database is updated</li></ol>
<b>Extensions:</b>	2.a If Incorrect information is given, an error message is displayed and an Employee's information is not saved.
<b>Frequency of Use:</b>	Estimation of use is moderate use.
<b>Status:</b>	In Progress
<b>Owner:</b>	Dante Hart
<b>Priority:</b>	MEDIUM

Template:

ID:	UC 003
Title:	Manager edits a pet in the inventory.
Description:	A manager edits a pet in the Pet Inventory database. A manager can change a pets information outside of the unique ID. Additionally a manager can remove a pet from the inventory or add one as needed.
Primary Actor:	Manager
Preconditions:	A Manager has signed into the application and is on their main menu screen
Postconditions:	A Pet in the Inventory has been edited in some way
Main Success Scenario:	<div>1. A manager clicks the Edit Inventory button on the main menu screen</div> <div>2. A Form appears that displays the Pet inventory to be edited</div> <div>3. The manager edits a field for a Pet and clicks a save button</div> <div>4. The pet is saved and the Edit form is closed</div> <div>5. The pet database is updated.</div>
Extensions:	<div>3.a) If invalid information is input an error message is displayed</div>
Frequency of Use:	Estimation of use is moderate use.
Status:	In Progress
Owner:	Dante Hart
Priority:	MEDIUM



## use cases 4-6 Garret Parker

<b>ID:</b>	<b>UC 004</b>
<b>Title:</b>	Edit Pet Catalog
<b>Description:</b>	A Manager edits a pet in the pet catalog.
<b>Primary Actor:</b>	Manager
<b>Preconditions:</b>	A valid manager is logged into the system.
<b>Post-conditions:</b>	The pet catalog is updated as per the manager's edits.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The Manager navigates to the pet catalog.</li> <li>2. The System displays the pet catalog.</li> <li>3. The Manager edits the catalog (add, delete, or update a pet type.)</li> <li>4. The System displays the edit.</li> <li>5. The Manager repeats steps 3-4 until all the edits have been made.</li> <li>6. The Manager notifies the System they have finished making edits.</li> <li>7. The System asks the Manager to confirm their identity.</li> <li>8. The Manager enters their username and password.</li> <li>9. The System confirms the person editing the catalog is a Manager.</li> <li>10. The System displays a list of the edits made.</li> <li>11. The Manager confirms the edits.</li> <li>12. The System confirms the edits have been made.</li> <li>13. The Manager logs out.</li> </ol>
<b>Extensions:</b>	<ol style="list-style-type: none"> <li>1. If the Manager's identity is not confirmed after the edits are made, the System discards the edits.</li> <li>2. If the Manager does not confirm the edits, they can return to the catalog and continue making edits.</li> </ol>
<b>Frequency of Use:</b>	When a new pet arrives, a pet is adopted, or information regarding a pet type needs to be updated.
<b>Status:</b>	In Progress
<b>Owner:</b>	Garrett Parker
<b>Priority:</b>	<b>MEDIUM</b>

<b>ID:</b>	UC 005
<b>Title:</b>	Edit Employee Database
<b>Description:</b>	A Manager edits an employee in the database.
<b>Primary Actor:</b>	Manager
<b>Preconditions:</b>	A valid Manager is logged into the system.
<b>Post-conditions:</b>	The employee database is updated successfully.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The Manager navigates to the employee database.</li> <li>2. The System displays the employee database.</li> <li>3. The Manager edits the database (add, remove, or edit employee.)</li> <li>4. The System displays the edit.</li> <li>5. Repeat steps 3 and 4 until the edits have been finished.</li> <li>6. The Manager notifies the System they have finished making edits.</li> <li>7. The System asks the Manager to confirm their identity.</li> <li>8. The Manager enters their username and password.</li> <li>9. The System confirms the person editing the database is a Manager.</li> <li>10. The System displays a list of the edits made.</li> <li>11. The Manager confirms the edits.</li> <li>12. The System confirms the edits have been made.</li> <li>13. The Manager logs out.</li> </ol>
<b>Extensions:</b>	<ol style="list-style-type: none"> <li>1. If the Manager's identity is not confirmed after the edits are made, the System discards the edits.</li> <li>2. If the Manager does not confirm the edits, they can return to the database and continue making edits.</li> </ol>
<b>Frequency of Use:</b>	When a new employee is hired, an employee is fired, or an employee needs their information updated.
<b>Status:</b>	In Progress
<b>Owner:</b>	Garrett Parker
<b>Priority:</b>	MEDIUM

<b>ID:</b>	UC 006
<b>Title:</b>	Edit Customer Database
<b>Description:</b>	A Manager edits a customer in the database.
<b>Primary Actor:</b>	Manager
<b>Preconditions:</b>	A valid Manager is logged into the system.
<b>Post-conditions:</b>	The customer base has been successfully updated.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The Manager navigates to the customer database.</li> <li>2. The System displays the customer database.</li> <li>3. The Manager edits the database (assign pets, update information.)</li> <li>4. The System displays the edit.</li> <li>5. Repeat steps 3 and 4 until all the edits have been completed.</li> <li>6. The Manager notifies the System they have finished making edits.</li> <li>7. The System asks the Manager to confirm their identity.</li> <li>8. The Manager enters their username and password.</li> <li>9. The System confirms the person editing the database is a Manager.</li> <li>10. The System displays a list of the edits made.</li> <li>11. The Manager confirms the edits.</li> <li>12. The System confirms the edits have been made.</li> <li>13. The Manager logs out.</li> </ol>
<b>Extensions:</b>	<ol style="list-style-type: none"> <li>1. If the Manager's identity is not confirmed, the edits are ignored.</li> <li>2. If the list of edits is not correct, the Manager can go back and fix mistakes.</li> </ol>
<b>Frequency of Use:</b>	When a customer adopts an animal, the customer database needs to be updated. When a customer needs to change their information.
<b>Status:</b>	In Progress
<b>Owner:</b>	Garrett Parker
<b>Priority:</b>	MEDIUM

## Use cases 7-9 Yihan Zhang

<b>ID:</b>	<b>UC 007</b>
<b>Title:</b>	Employee Signs in and checks out a Customer(not registered)
<b>Description:</b>	An Employee Signs in and checks out a Customer who is not registered.
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The Employee has registered in the system and the system has the information of the Employee in the database. The Customer is not registered.
<b>Post-conditions:</b>	A Customer who is not registered is checked out by an Employee.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. A Customer who is not registered asks for checking out.</li> <li>2. An Employee comes and offer help.</li> <li>3. The Employee sign in to the system with its username and password.</li> <li>4. The Employee asks the Customer to create an account.</li> <li>5. The customer provide its information.</li> <li>6. The Employee helps to register for the Customer.</li> <li>7. The Employee creates a new sale for the Customer.</li> <li>8. The Employee enters information about what the Customer wants to adopt.</li> <li>9. The system returns the total costs plus tax.</li> <li>10. The Employee asks for payment method.</li> <li>11. The Customer paid.</li> <li>12. The system returns the receipt.</li> <li>13. The Employee ends the sale.</li> </ol>
<b>Extensions:</b>	<ol style="list-style-type: none"> <li>1. If the Employee has not registered in the system, register now.</li> <li>2. If the Customer is registered, apply UC 008.</li> <li>3. If the Customer waives payment, the Employee closes the sale.</li> <li>4. If the Customer do not want to create an account, skip to step 7.</li> </ol>
<b>Frequency of Use:</b>	May be considered to use every time a customer wants to check out depend on the preconditions.
<b>Status:</b>	In Progress
<b>Owner:</b>	Yihan Zhang
<b>Priority:</b>	<b>MEDIUM</b>

<b>ID:</b>	<b>UC 008</b>
<b>Title:</b>	Employee Signs in and checks out a Customer(registered)
<b>Description:</b>	An Employee Signs in and checks out a Customer who is registered.
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The Employee has registered in the system and the system has the information of the Employee in the database. The Customer is registered.
<b>Post-conditions:</b>	A Customer who is registered is checked out by an Employee.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. A Customer who is registered asks for checking out.</li> <li>2. An Employee comes and offer help.</li> <li>3. The Employee sign in to the system with its username and password.</li> <li>4. The Employee creates a new sale for the Customer.</li> <li>5. The Employee asks for the Customers account information.</li> <li>6. The Employee views the Customer's shopping cart.</li> <li>7. The Employee selects items that the Customer wants to buy and check out.</li> <li>8. The system returns the total costs plus tax.</li> <li>9. The Employee asks for payment method.</li> <li>10.The Customer paid.</li> <li>11. The system returns the receipt.</li> <li>12.The Employee ends the sale.</li> </ol>
<b>Extensions:</b>	<ol style="list-style-type: none"> <li>1. If the Employee has not registered in the system, register now.</li> <li>2. If the Customer is not registered, apply UC 007.</li> <li>3. If the Customer waives payment, the Employee closes the sale.</li> </ol>
<b>Frequency of Use:</b>	May be considered to use every time a customer wants to check out depend on the preconditions.
<b>Status:</b>	In Progress
<b>Owner:</b>	Yihan Zhang
<b>Priority:</b>	<b>MEDIUM</b>

<b>ID:</b>	<b>UC 009</b>
<b>Title:</b>	Manager Signs in and checks out A customer (not registered)
<b>Description:</b>	A Manager Signs in and checks out a Customer who is not registered.
<b>Primary Actor:</b>	Manager
<b>Preconditions:</b>	The Manager has registered in the system and the system has the information of the Manager in the database. The Customer is not registered.
<b>Post-conditions:</b>	A Customer who is not registered is checked out by a Manager.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. A Customer who is not registered asks for checking out.</li> <li>2. A Manager comes and offer help.</li> <li>3. The Manager sign in to the system with its username and password.</li> <li>4. The Manager asks the Customer to create an account.</li> <li>5. The Customer provide its information.</li> <li>6. The Manager helps to register for the Customer.</li> <li>7. The Manager creates a new sale for the Customer.</li> <li>8. The Manager enters information about what the Customer wants to buy.</li> <li>9. The system returns the total costs plus tax.</li> <li>10. The Manager asks for payment method.</li> <li>11. The Customer paid.</li> <li>12. The system returns the receipt.</li> <li>13. The Manager ends the sale.</li> </ol>
<b>Extensions:</b>	<ol style="list-style-type: none"> <li>1. If the Manager has not registered in the system, register now.</li> <li>2. If the Customer is registered, apply UC 010.</li> <li>3. If the Customer waives payment, the Manager closes the sale.</li> <li>4. If the Customer do not want to create an account, skip to step 7.</li> </ol>
<b>Frequency of Use:</b>	May be considered to use every time a customer wants to check out depend on the preconditions.
<b>Status:</b>	In Progress
<b>Owner:</b>	Yihan Zhang
<b>Priority:</b>	<b>MEDIUM</b>

Use case: Manager Signs in and checks out A Customer (registered)

ID:	UC010
Title:	Manager Signs in and checks out A Customer(registered)
Description:	This use case describes the process of how a manager help the registered customer checks out
Primary Actor:	Manager
Preconditions:	The employee has registered into system as manager, and the customer has registered before
Flow of events	1. The manager sign in to the system 2. The system checks manager’s identity 3. The manager starts application for the adoption process 4. The system searches adoption info and registered customer info 5. The manager confirm the info 6. The system requests payment
Postconditions:	1. The customer may change the animal he/she want to adopt rather than the featured one
Alternative flow:	1. At any point the customer may leave, then manager exit the process and leave

Use case: A pet is added to the catalog by the manager

ID:	UC011
Title:	A pet is added to the catalog by the manager
Description:	This use case describes the process of how a manager add a pet (and related info) into catalog and
Primary Actor:	Manager
Preconditions:	The employee has registered into system as manager and sign in to system
Flow of events	1. The manager chooses add view 2. The system redirects to add pet form page 3. The manager fills-out the info form 4. The system requests confirmation 5. The manager checks the info typed before and confirm a) If the info is wrong, correct it and re-submit the form
Postconditions:	The pet might be added by other manager
Alternative flow:	At any point the manager may exit the process and leave



Use case: A customer changes his mind and decides to not purchase a pet mid-transaction

ID:	UC012
Title:	A customer changes his mind and decides to not purchases a pet mid-transaction
Description:	This use case describes the process of during the transaction of adopting process, the customer chooses to end purchasing.
Primary Actor:	Manager
Preconditions:	The customer has registered and decides to apply adopting process.
Flow of events	1. The manager starts application for the adoption process 2. The system searches adoption info and registered customer info 3. The manager confirm the info 4. The system requests payment 5. The customer changes his mind, then manager cancel the process
Postconditions:	The customer might change his mind and apply process of adoption again
Alternative flow:	The customer might change his mind after payment, then manager need to apply refund and mark the pet available

**ID: UC 013**

**Title: Employee Display**

**Description:**

To show the pet and accessories for pet, like pet food, pet toy to the customer which is the one using our app.

**Primary Actor:**

Employee

**Precondition:**

The employee has already registered in the system and the has all information about the user login informations.

The inventory the employee wants to display is on the system or in stock.

**Main Flow:**

1. Employee chooses to display the inventory.
2. Employee ask manager for permission
  - 2.1. Manager refuse to display the inventory
3. Employee chooses the inventory that he wants to display
4. Employee get that inventory from the database
5. Employee provide the information of the inventory
6. Employee give an advertising sentence for the inventory
7. Employee post the inventory to the app

**Frequency of Use:**

Consider as an optional event, that only occur when the employee decides to display the inventory to the customer.

**Status:**

In progress

**Owner:**

Ruiqi Zhao

**Priority:**

LOW

**ID: UC 014**

**Title: Manager Display**

**Description:**

To show the pet and accessories for pet, like pet food, pet toy to the customer which is the one using our app.

**Primary Actor:**

Manager

**Precondition:**

The manager has already registered in the system and the has all information about the user login informations.

The inventory the manager wants to display is on the system or in stock.

**Main Flow:**

1. Manager chooses to display the inventory.
2. Manager ask manager for permission
3. Manager chooses the inventory that he wants to display
4. Manager get that inventory from the database
5. Manager provide the information of the inventory
6. Manager give an advertising sentence for the inventory
7. Manager post the inventory to the app

**Frequency of Use:**

Consider as an optional event, that only occur when the manager decides to display the inventory to the customer.

**Status:**

In progress

**Owner:**

Ruiqi Zhao

**Priority:**

LOW

**ID: UC 015**

**Title: Employee Adopt**

**Description:**

When the employee likes a pet and wants to adopt for their own.

**Primary Actor:**

Employee

**Precondition:**

The pets are listed on the app and the information of the pet are stored in the database.

The employee likes the pet and want to adopt the pet.

**Post-condition:**

The pet which been adopted will not be list on the app anymore, and will be hide from the databases.

**Main Flow:**

1. Employee chooses the pet he/she wants to adopted.
2. Check the information of the pet
  - 2.1. If he decide not to adopt this pet, go back
3. Go to check out
4. Pay for the adoption fee
  - 4.1. Choose payment method
  - 4.2. If transaction didn't go through
    - 4.2.1. choose another payment method
5. The pet is successfully adopted
6. System will give customer a number to let user bring the pet home

**Frequency of Use:**

Will be frequently using, the adoption is the main part of the software, so this should be fully functional and useful.

**Status:**

In progress

**Owner:**

Ruiqi Zhao

**Priority:**

High

use cases 20, 17, 16 Zane Pitzer

<b>ID:</b>	<b>UC 020</b>
<b>Title:</b>	A pet is checked into the clinic by a manager
<b>Description:</b>	A new pet is brought to the clinic to be put up for adoption
<b>Primary Actor:</b>	manager
<b>Preconditions:</b>	The manager has registered in the system and is logged into the system
<b>Post-conditions:</b>	A new pet is added to the clinic and is ready to be adopted
<b>Main Flow:</b>	<ol style="list-style-type: none"> <li>1. Manager selects to add a new pet to the system</li> <li>2. Manager adds new pets name</li> <li>3. Manager adds new pets known or estimated age</li> <li>4. Manager adds new pets sex</li> <li>5. Manager adds new pets type(cat or dog)</li> <li>6. Manager decides on and adds new pets price</li> <li>7. Manager adds a picture of the new pets</li> <li>8. Manager selects to complete process and adds the new pet to the system</li> </ol>
<b>Frequency of Use:</b>	1. Will be used every time a new pet is brought into the clinic and is checked in by a manager
<b>Status:</b>	In Progress
<b>Owner:</b>	Zane Pitzer
<b>Priority:</b>	<b>MEDIUM</b>

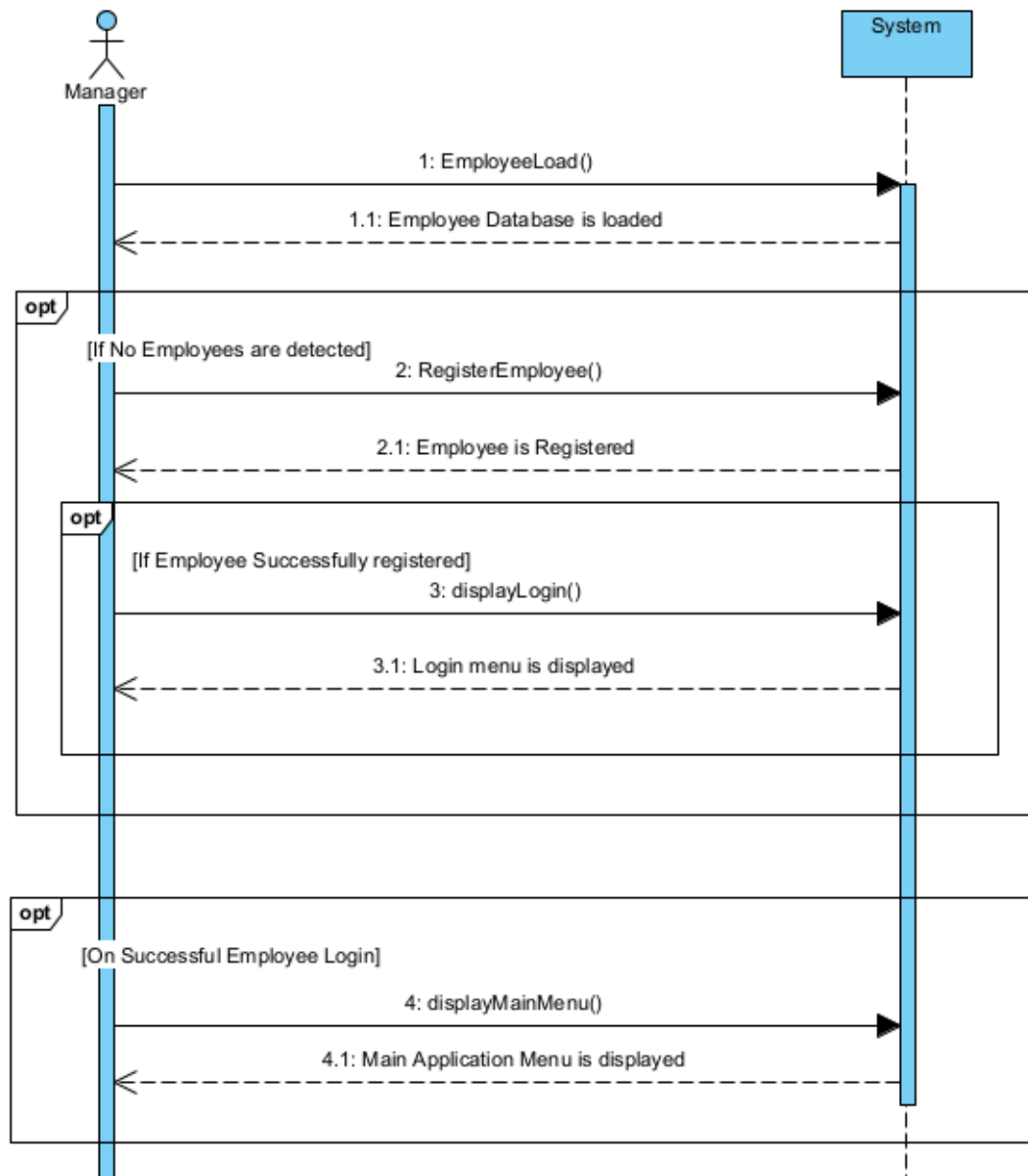
<b>ID:</b>	<b>UC 017</b>
------------	---------------

<b>Title:</b>	An employee/manager signs in but gives incorrect information
<b>Description:</b>	An employee/manager signs in but gives incorrect information
<b>Primary Actor:</b>	employee/manager
<b>Preconditions:</b>	The employee/manager has registered in the system and the system has the information of the employee/manager in the database.
<b>Post-conditions:</b>	The employee/manager is asked to reenter their information
<b>Main Flow:</b>	<ol style="list-style-type: none"> <li>1. The employee/manager tries sign in to the system with its username and password.</li> <li>2. The employee/manager puts in either an incorrect username or password</li> <li>3. The employee/manager receives an error message from the system and is asked to reenter their information</li> </ol>
<b>Frequency of Use:</b>	Only necessary when incorrect information is entered so incorrect information will vary
<b>Status:</b>	In Progress
<b>Owner:</b>	Zane Pitzer
<b>Priority:</b>	MEDIUM

<b>ID:</b>	UC 016
<b>Title:</b>	When the manager wants to adopt a pet of their own
<b>Description:</b>	Manager likes a pet and wants to adopt them
<b>Primary Actor:</b>	manager
<b>Preconditions:</b>	The pets are listed on the app and the information of the pets are stored in the database. The manager likes the pet and wants to adopt the pet.
<b>Post-conditions:</b>	The pet which has been adopted will not be listed in the app anymore, and will be removed from the databases
<b>Main Flow:</b>	<ol style="list-style-type: none"> <li>1. The manager chooses the pet they want to adopt</li> <li>2. The manager checks the information of the pet <ol style="list-style-type: none"> <li>2.1. If they decide to not adopt this pet go back</li> </ol> </li> <li>3. Go to checkout</li> <li>4. Pay the adoption fee <ol style="list-style-type: none"> <li>4.1. Choose payment method</li> <li>4.2. If the transaction did not go through <ol style="list-style-type: none"> <li>4.2.1. choose another payment method</li> </ol> </li> </ol> </li> <li>5. The pet is successfully adopted</li> <li>6. System will give the customer a number to let the user bring the pet home</li> </ol>
<b>Frequency of Use:</b>	1. Will be used frequently
<b>Status:</b>	In Progress
<b>Owner:</b>	Zane Pitzer
<b>Priority:</b>	HIGH

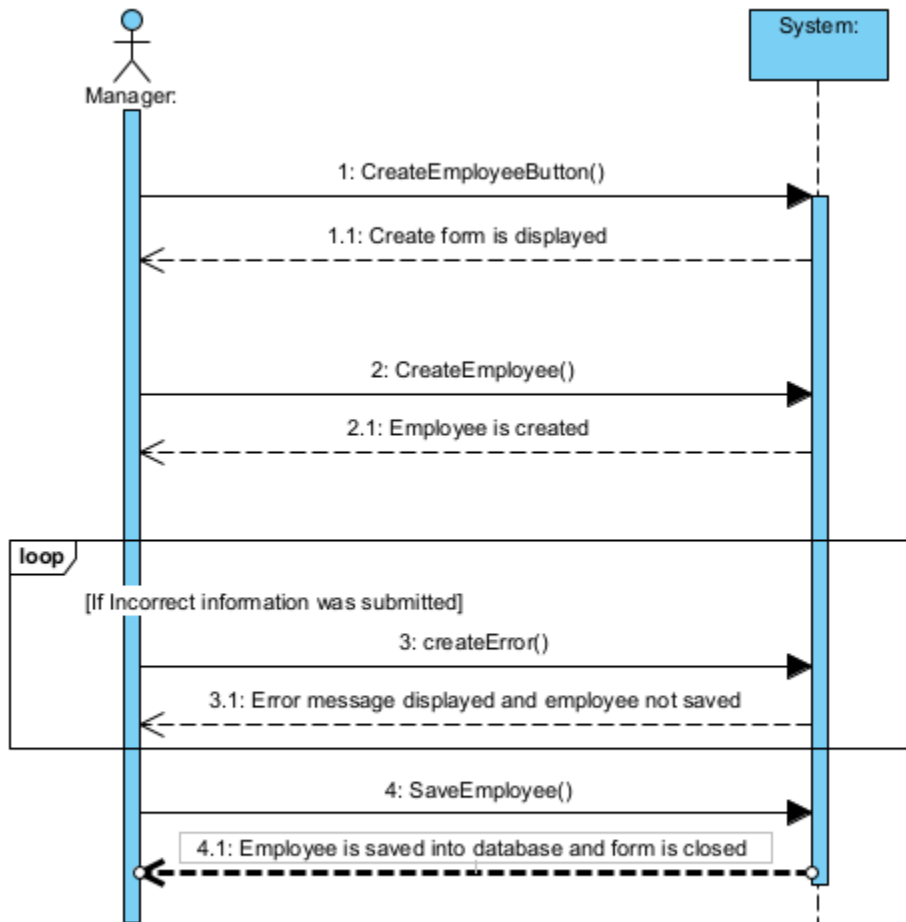
## SSD 1-3 Dante Hart

### SSD UC 001

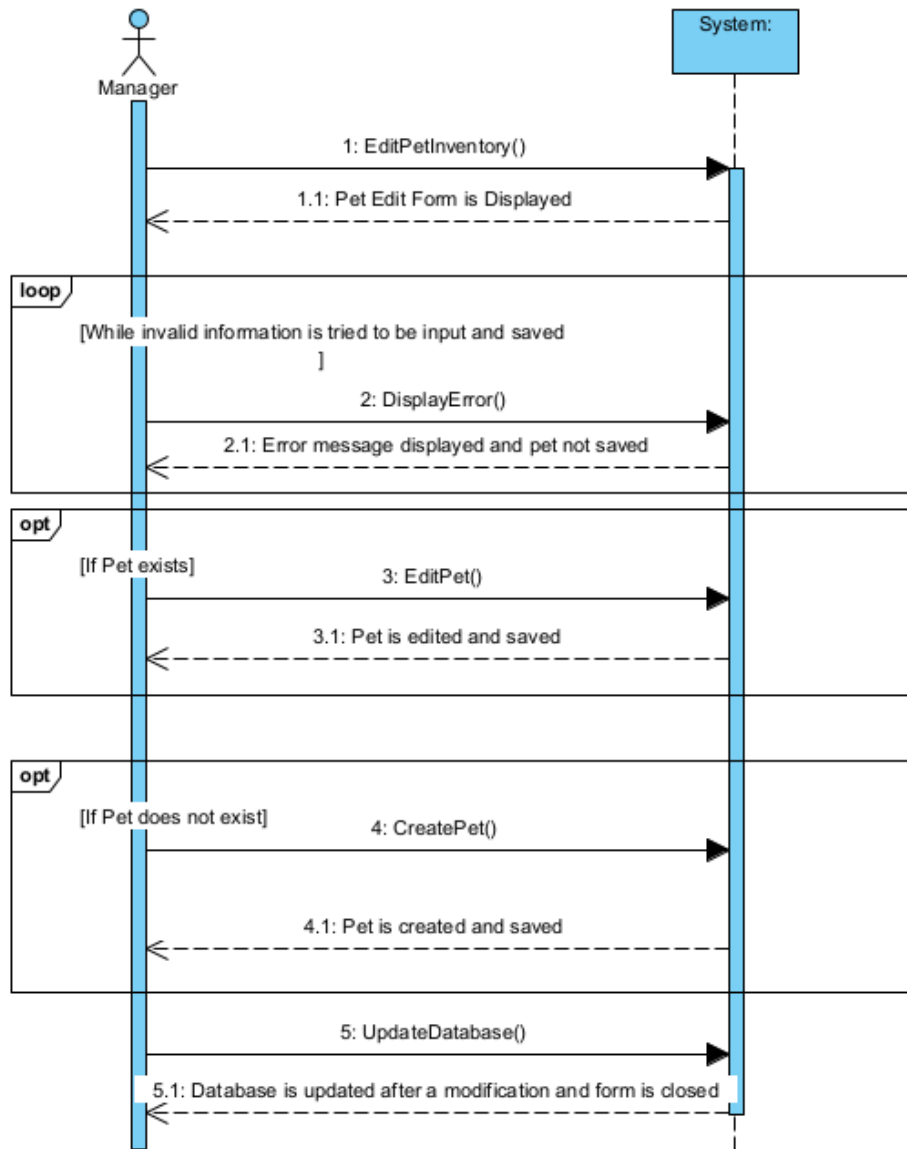




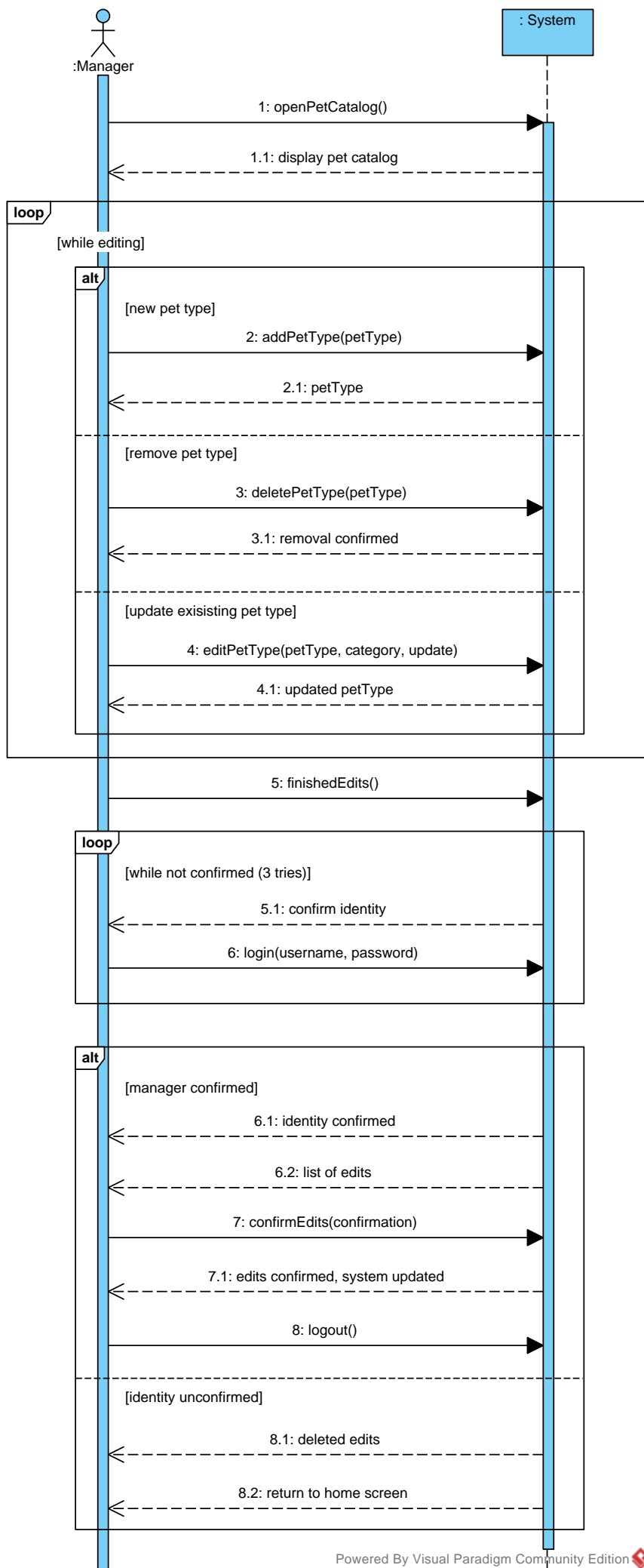
SSD UC 002

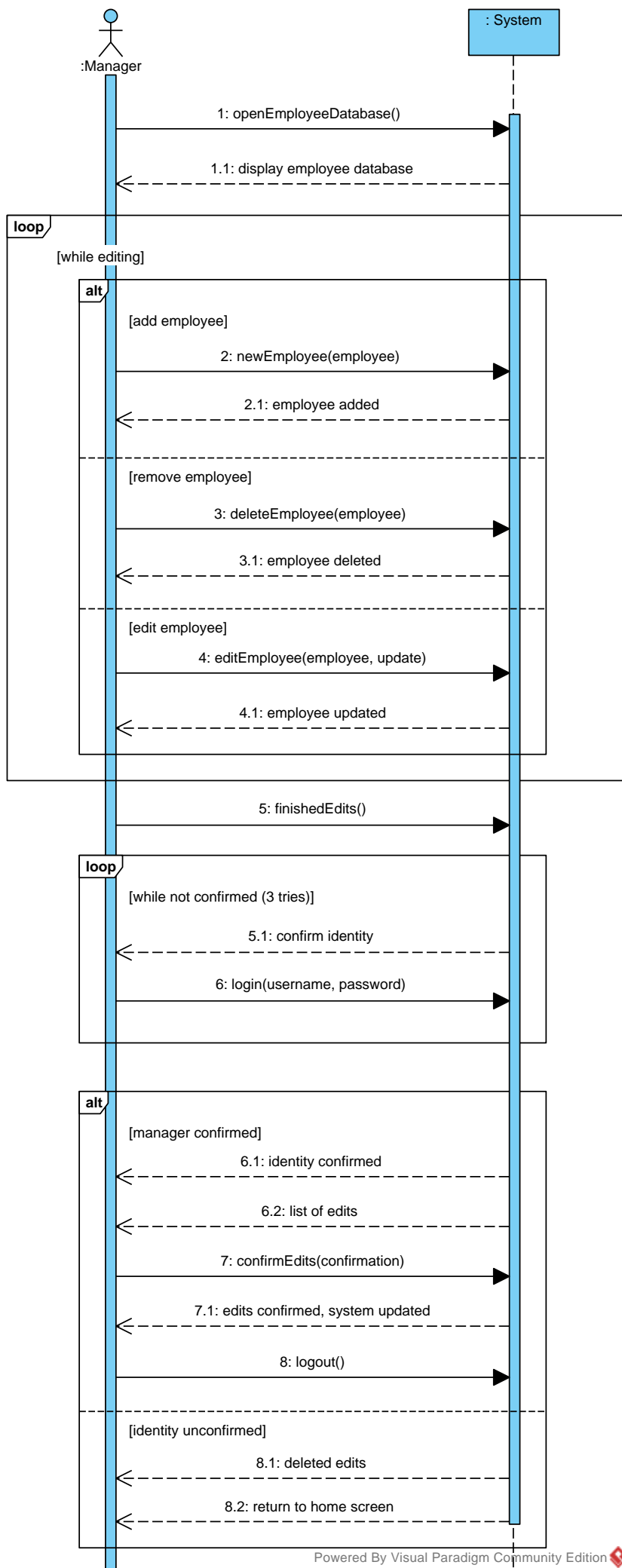


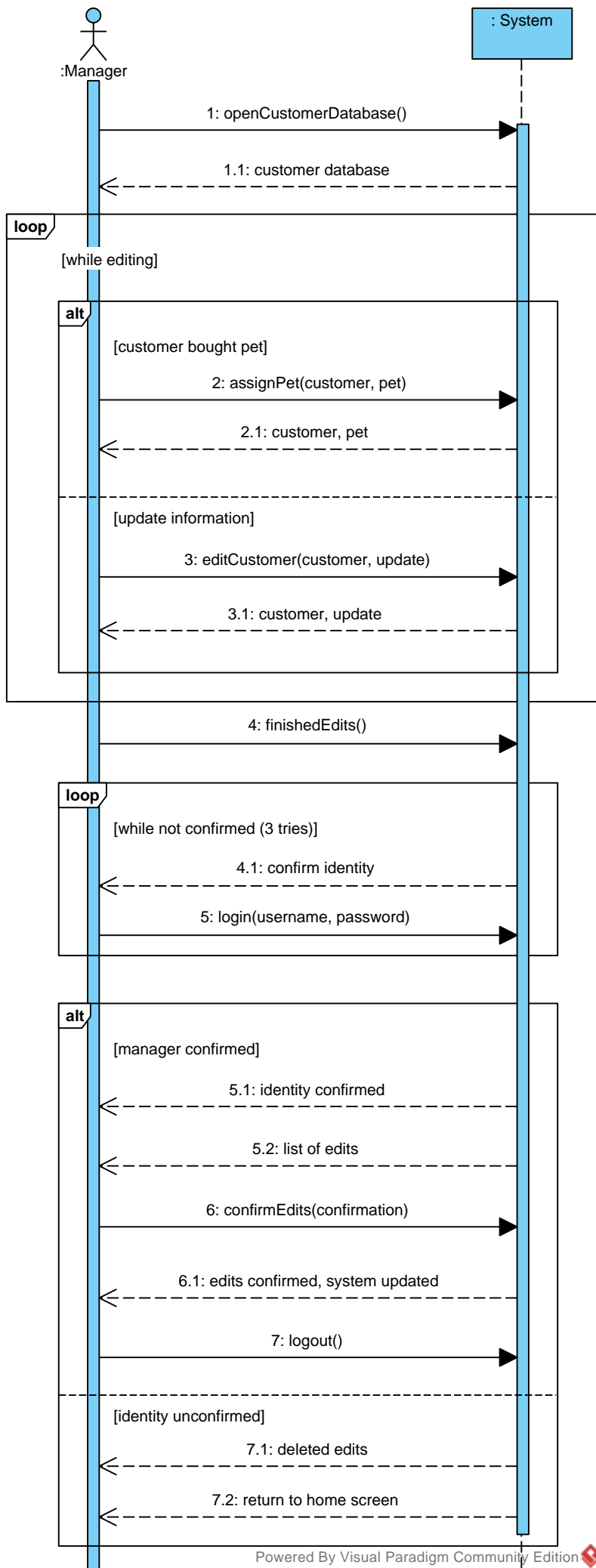
# SSD UC 003



## SSD 4-6 Garret Parker



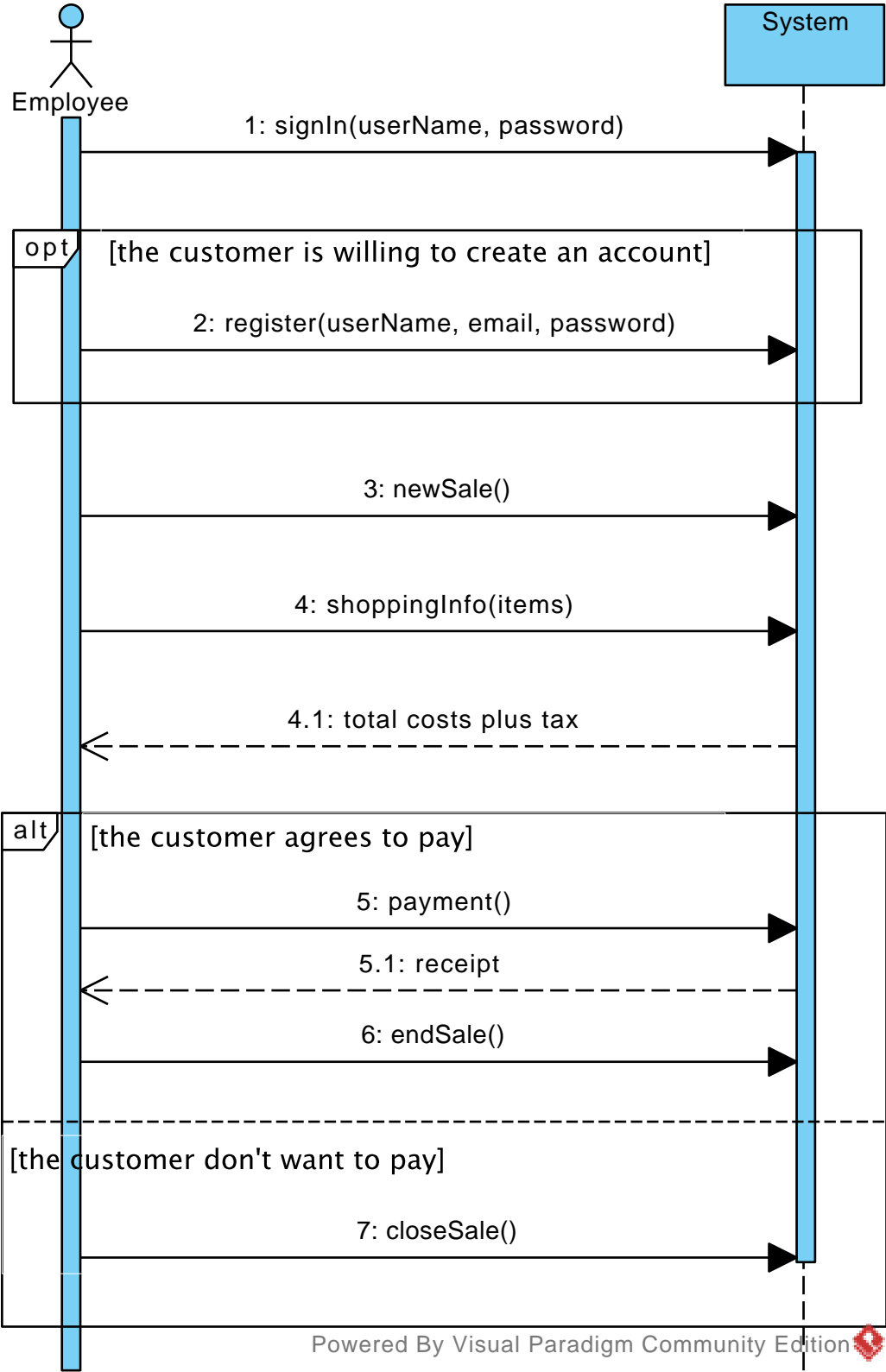


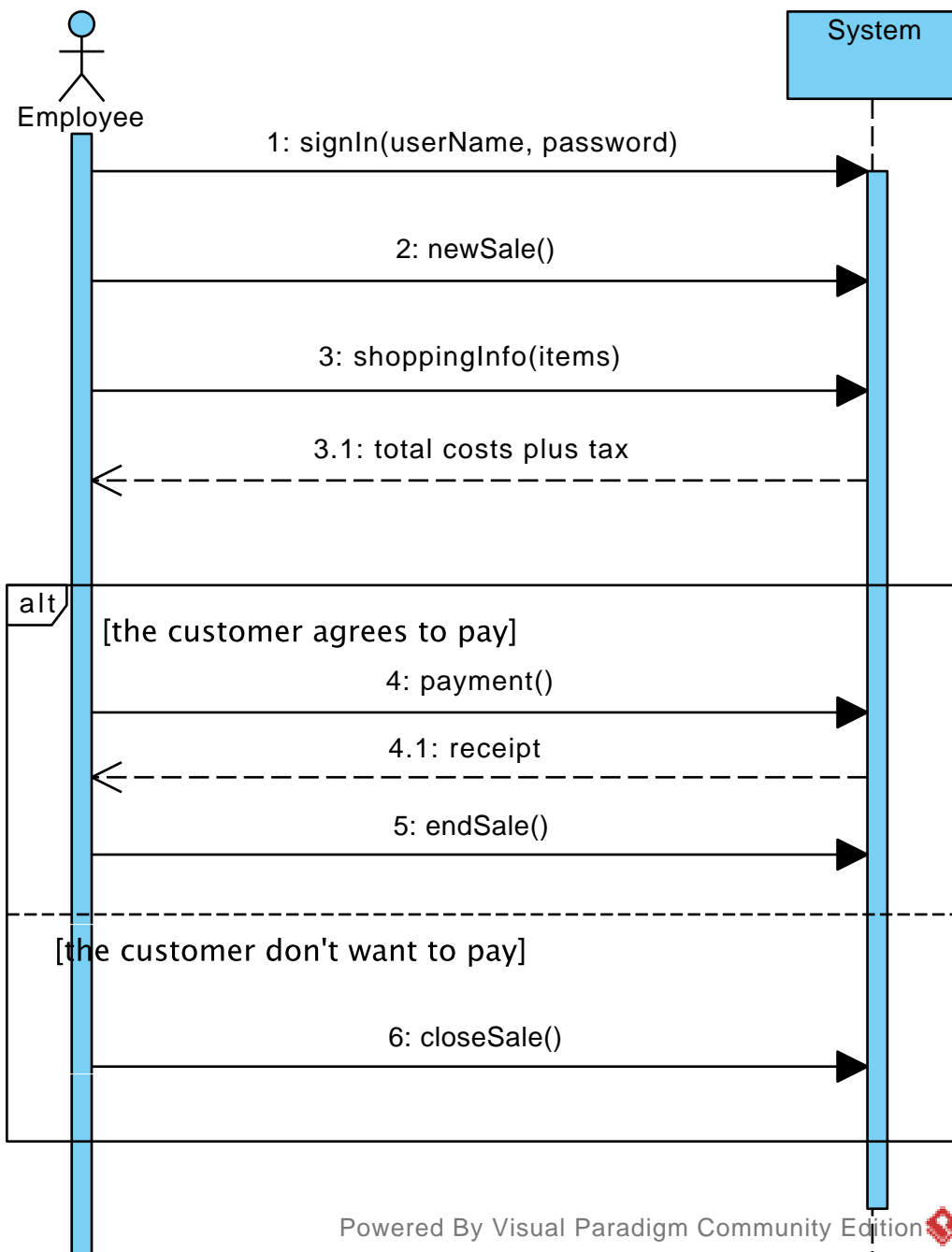


## System

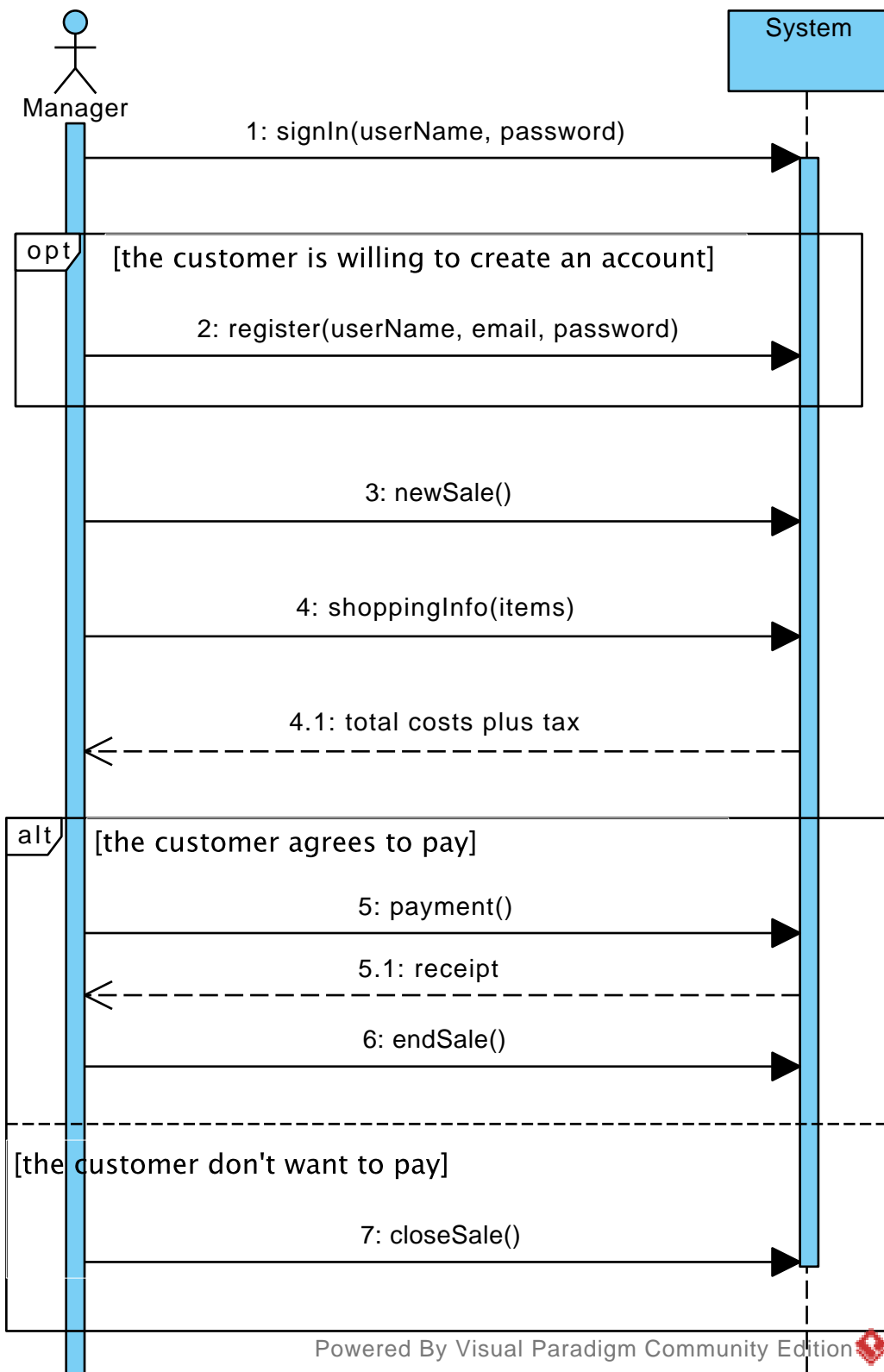
- +openPetCatalog()
- +addPetType(petType)
- +deletePetType(petType)
- +editPetType(petType, category, update)
- +finishedEdits()
- +login(username, password)
- +confirmEdits(confirmation)
- +logout()
- +openEmployeeDatabase()
- +openCustomerDatabase()
- +newEmployee(employee)
- +deleteEmployee(employee)
- +editEmployee(employee, update)
- +assignPet(customer, pet)
- +editCustomer(customer, update)

Powered By Visual Paradigm Community Edition 





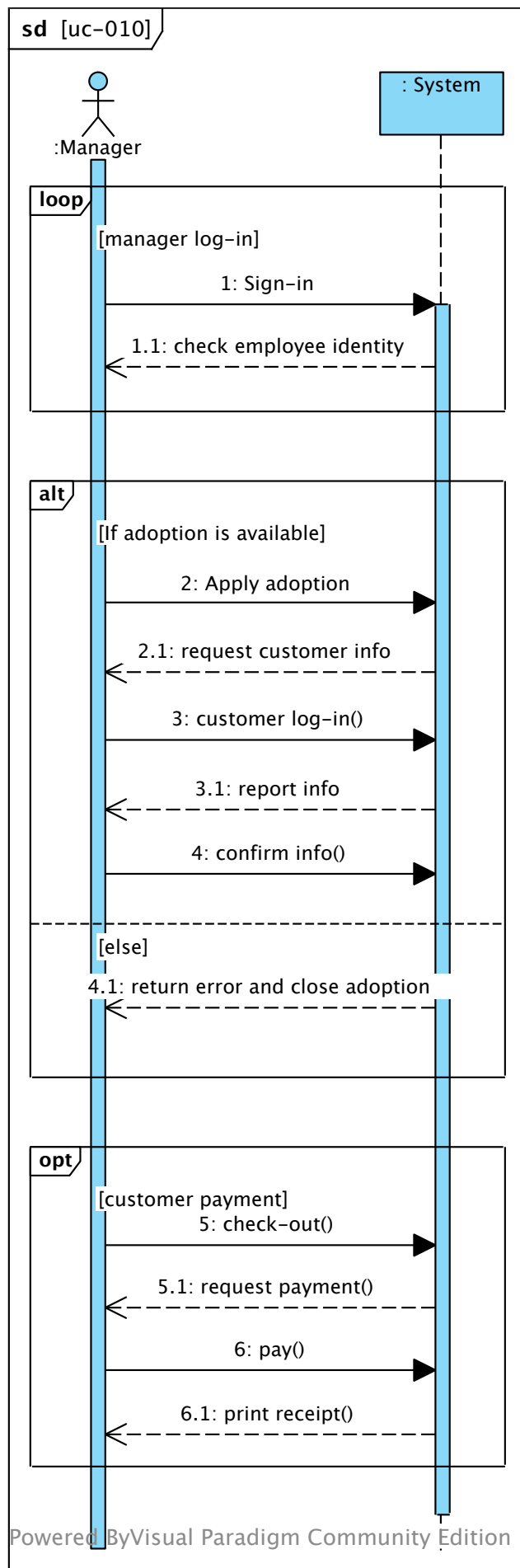




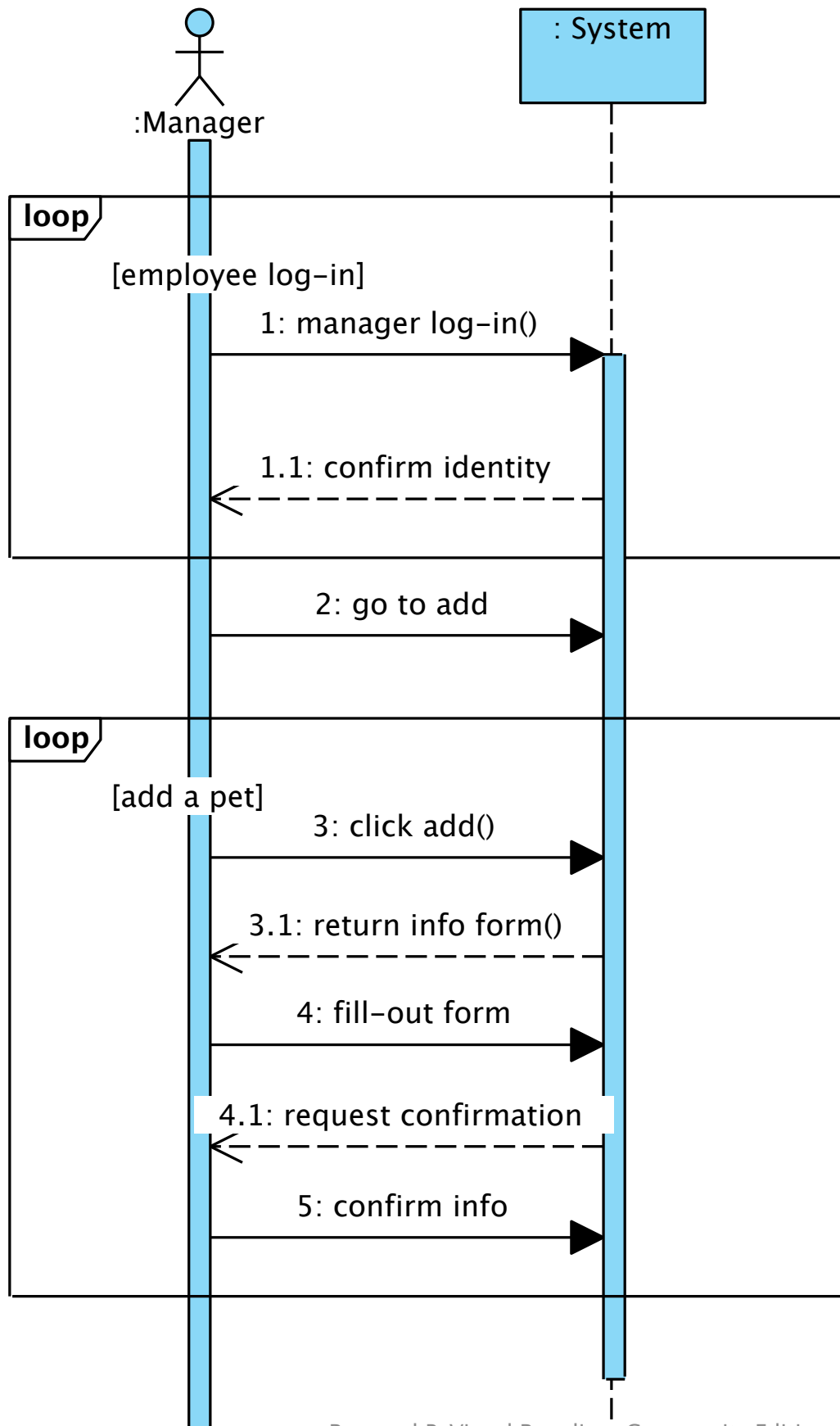
## System

```
+signIn(userName, password)
+register(userName, email, password)
+newSale()
+shoppingInfo(items)
+payment()
+endSale()
+closeSale()
```

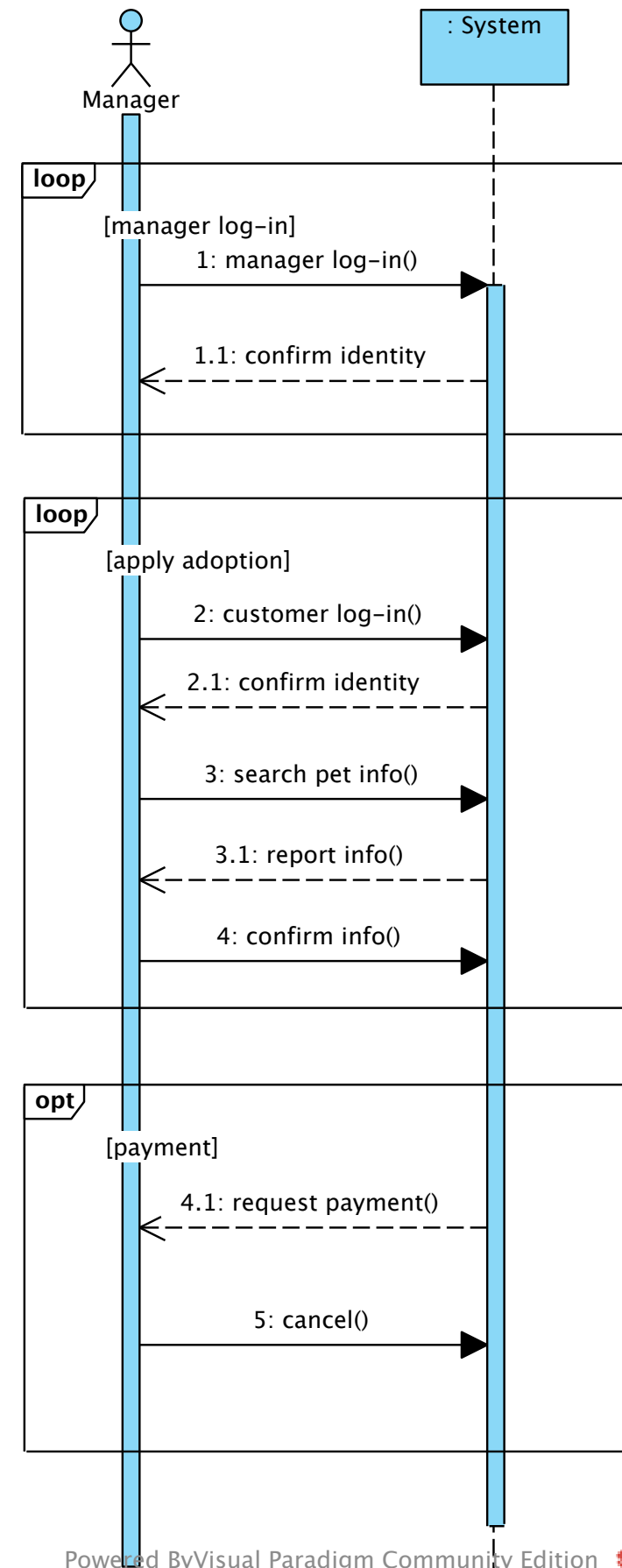
Powered By Visual Paradigm Community Edition 



sd [UC-011]



sd [UC-012]

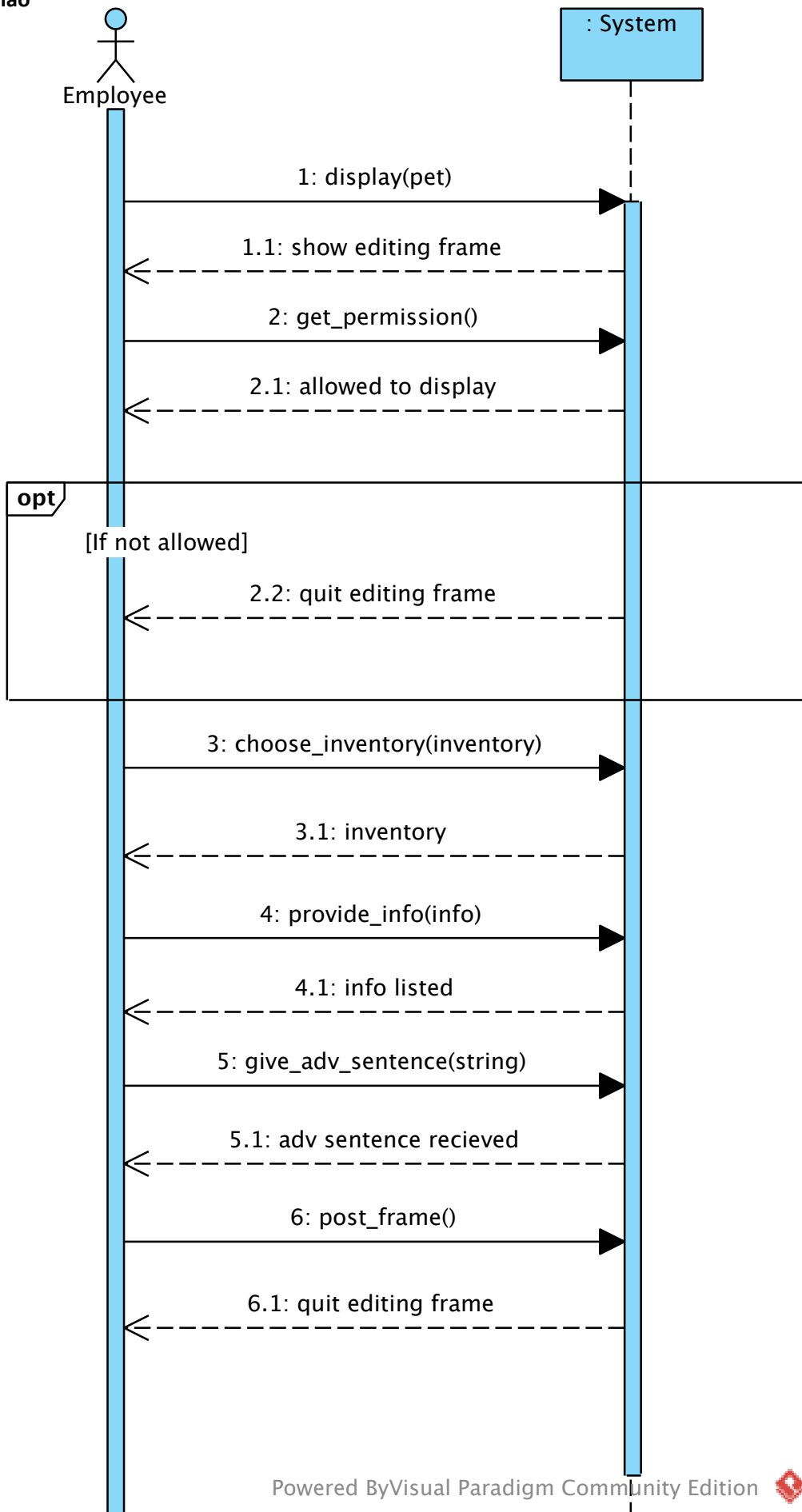


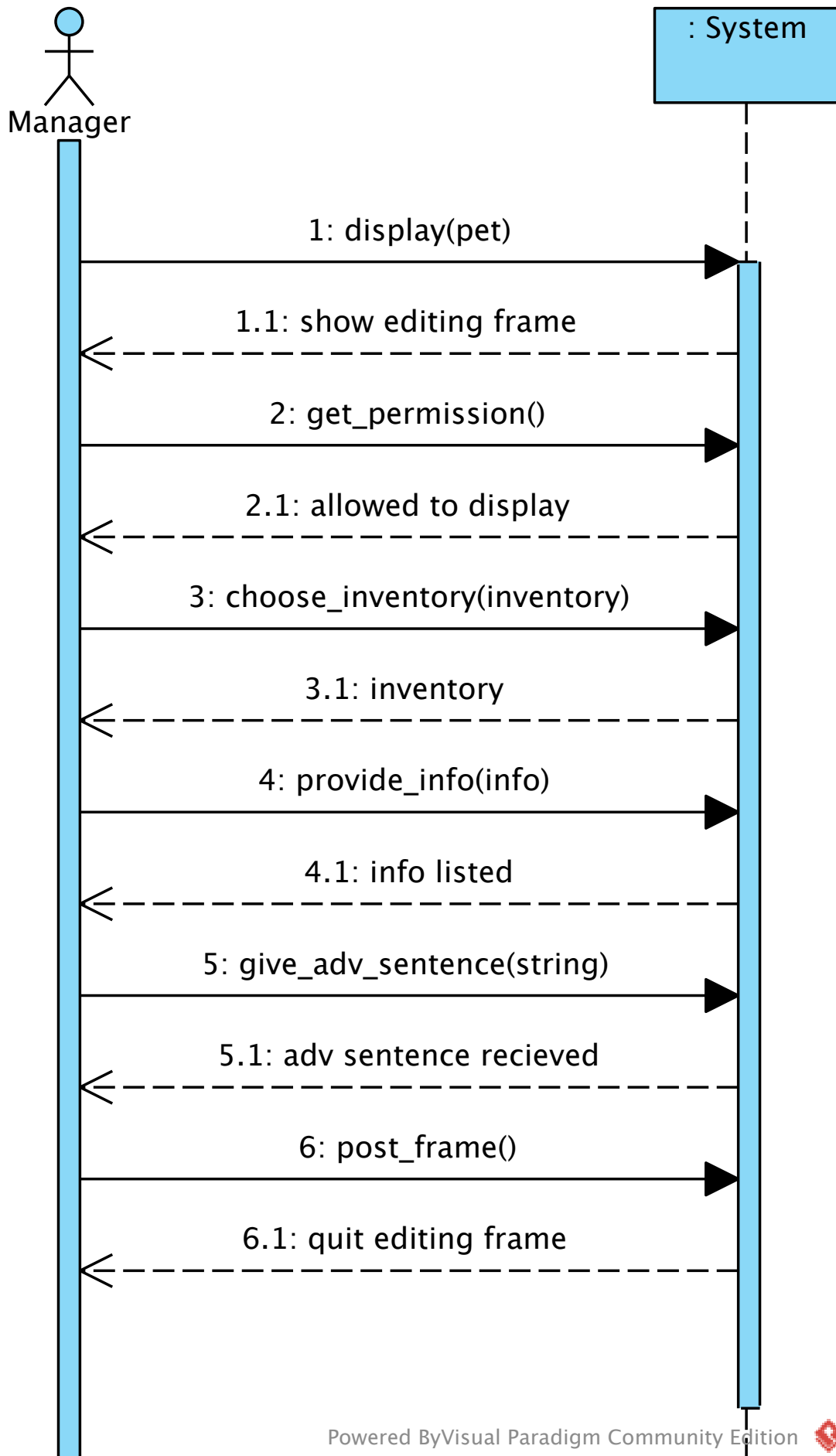
## System Operations of UC10-12

### System

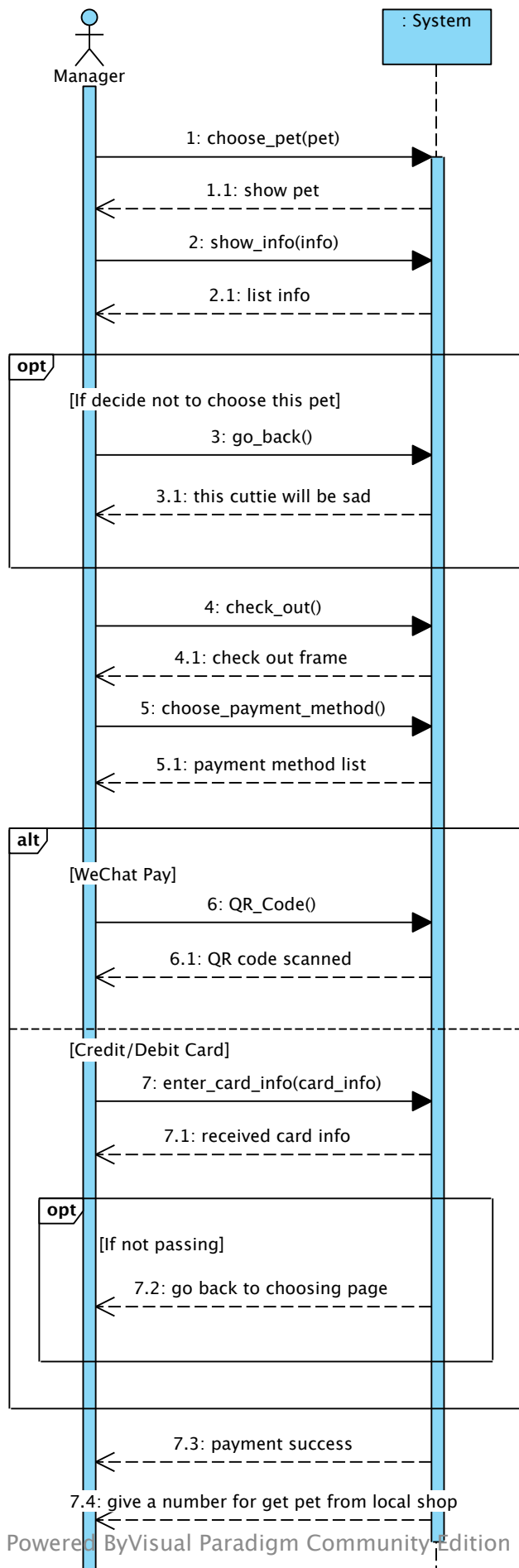
- +manager log in()
- +go to add()
- +click add()
- +fill out form()
- +confirm info()
- +customer log in()
- +search pet info()
- +cancel()
- +sign in()
- +apply adoption()
- +check out()
- +pay()



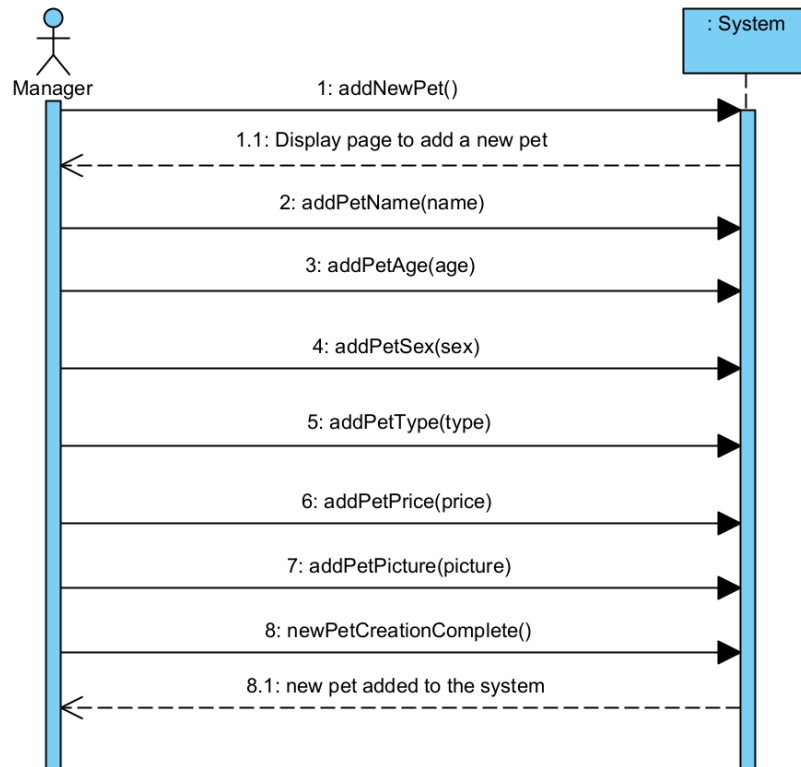


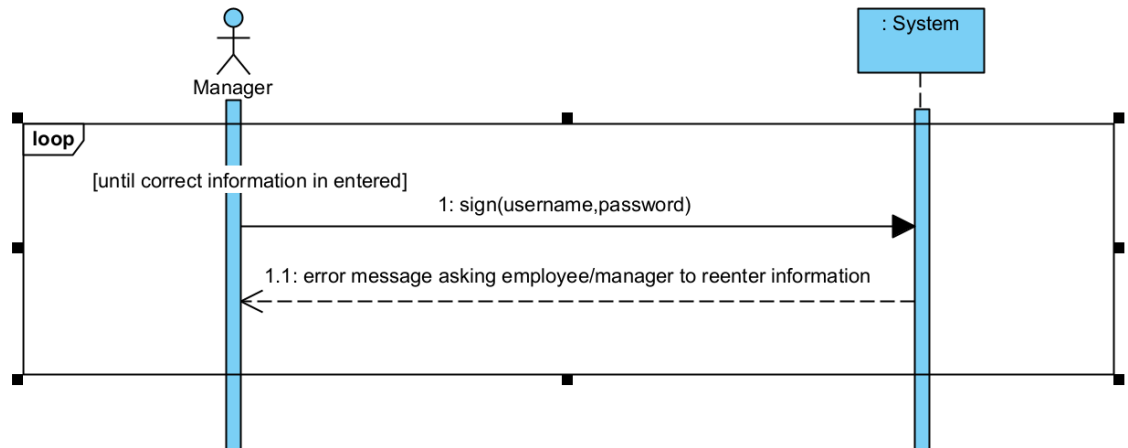


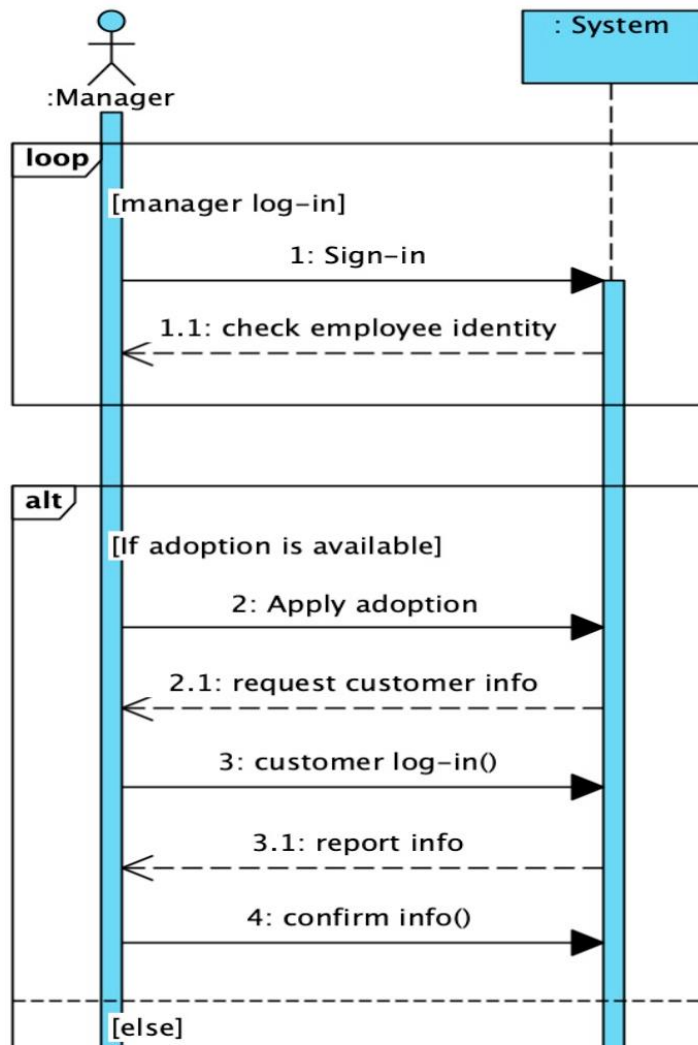


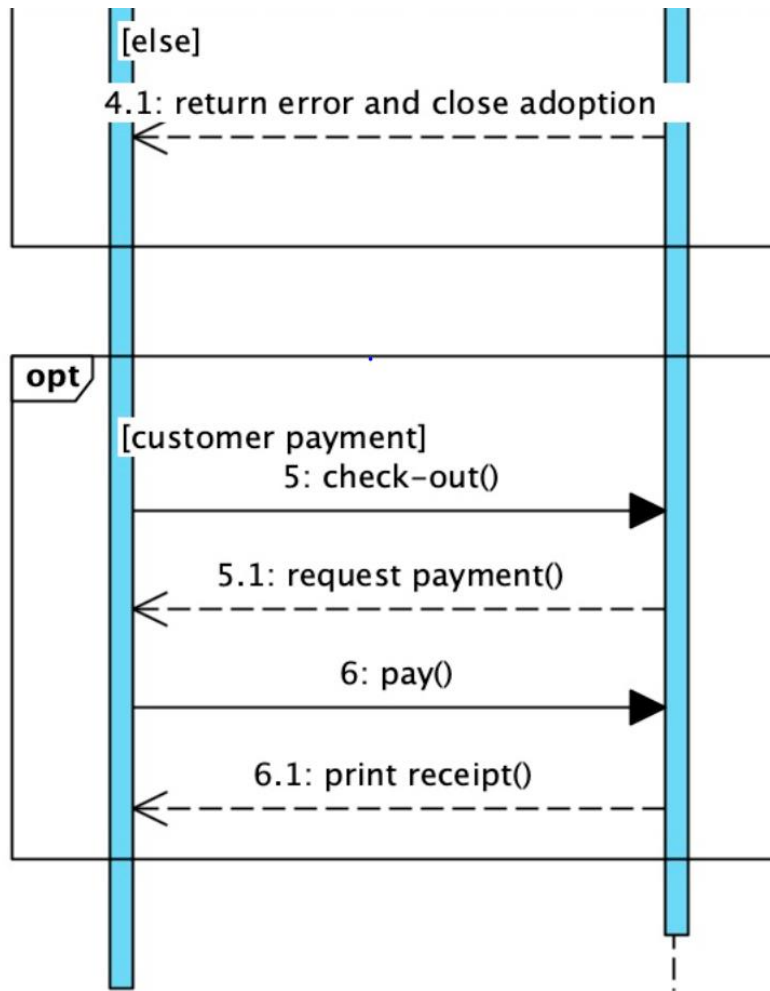


## SSD 16, 17, 20 Zane Pitzer









## **Issue tracking**

[https://github.com/KillerRaptor247/STAY\\_Negative.git](https://github.com/KillerRaptor247/STAY_Negative.git)