

Pet Adoption Clinic

Project vision

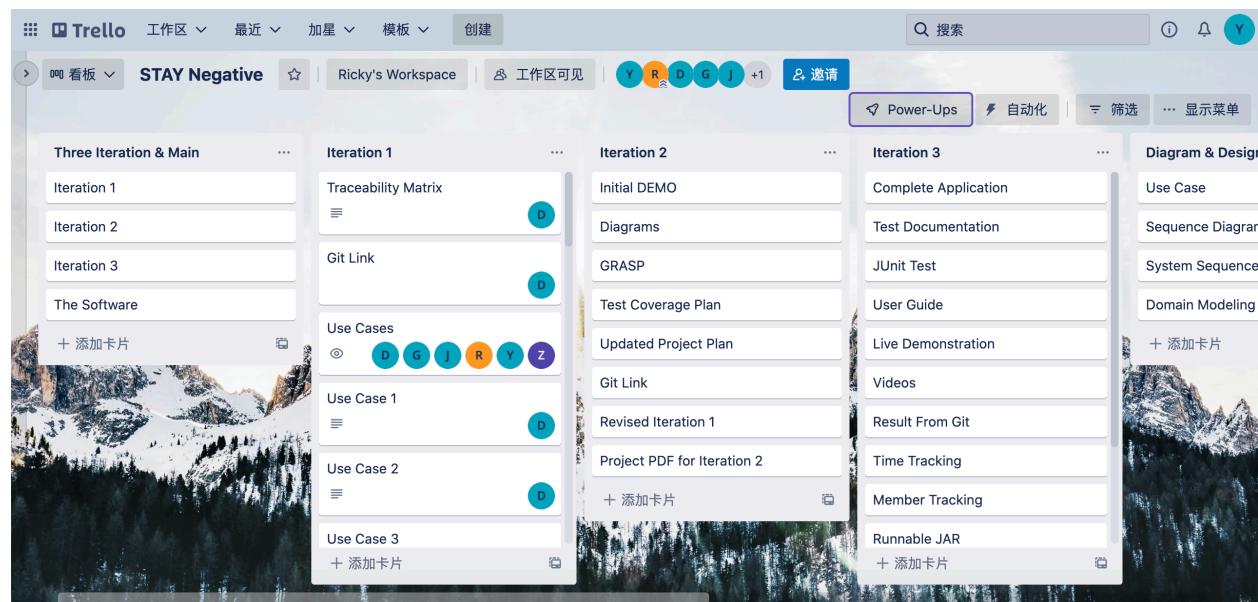
- The application will be a pet adoption clinic application that will function as software for day-to-day activities for a Pet adoption center. This software will function as a database for holding the current inventory of pets as well as different kinds of pet the center can house via a catalog. Additionally, the software will allow employees to checkout pets from the inventory to a customer. The database for the inventory and catalog of pets will be stored by reading and writing to a CSV file. There will also be a database of Employees and Customers to use their information to sign into the application. Among the Employees, there will be a manager that will be able to edit and add existing employees as well as manage the pet catalog and inventory as deemed necessary by the manager. There will be multiple forms corresponding with their purpose.(i.e Sign-in, main menu, edit pet, checkout....etc). These forms will be accessible from a main menu form that will appear after a successful employee sign in. Some forms will be inaccessible if an Employee is not a Manager.

Team members

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

Trello

<https://trello.com/b/v8jQjzrE/stay-negative>



Our Website

https://killerraptor247.github.io/STAY_Negative/

Our GitHub Link

https://github.com/KillerRaptor247/STAY_Negative

Our Issue Tracking

[https://github.com/KillerRaptor247/STAY_Negative/
issues](https://github.com/KillerRaptor247/STAY_Negative/issues)

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

Traceability Matrix

	Requirements																										
	REQ-1 Pet Class Implementation	REQ-2 PetCatalog Class Implementation	REQ-3 Dog Option Implementation	REQ-4 Cat Option Implementation	REQ-5 Customer Class Implementation	REQ-6 Employee Class Implementation	REQ-7 Database Class Implementation	REQ-8 Customer Database Class Implementation	REQ-9 Employee Database Class Implementation	REQ-10 Employee Checkout	REQ-11 Manager Identification	REQ-12 Login Screen Implementation	REQ-13 No Employee Detection	REQ-14 Employee Creation	REQ-15 Display Screen Implementation	REQ-16 Inventory Display from Manager	REQ-17 CSV File Read-in	REQ-18 CSV File Write-out	REQ-19 Manager Database Auto-Update	REQ-20 Database Auto-Update	REQ-21 Checkout Screen Menu	REQ-22 Customer Checkout Registration	REQ-23 Customer Login	REQ-24 Unique Pet ID			
UC-1 First Boot-up of System						X		X		X	X	X	X	X	X												
UC-2 Manager Creates an Employee						X	X	X		X	X	X	X	X	X				X	X							
UC-3 Manager edits a Pet in the inventory	X		X	X		X	X			X				X					X	X				X			
UC-4 Manager edits a Pet in the catalog	X	X	X	X		X	X			X			X				X			X	X						
UC-5 Manager edits Employee in Database						X	X	X		X	X				X				X	X							
UC-6 Manager edits Customer Database						X	X	X				X				X				X	X						
UC-7 Employee Sign-in nonregistered Customer	X	X	X	X	X	X	X	X	X	X	X	X		X					X	X	X	X	X				
UC-8 Employee Sign-in registered Customer	X	X	X	X	X	X	X	X	X	X	X	X	X		X				X	X	X	X	X				
UC-9 Manager Sign-in nonregistered Customer	X	X	X	X	X	X	X	X	X	X	X	X	X		X				X	X	X	X	X				
UC-10 Manager Sign-in registered Customer	X	X	X	X	X	X	X	X	X	X	X	X	X		X				X	X	X	X	X				
UC-11 Pet added to catalog by Manager	X	X	X	X		X	X			X				X				X			X						
UC-12 Customer cancels Pet transaction	X		X	X	X	X	X	X	X	X					X					X	X						
UC-13 Employee displays Inventory	X	X	X	X		X	X							X	X												
UC-14 Manager displays Inventory	X	X	X	X		X	X			X				X	X												
UC-15 Employee Adopts A Pet	X		X	X		X			X	X				X					X	X	X	X	X				
UC-16 Manager Adopts A Pet	X		X	X		X			X	X	X			X					X	X	X	X	X				
UC-17 Incorrect Employee Sign-in						X			X		X		X		X					X	X	X					
UC-18 Incorrect Customer Sign-in						X	X		X		X		X		X			X			X	X	X				
UC-19 Application is closed						X									X			X									
UC-20 A Pet is Checked in by Manager	X	X	X	X		X			X		X		X		X			X		X	X				X		

Wireframes

please adopt a pet				
Name	Age	Birthday	Color	Weight
<input type="button" value="cancel"/> <input type="button" value="adppr"/>				

Wireframes

Edit Form

Name:

Age:

Birthday:

Color:

Weight:

Gender: Male Female

Species: Cat Dog

Untitled Gantt Project

2022329

<http://>

Project manager

Dante Hart

Project dates

202228 - 2022416

Completion

88%

Tasks

15

Resources

6

Tasks

Name	Begin date	End date
Iteration 1		
Group meeting 1	2022/2/8	2022/2/8
Project visiton	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 Diagram	2022/2/22	2022/2/23
Presentation	2022/2/24	2022/2/24
Iteration 2		
Group meeting2	2022/3/1	2022/4/1
Group meeting3	2022/3/1	2022/3/1
	2022/4/1	2022/4/1
Iteration 3		
Group meeting4	2022/4/15	2022/4/15
	2022/4/15	2022/4/15

Resources

3

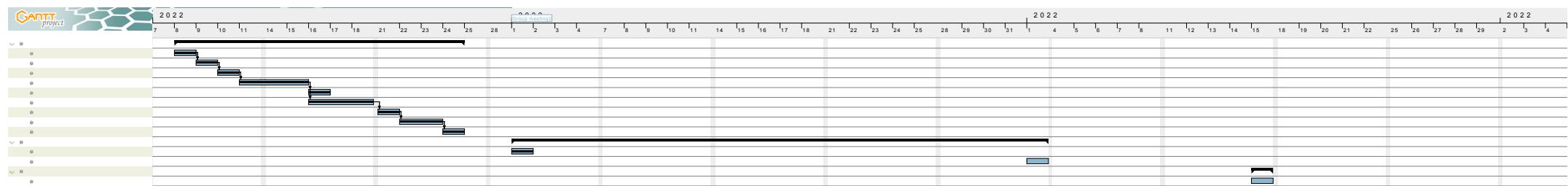
Name	Default role
Dante Hart	project manager
Garrett Parker	developer
Yihan Zhang	graphic designer
Jingke Shi	doc writer
Ruiqi Zhao	tester
Zane Pitzer	analysis

Untitled Gantt Project

2022329

4

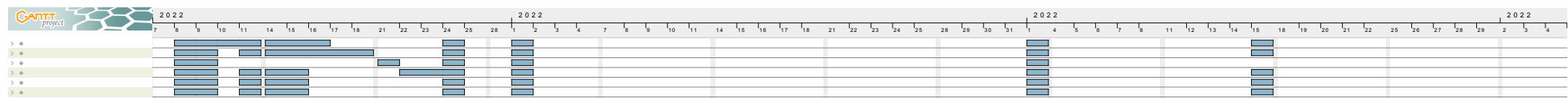
Gantt Chart

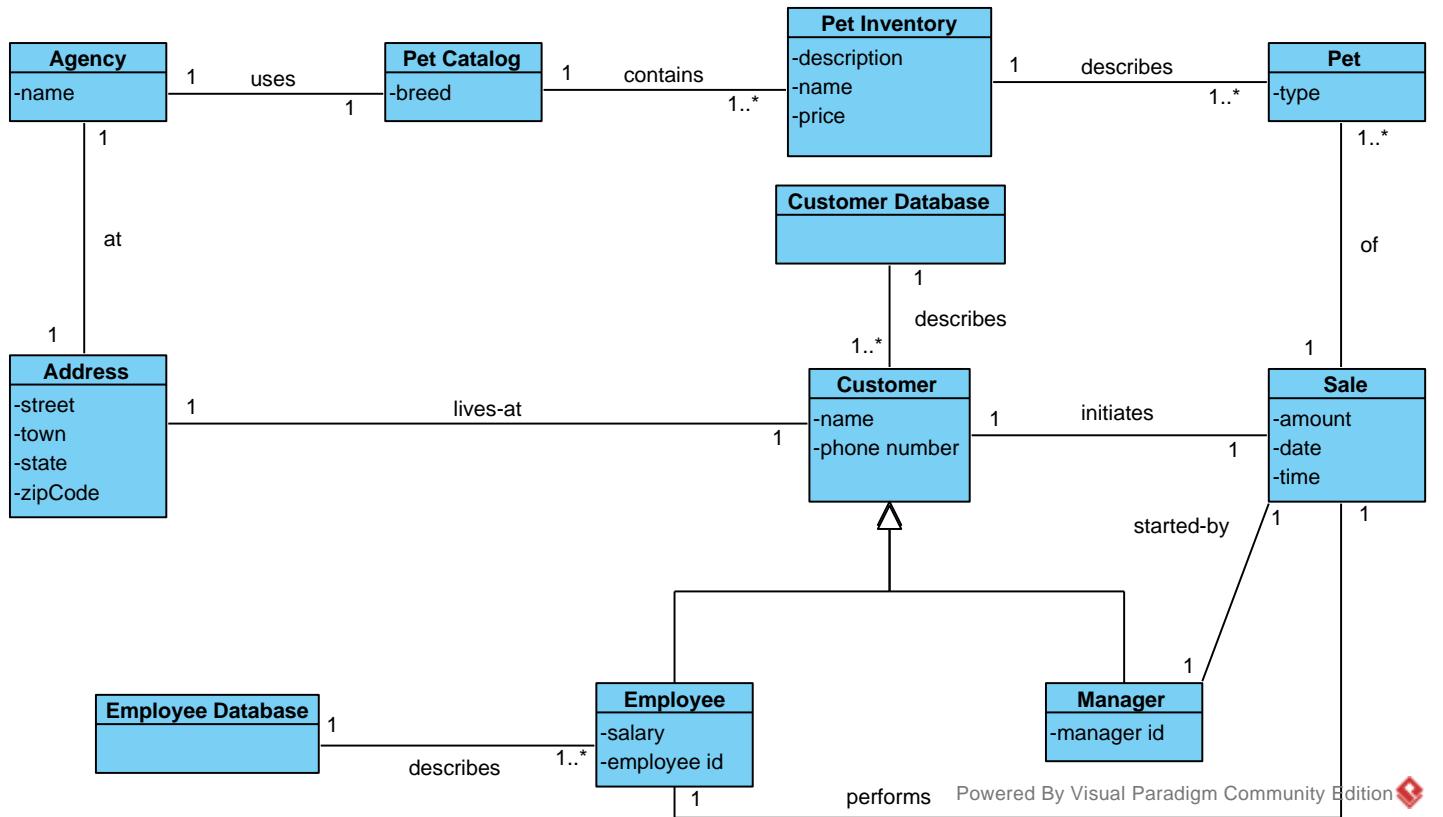


Untitled Gantt Project

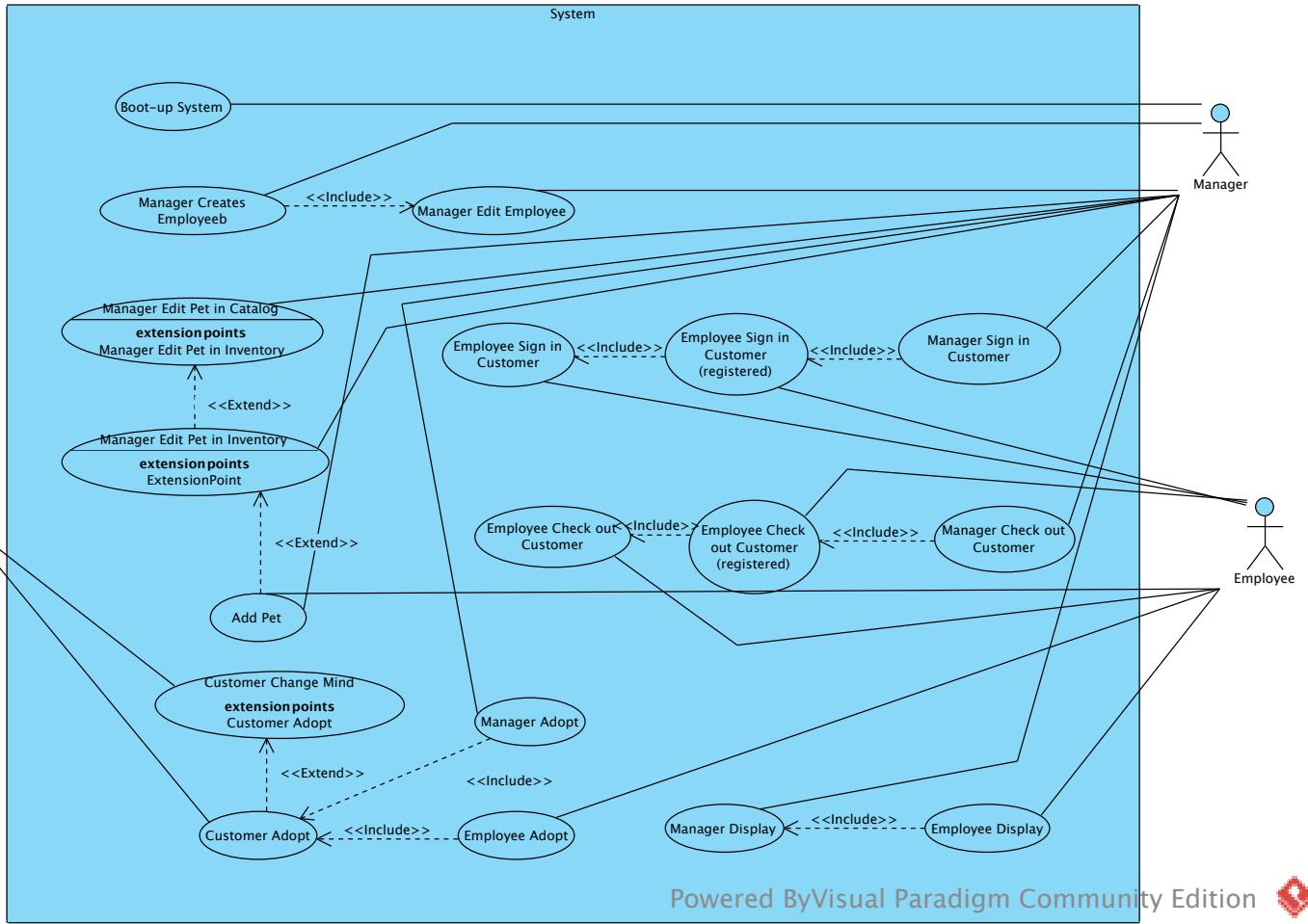
2022329

Resources Chart





uc [Use Case Diagram1]



Powered By Visual Paradigm Community Edition

Actors

1. Manager: The person that runs the pet adoption agency. The manager oversees updating the pet inventory, customer database, and employee database whenever changes are needed. The Manager can perform sales, as well as adopt their own pet.
2. Employee: The Employee is in charge of performing sales. The Employee is also able to adopt their own pet, but they must perform this sale with a manager, they cannot do it themselves. Employees must interact with a customer when they are adopting a pet.
3. Customer: The customer interacts with the system minimally. They are only involved when adopting a pet, and if they cancel the adoption midway through the process. When they register, their information is stored in the customer database.

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

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:	<ol style="list-style-type: none"> 1. A user starts the application 2. The System checks the employee database and detects there are no employees 3. Instead of prompting a sign-in, The system prompts for a Manager registration. 4. The Manager creates an Employee with the title of Manager in the Employee database. 5. The register form closes and the login form appears for the Manager to login
Extensions:	<p>2.a If there are employees found, the login form appears instead</p> <p>3.a If the Manager closes the form without a complete registration, and error message will be displayed and the application will close.</p>
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
Primary Actor:	Manager
Preconditions:	A Manager has signed into the application and is on their main menu screen
Postconditions:	An Employee has been successfully created
Main Success Scenario:	<ol style="list-style-type: none"> 1. A manager clicks on a button that takes them to the create employee form 2. An employee puts in information for an employee into the system and presses a button to commit the changes 3. Upon valid information, an Employee is added into the database 4. The form closes back to the main menu and the Employee Database is updated
Extensions:	<p>2.a If Incorrect information is given, an error message is displayed and an Employee's information is not saved.</p>
Frequency of Use:	Estimation of use is moderate use.
Status:	In Progress
Owner:	Dante Hart
Priority:	MEDIUM

Use Case Template:

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:	<ol style="list-style-type: none">1. A manager clicks the Edit Inventory button on the main menu screen2. A Form appears that displays the Pet inventory to be edited3. The manager edits a field for a Pet and clicks a save button4. The pet is saved and the Edit form is closed5. The pet database is updated.
Extensions:	3.a) If invalid information is input an error message is displayed
Frequency of Use:	Estimation of use is moderate use.
Status:	In Progress
Owner:	Dante Hart
Priority:	MEDIUM

	ID: UC 005
	Title: Edit Employee Database
	Description: A Manager edits an employee in the database.
	Primary Actor: Manager
	Preconditions: A valid Manager is logged into the system.
	Post-conditions: The employee database is updated successfully.
Success Scenario:	<p>Main 1. The Manager navigates to the employee database.</p> <p>2. The System displays the employee database.</p> <p>3. The Manager edits the database (add, remove, or edit employee.)</p> <p>4. The System displays the edit.</p> <p>5. Repeat steps 3 and 4 until the edits have been finished.</p> <p>6. The Manager notifies the System they have finished making edits.</p> <p>7. The System asks the Manager to confirm their identity.</p> <p>8. The Manager enters their username and password.</p> <p>9. The System confirms the person editing the database is a Manager.</p> <p>10. The System displays a list of the edits made.</p> <p>11. The Manager confirms the edits.</p> <p>12. The System confirms the edits have been made.</p> <p>13. The Manager logs out.</p>
Extensions:	<p>1. If the Manager's identity is not confirmed after the edits are made, the System discards the edits.</p> <p>2. If the Manager does not confirm the edits, they can return to the database and continue making edits.</p>
Frequency of Use:	When a new employee is hired, an employee is fired, or an employee needs their information updated.
Status:	In Progress
Owner:	Garrett Parker
Priority:	MEDIUM

ID:	UC 006
Title:	Edit Customer Database
Description:	A Manager edits a customer in the database.
Primary Actor:	Manager
Preconditions:	A valid Manager is logged into the system.
Post-conditions:	The customer base has been successfully updated.
Main Success Scenario:	<p>1. The Manager navigates to the customer database.</p> <p>2. The System displays the customer database.</p> <p>3. The Manager edits the database (assign pets, update information.)</p> <p>4. The System displays the edit.</p> <p>5. Repeat steps 3 and 4 until all the edits have been completed.</p> <p>6. The Manager notifies the System they have finished making edits.</p> <p>7. The System asks the Manager to confirm their identity.</p> <p>8. The Manager enters their username and password.</p> <p>9. The System confirms the person editing the database is a Manager.</p> <p>10. The System displays a list of the edits made.</p> <p>11. The Manager confirms the edits.</p> <p>12. The System confirms the edits have been made.</p> <p>13. The Manager logs out.</p>
Extensions:	<p>1. If the Manager's identity is not confirmed, the edits are ignored.</p> <p>2. If the list of edits is not correct, the Manager can go back and fix mistakes.</p>
Frequency of Use:	When a customer adopts an animal, the customer database needs to be updated. When a customer needs to change their information.
Status:	In Progress
Owner:	Garrett Parker
Priority:	MEDIUM

Use cases 7-9 Yihan Zhang

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

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

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

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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none">1. The manager chooses add view2. The system redirects to add pet form page3. The manager fills-out the info form4. The system requests confirmation5. The manager checks the info typed before and confirm<ol style="list-style-type: none">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	<ol style="list-style-type: none">1. The manager starts application for the adoption process2. The system searches adoption info and registered customer info3. The manager confirm the info4. The system requests payment5. 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 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

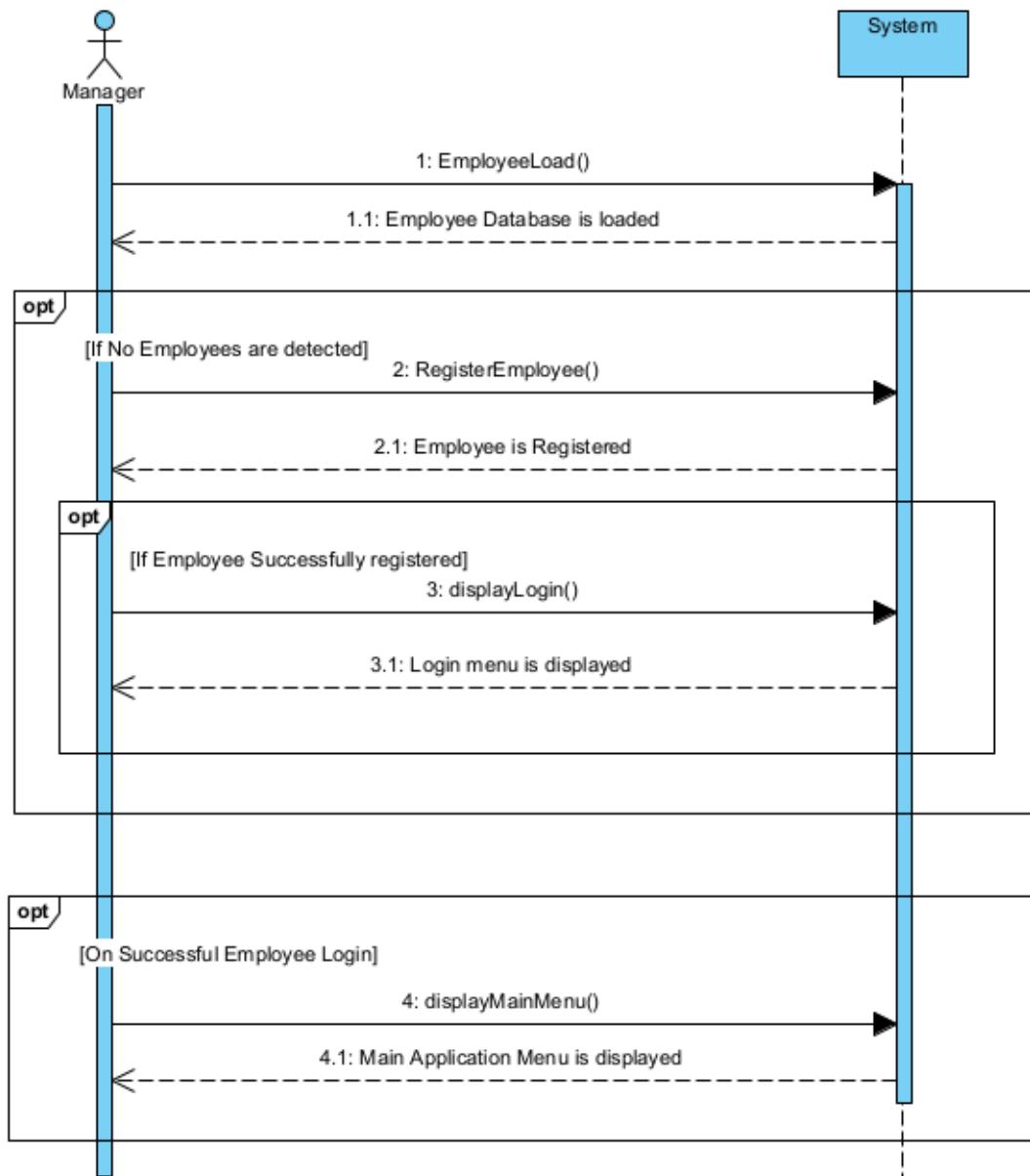
ID:	UC 016
Title:	When the manager wants to adopt a pet of their own
Description:	Manager likes a pet and wants to adopt them
Primary Actor:	manager
Preconditions:	The pets are listed on the app and the information of the pets are stored in the database. The manager likes the pet and wants to adopt the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

ID: UC 017	
Title:	An employee/manager signs in but gives incorrect information
Description:	An employee/manager signs in but gives incorrect information
Primary Actor:	employee/manager
Preconditions:	The employee/manager has registered in the system and the system has the information of the employee/manager in the database.
Post-conditions:	The employee/manager is asked to reenter their information
Main Flow:	<ol style="list-style-type: none"> 1. The employee/manager tries sign in to the system with its username and password. 2. The employee/manager puts in either an incorrect username or password 3. The employee/manager receives an error message from the system and is asked to reenter their information
Frequency of Use:	Only necessary when incorrect information is entered so incorrect information will vary
Status:	In Progress
Owner:	Zane Pitzer
Priority:	MEDIUM

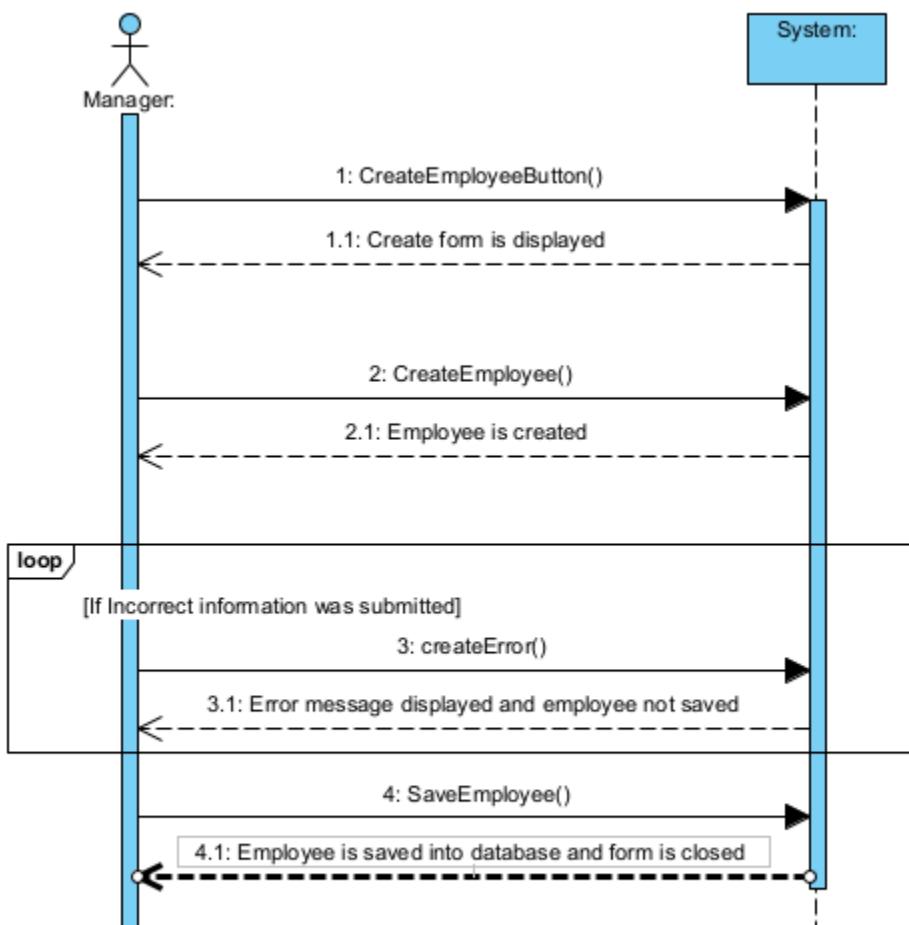
ID:	UC 019
Title:	Close Application
Description:	The application is shut down.
Primary Actor:	Manager
Preconditions:	A valid manager is logged into the system.
Post-conditions:	The system has been shut down and the pet inventory, employee database, and customer database have been exported to their respective files.
Main Success Scenario:	<p>1. The Manager attempts to close the application.</p> <p>2. The System asks the Employee to verify if they want to exit.</p> <p>3. The Employee confirms the closing of the application.</p> <p>4. The System writes the pet inventory to an external file.</p> <p>5. The System writes the employee database to an external file.</p> <p>6. The System writes the customer database to an external file.</p> <p>7. The System closes the application.</p>
Extensions:	<p>1. If the Manager does not confirm they want to exit, they can continue working with the System.</p>
Frequency of Use:	When a manager closes the application.
Status:	In Progress
Owner:	Garrett Parker
Priority:	MEDIUM

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

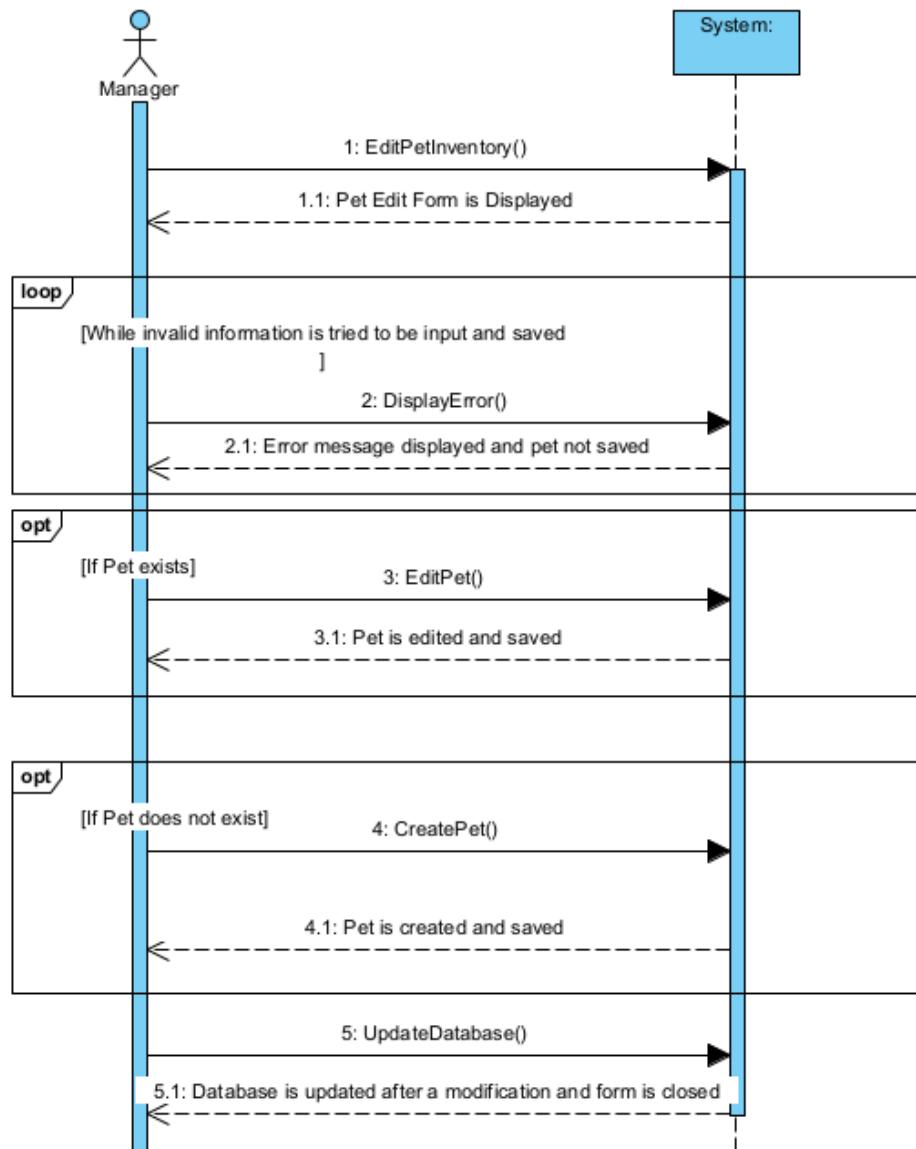
SSD UC 001

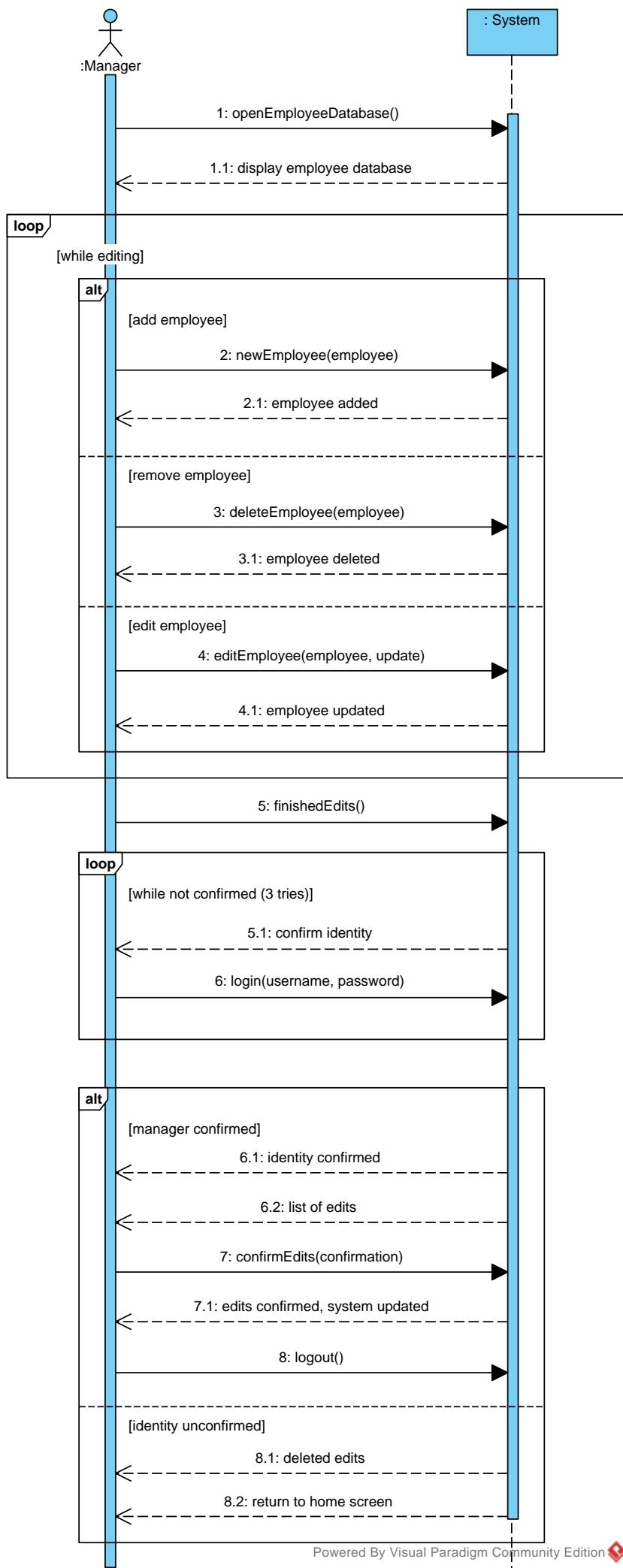


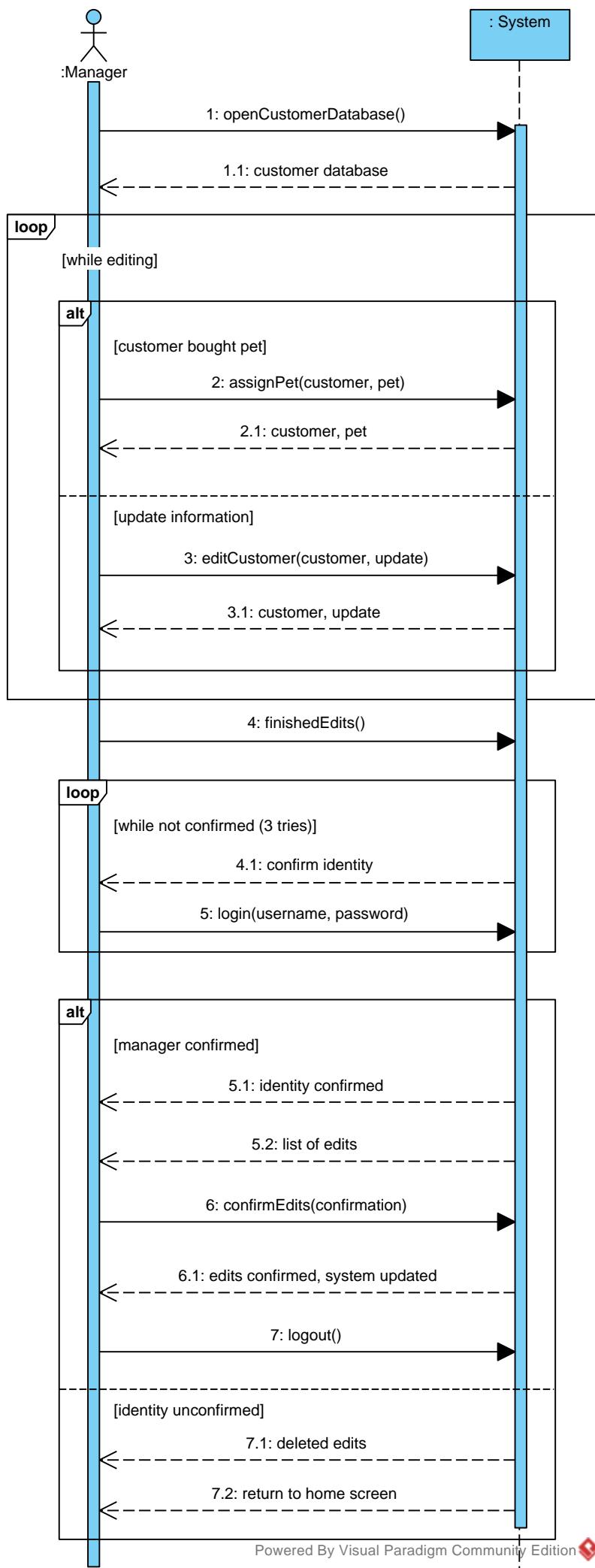
SSD UC 002



SSD UC 003



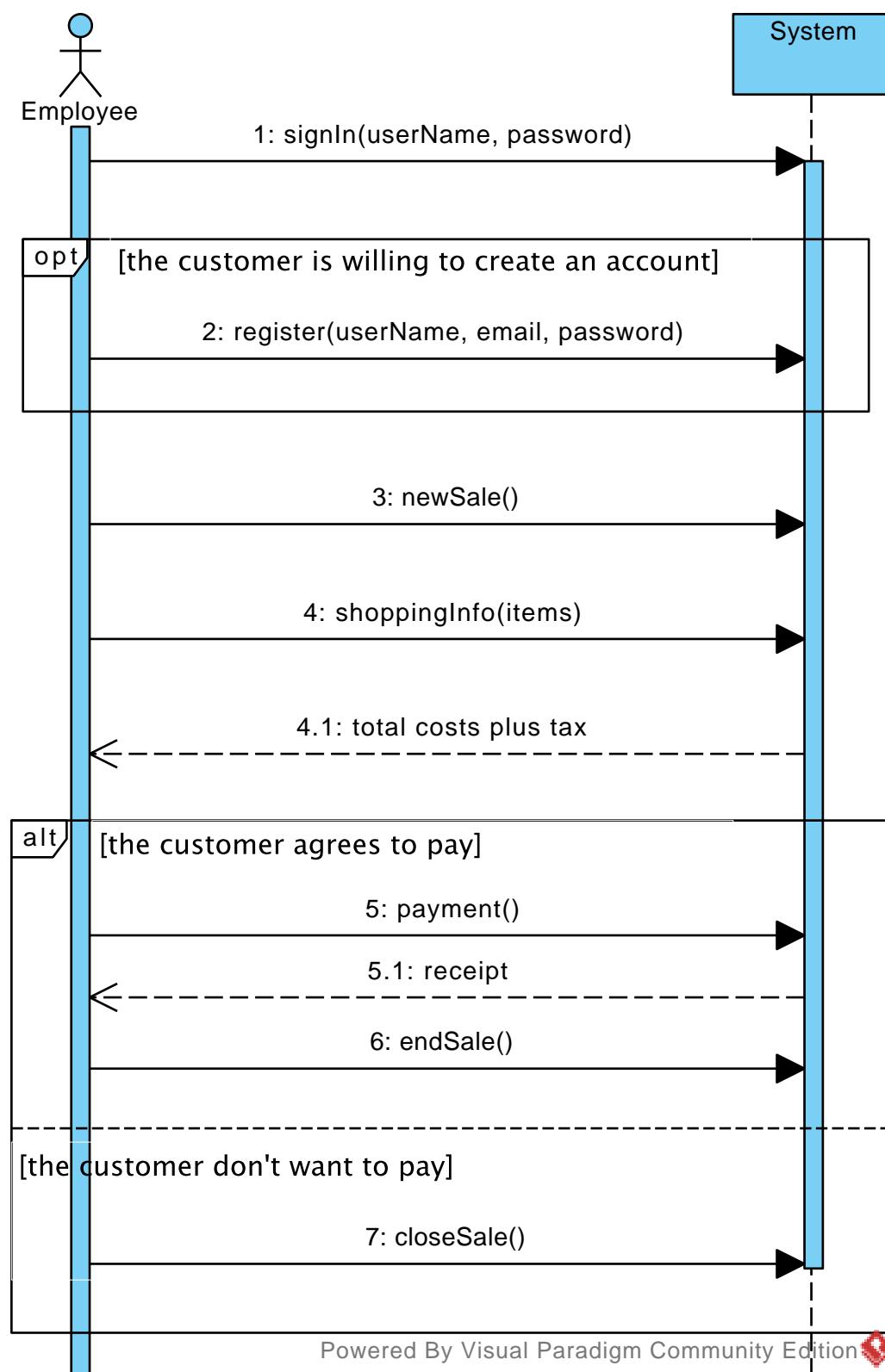


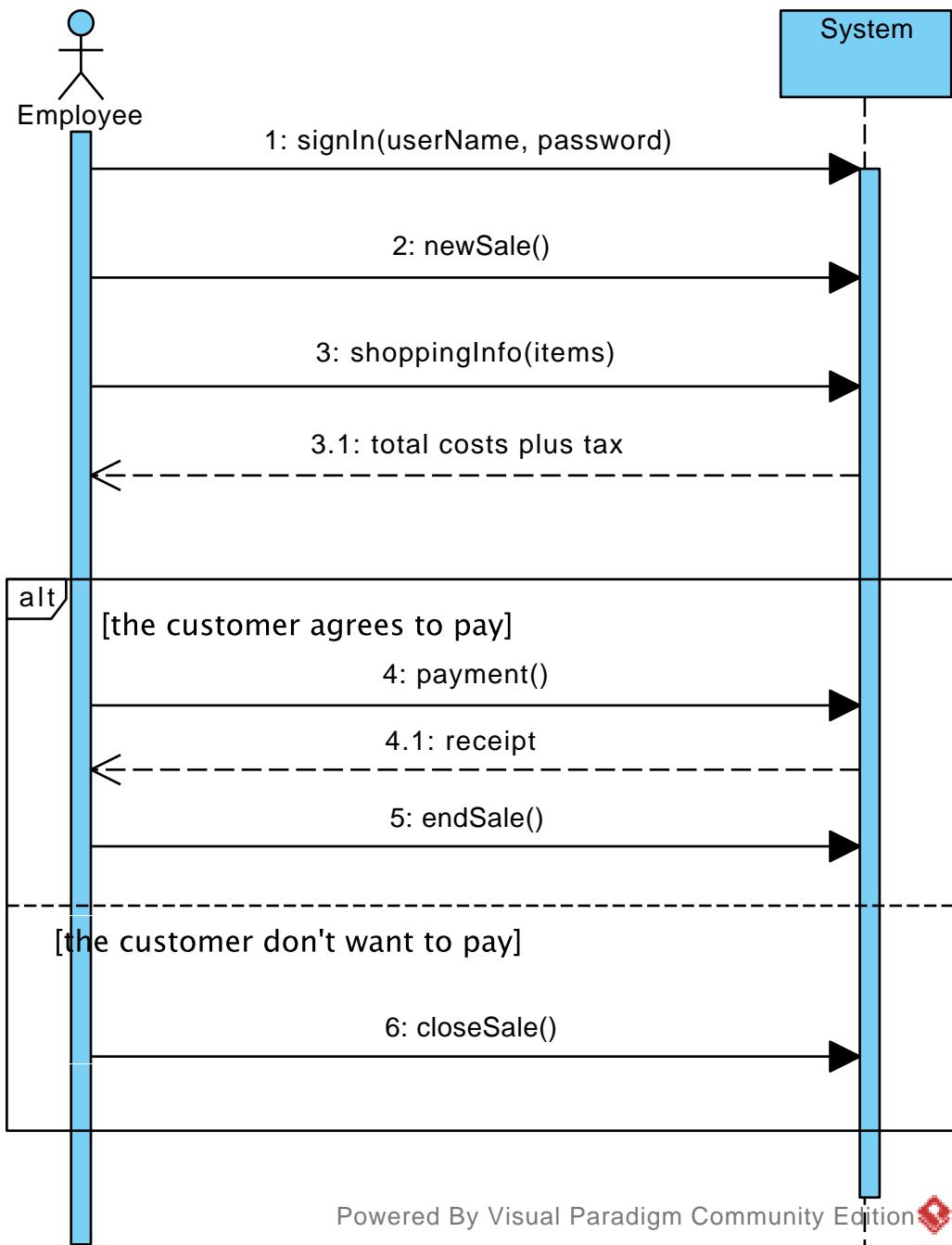


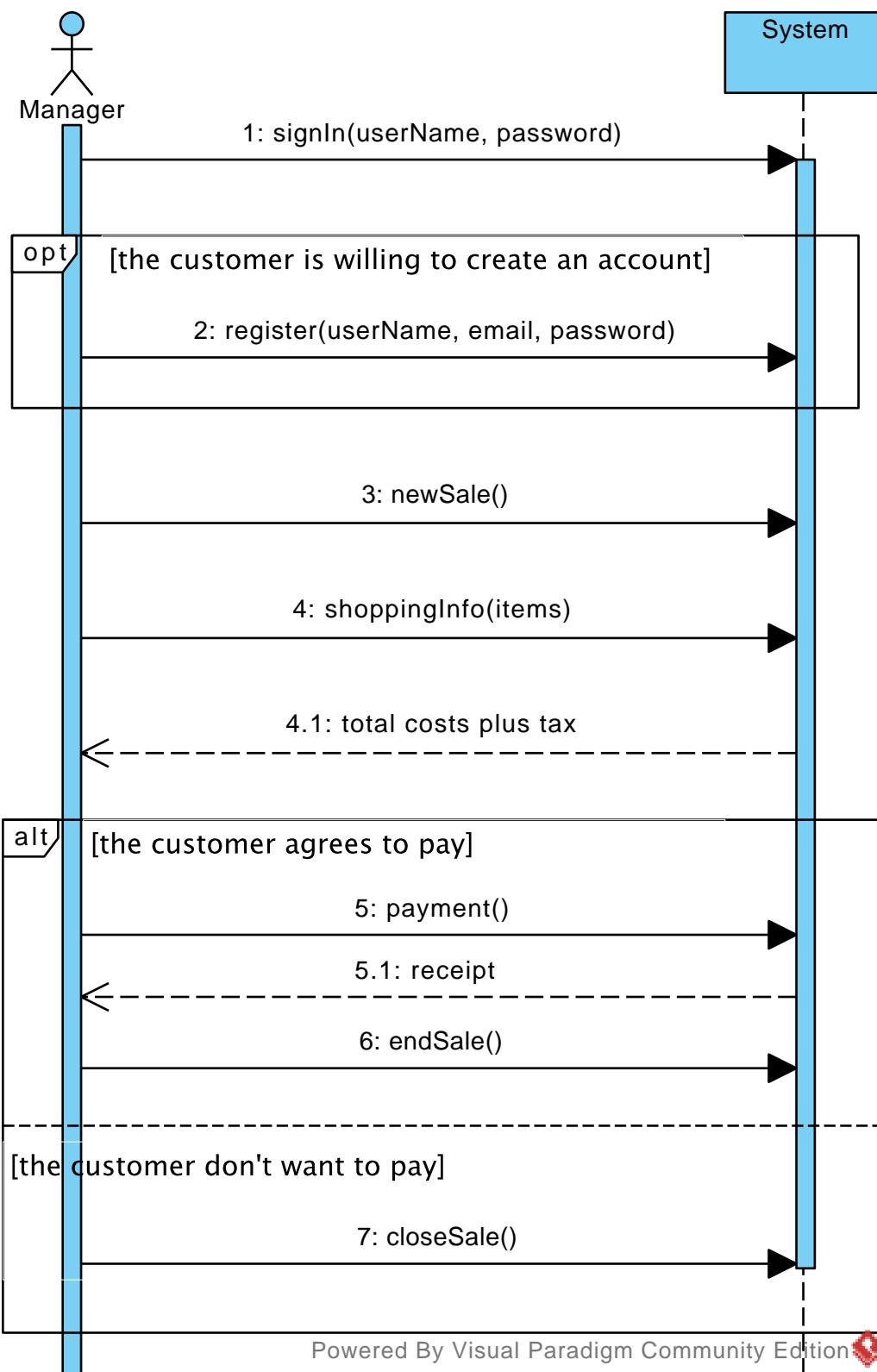
System

- +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 



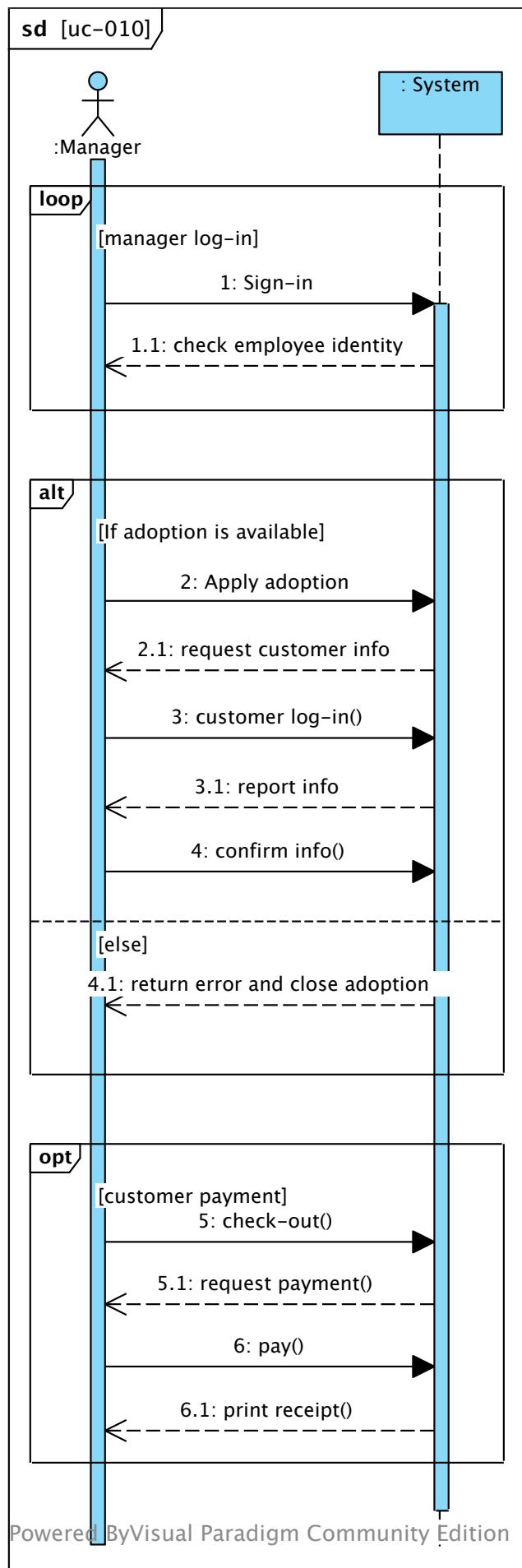




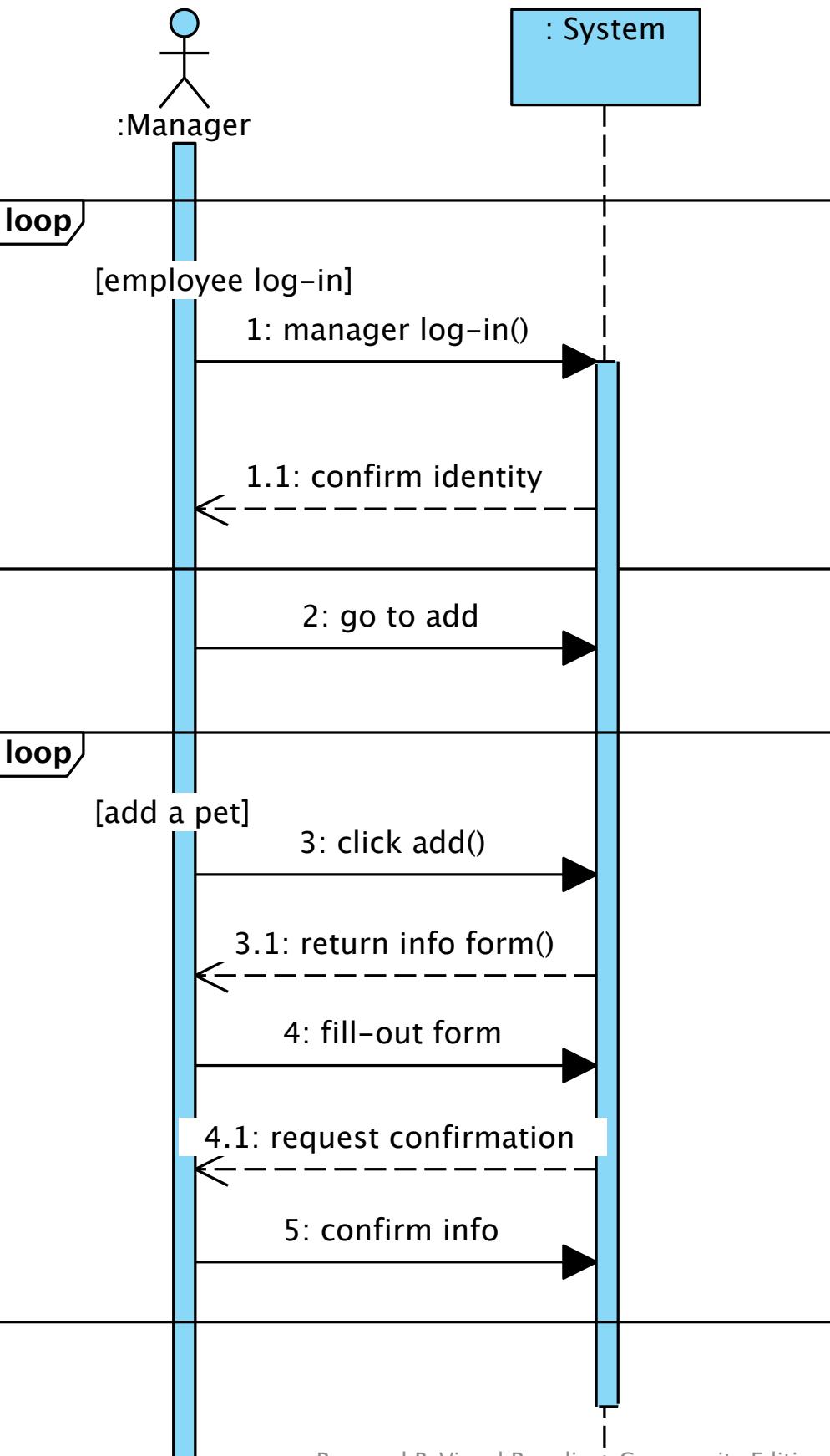
System

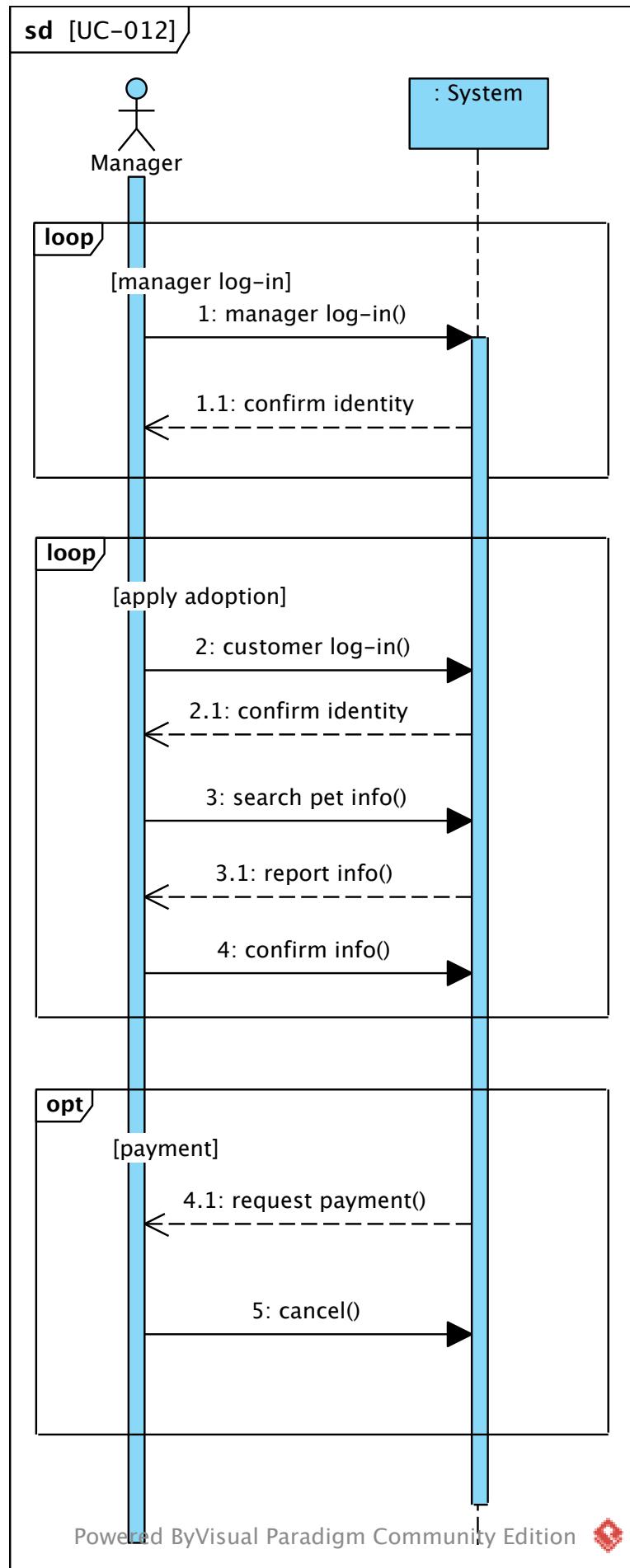
```
+signIn(userName, password)
+register(userName, email, password)
+newSale()
+shoppingInfo(items)
+payment()
+endSale()
+closeSale()
```

Powered By Visual Paradigm Community Edition 



sd [UC-011]



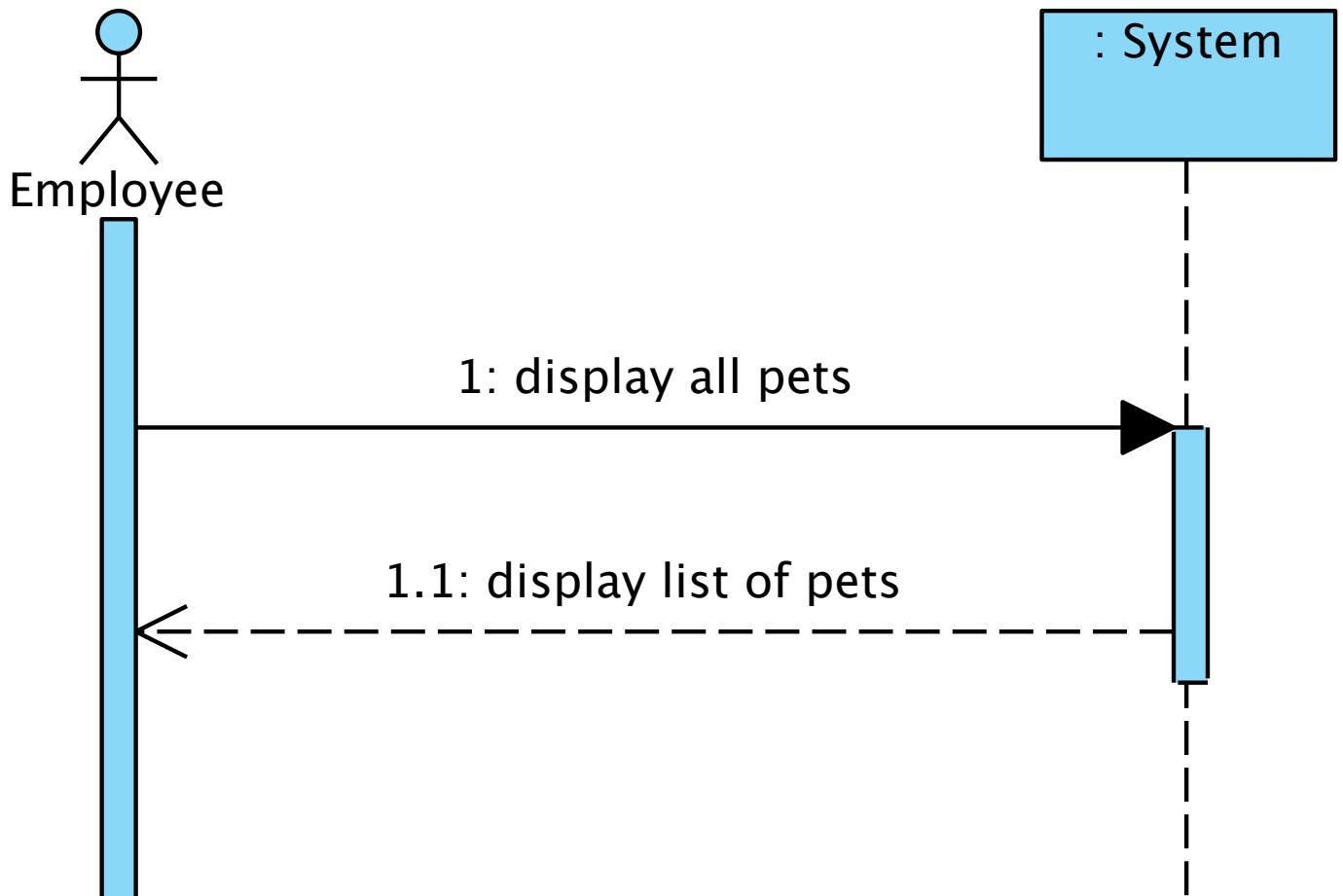


System Operations of UC10-12

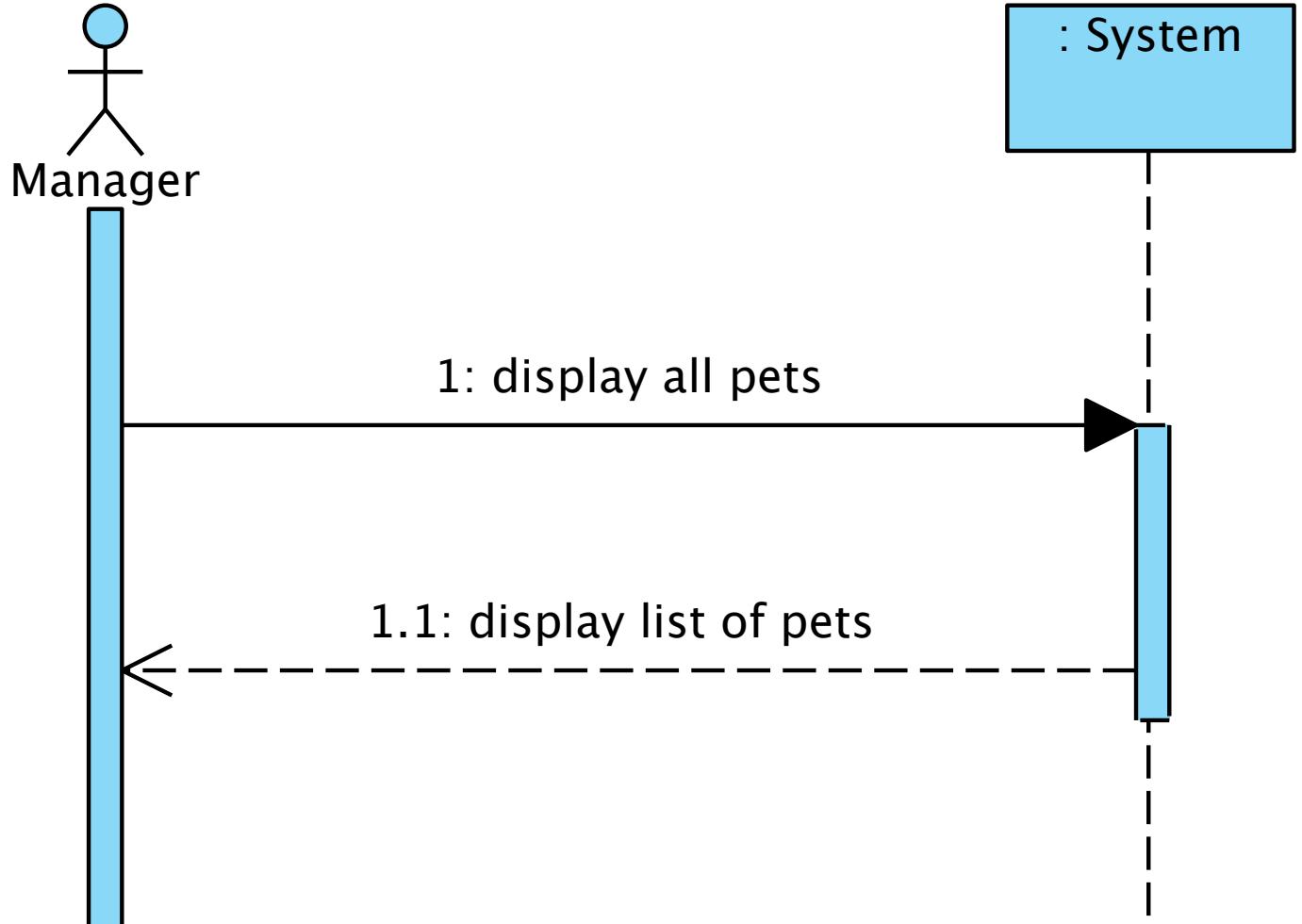
System

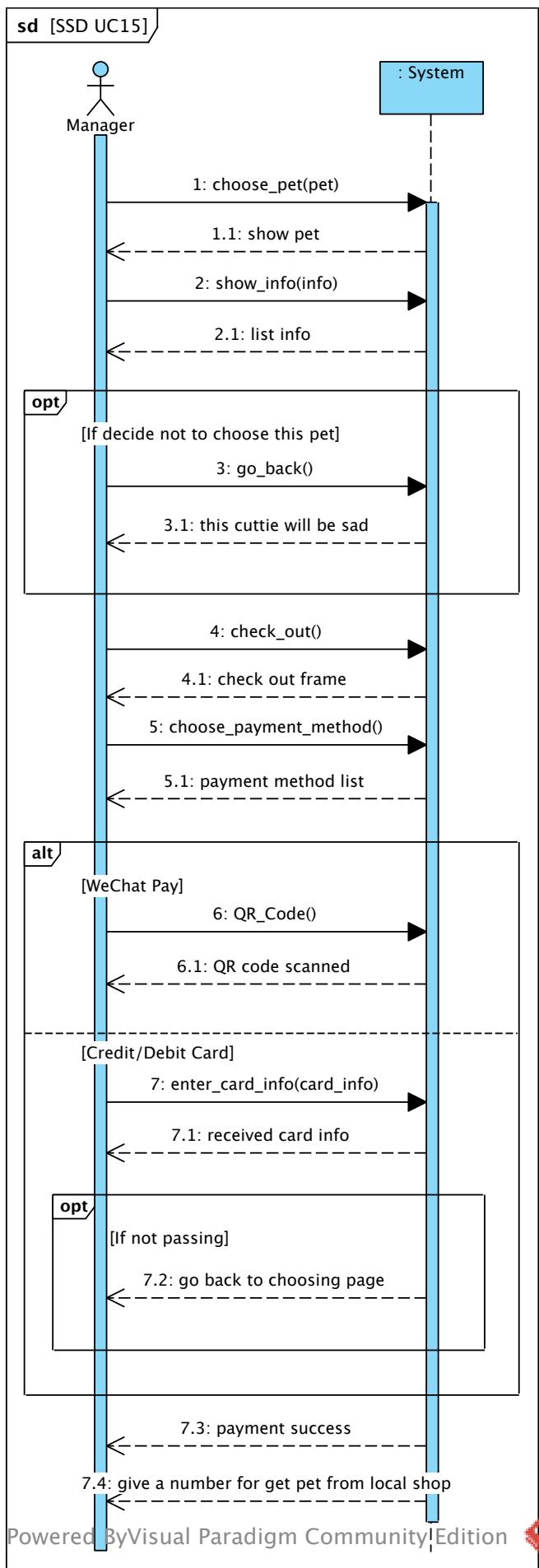
- +manager log in()
- +go to add()
- +click add()
- +fill out form()
- +confirm info()
- +customer log in()
- +search pet info()
- +cancel()
- +sign in()
- +apply adoption()
- +check out()
- +pay()

sd [SSD UC13]



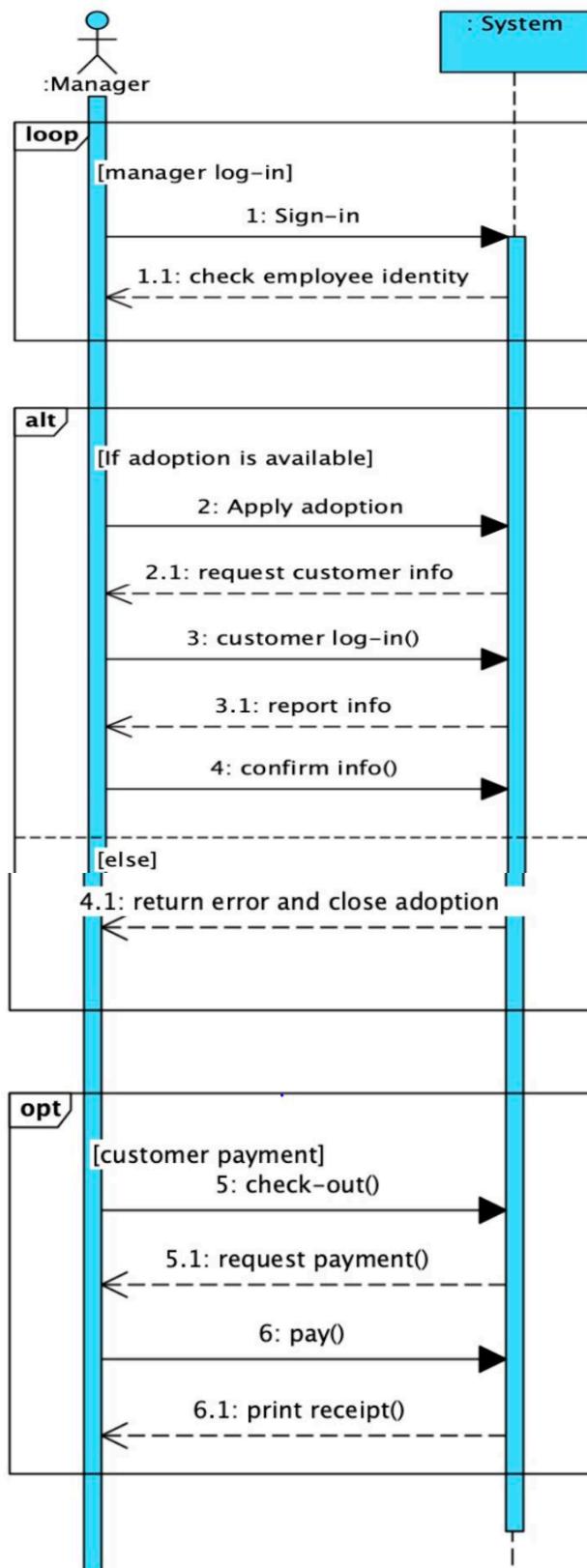
sd [SSD UC14]



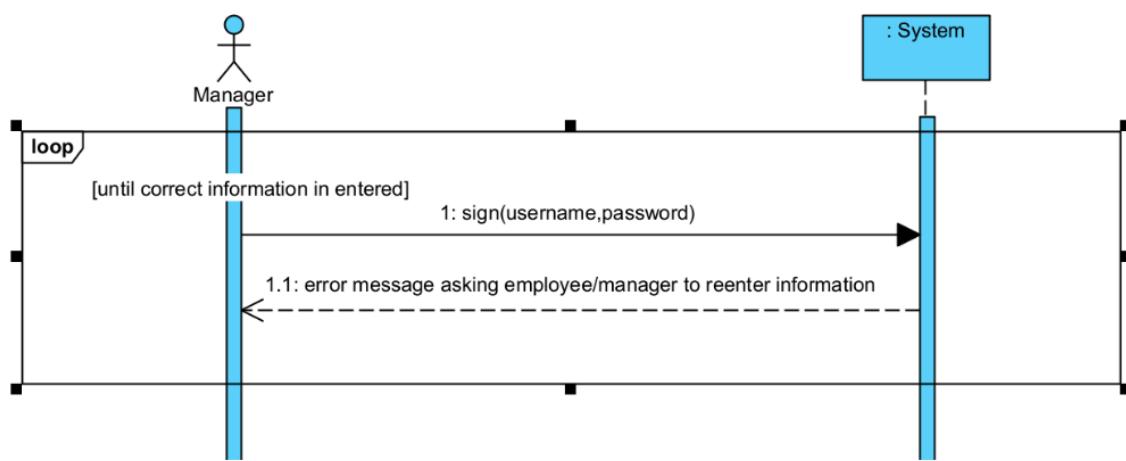


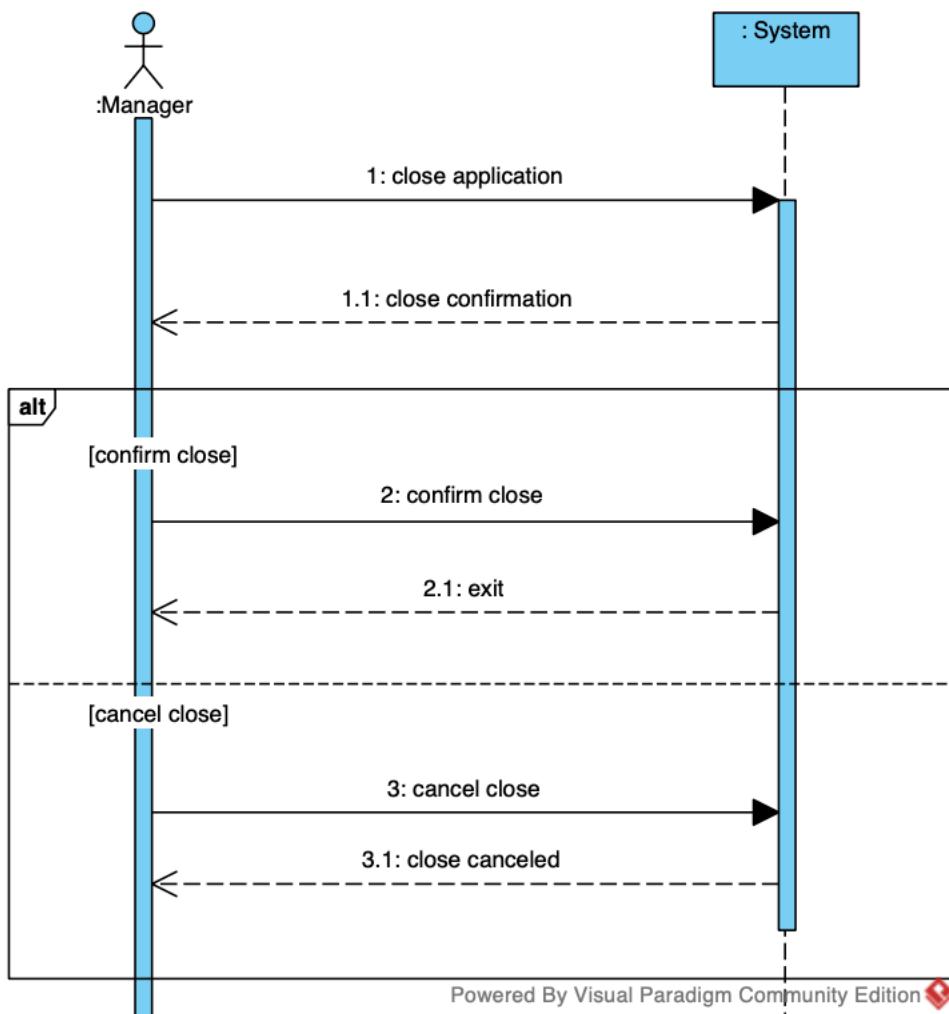
SSD

UC 016



SSD UC 017





SSD UC 020

