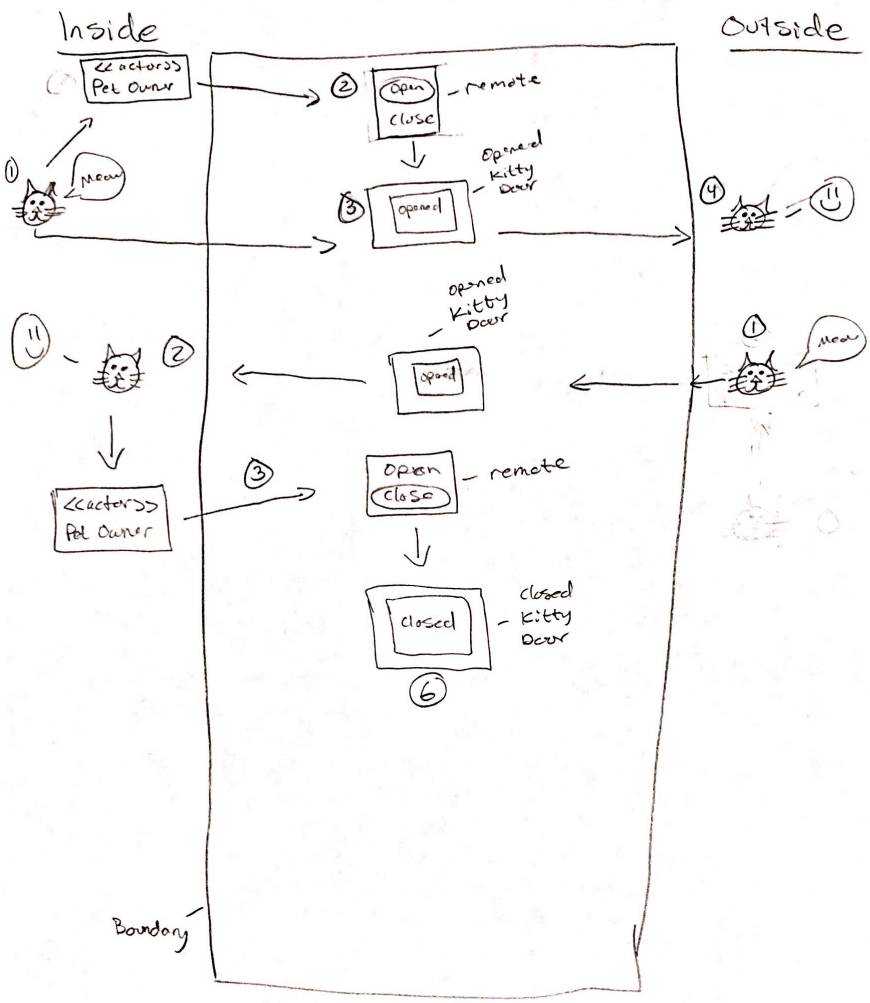
Mubasshir Karim
G498M939
HW5

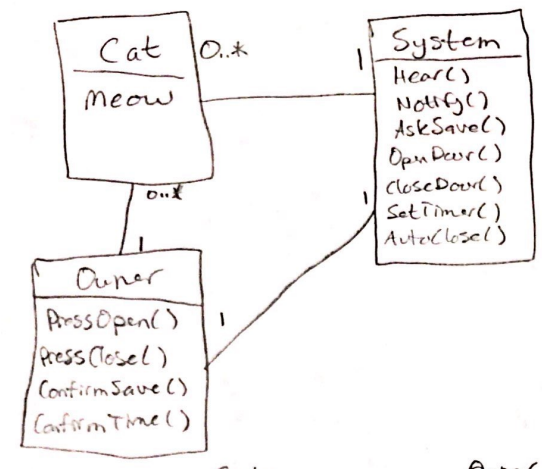
**Part A**

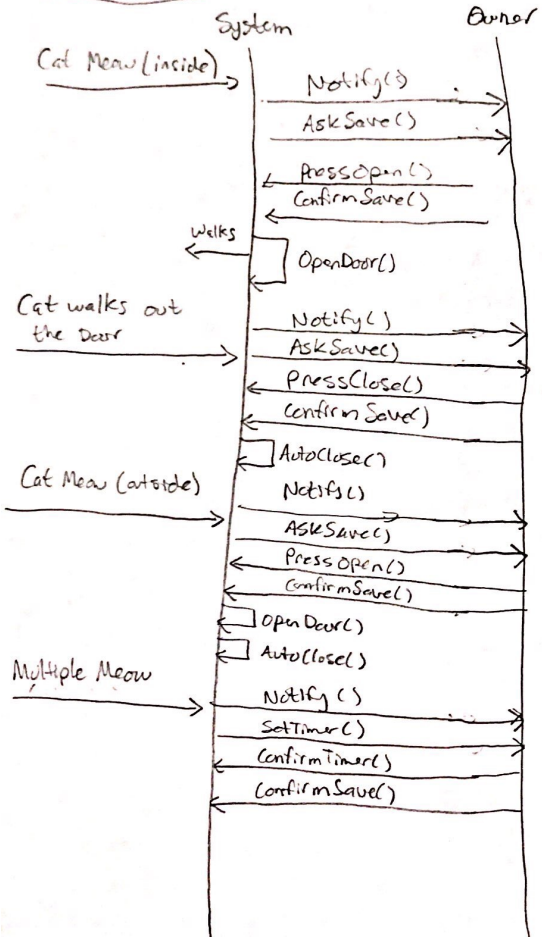
1. Create a UML use case context diagram for use case UC01.



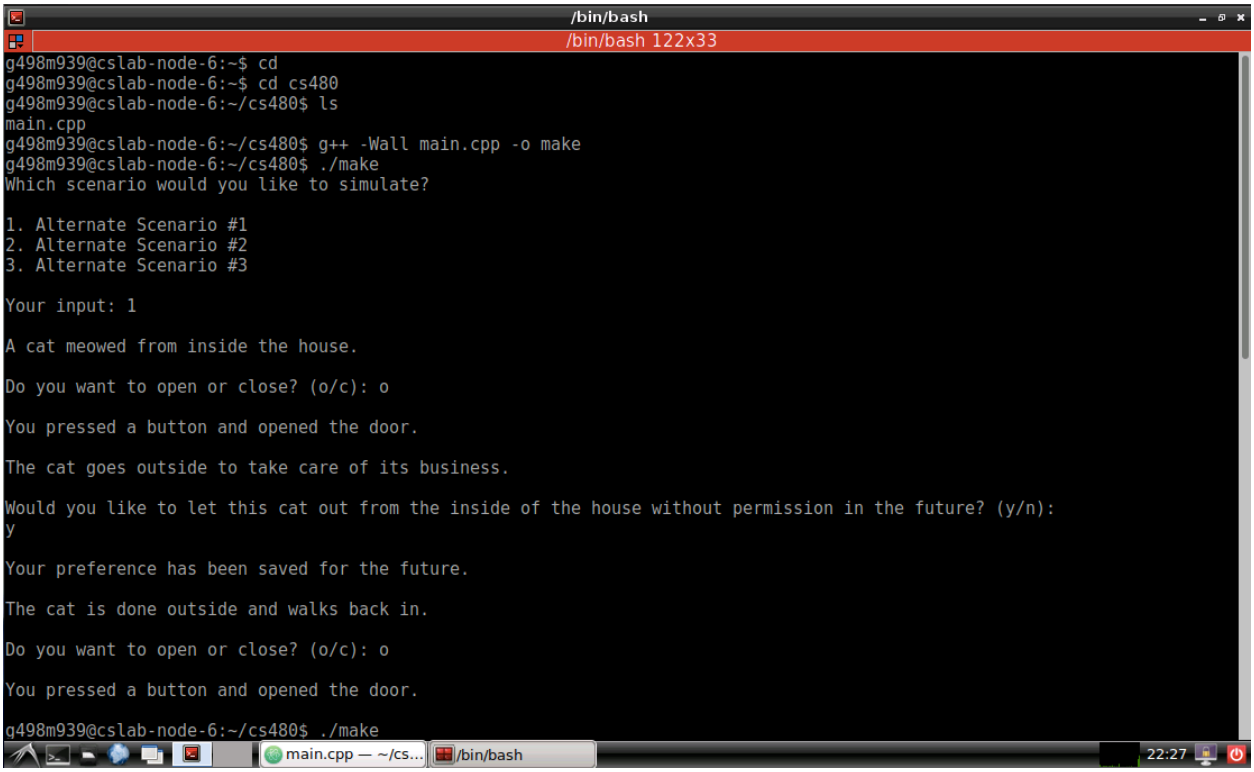
2. UC02 - Electronic Kitty Door
   Main Success Scenario: A cat is meowing, because it wants to either go out of the house or come in the house. The electronic kitty door then…
   **Alternate Scenario 1:** Recognizes the meow from the inside of the house and sends an alert to the user's device stating a cat has meowed and if the electronic kitty door let it go out. The user allows the kitty door to let the cat go out. The electronic kitty door then opens to let the cat out and the cat goes out. The system asks the user if they would like to save this 'meow' in the system for future use, so that the cat can go out as it pleases without the user's permission. The user responds, and the electronic kitty door saves this preference and will remember it for future use.
   **Alternate Scenario 2:** Recognizes the meow from the inside of the house and lets the cat go out with user's permission. However, the electronic kitty door alerts the user if they would like for the door to close automatically as well, after letting the cat out. The user responds to the system and the electronic kitty door saves this preference and will remember it for future use.
   **Alternate Scenario 3:** Recognizes the meow from the outside of the house and alerts the user that it hears a 'saved' cat meowing and wanting to come in. The system for the electronic kitty door will then prompt the user if they would like to allow this cat to come in whenever it pleases without permission. The user responds to the system and the electronic kitty door saves this preference and will remember it for future use.

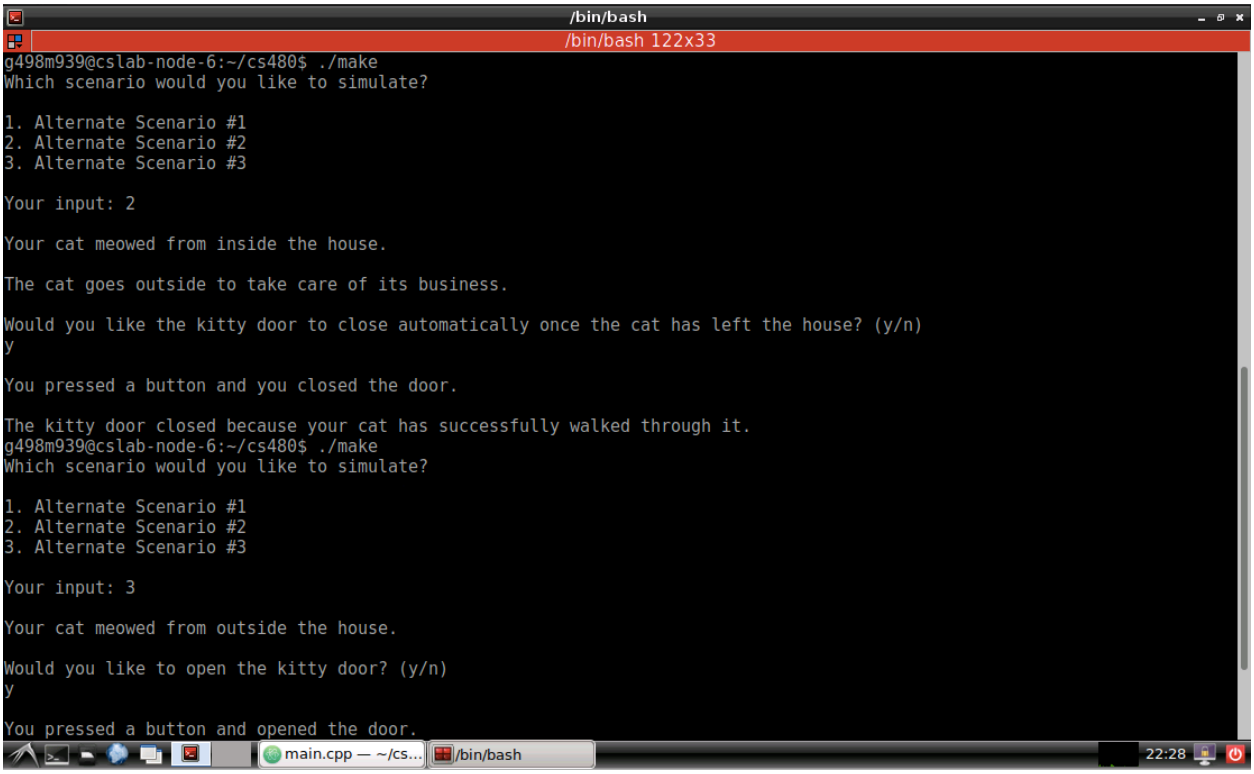
3. Design for use case UC02 using UML sequence and class diagrams.

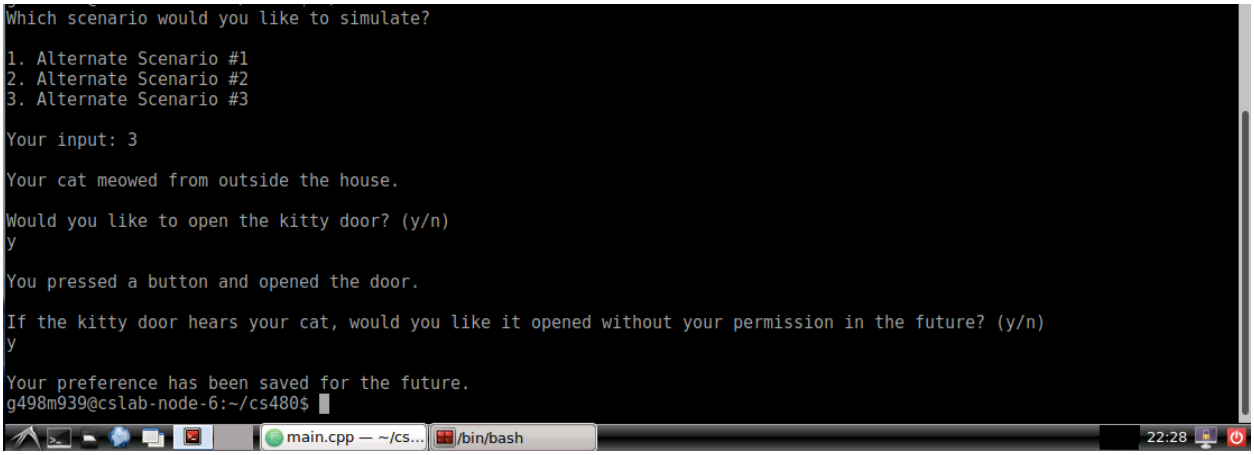


## Part B


/bin/bash
/bin/bash 122x33
g498m939@cslab-node-6:~$ cd
g498m939@cslab-node-6:~$ cd cs480
g498m939@cslab-node-6:~/cs480$ ls
main.cpp
g498m939@cslab-node-6:~/cs480$ g++ -Wall main.cpp -o make
g498m939@cslab-node-6:~/cs480$ ./make
Which scenario would you like to simulate?
1. Alternate Scenario #1
2. Alternate Scenario #2
3. Alternate Scenario #3
Your input: 1
A cat meowed from inside the house.
Do you want to open or close? (o/c): o
You pressed a button and opened the door.
The cat goes outside to take care of its business.
Would you like to let this cat out from the inside of the house without permission in the future? (y/n):
y
Your preference has been saved for the future.
The cat is done outside and walks back in.
Do you want to open or close? (o/c): o
You pressed a button and opened the door.
g498m939@cslab-node-6:~/cs480$ ./make
main.cpp — ~/cs...
/bin/bash
22:27


/bin/bash
/bin/bash 122x33
g498m939@cslab-node-6:~/cs480$ ./make
Which scenario would you like to simulate?
1. Alternate Scenario #1
2. Alternate Scenario #2
3. Alternate Scenario #3
Your input: 2
Your cat meowed from inside the house.
The cat goes outside to take care of its business.
Would you like the kitty door to close automatically once the cat has left the house? (y/n)
y
You pressed a button and you closed the door.
The kitty door closed because your cat has successfully walked through it.
g498m939@cslab-node-6:~/cs480$ ./make
Which scenario would you like to simulate?
1. Alternate Scenario #1
2. Alternate Scenario #2
3. Alternate Scenario #3
Your input: 3
Your cat meowed from outside the house.
Would you like to open the kitty door? (y/n)
y
You pressed a button and opened the door.
main.cpp — ~/cs...
/bin/bash
22:28


Which scenario would you like to simulate?
1. Alternate Scenario #1
2. Alternate Scenario #2
3. Alternate Scenario #3
Your input: 3
Your cat meowed from outside the house.
Would you like to open the kitty door? (y/n)
y
You pressed a button and opened the door.
If the kitty door hears your cat, would you like it opened without your permission in the future? (y/n)
y
Your preference has been saved for the future.
g498m939@cslab-node-6:~/cs480$
main.cpp — ~/cs...
/bin/bash
22:28