Kierstin Matsuda, Kevin Garcia, Eric Cao

COP 5614 – Introduction to Operating Systems

Group 9 – Assignment 2

Professor Dong Chen

Wednesday March 27th

Testing, Performance and Latency

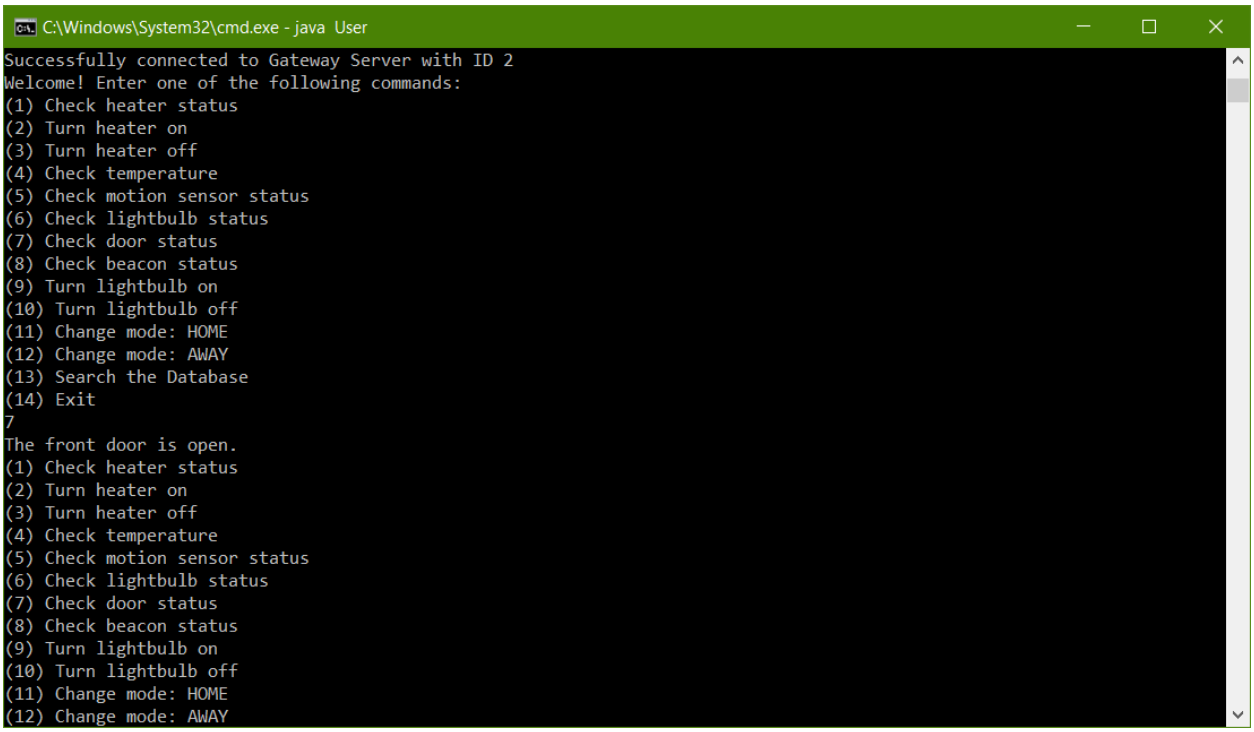
Test Case 1:

Input: User chooses option 7, “Check door status”.

Pre-requisites: The server is running, User.java and the door are connected to it, and the door is open.

Expected result: User receives a text message saying, “The front door is open.”

Actual result: User receives a text message saying, “The front door is open.”



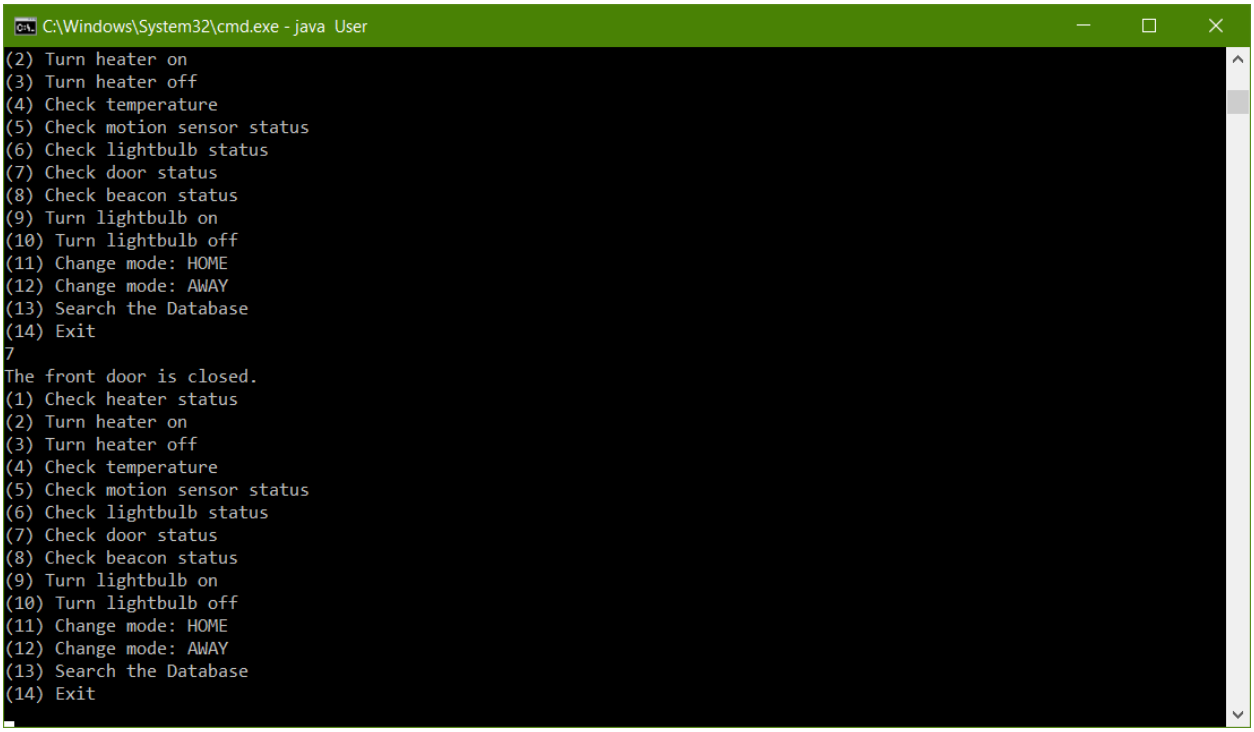
Test Case 2:

Input: User chooses option 7, “Check door status”.

Pre-requisites: The server is running, User.java and the door are connected to it, and the door is closed.

Expected result: User receives a text message saying, “The front door is closed.”

Actual result: User receives a text message saying, “The front door is closed.”



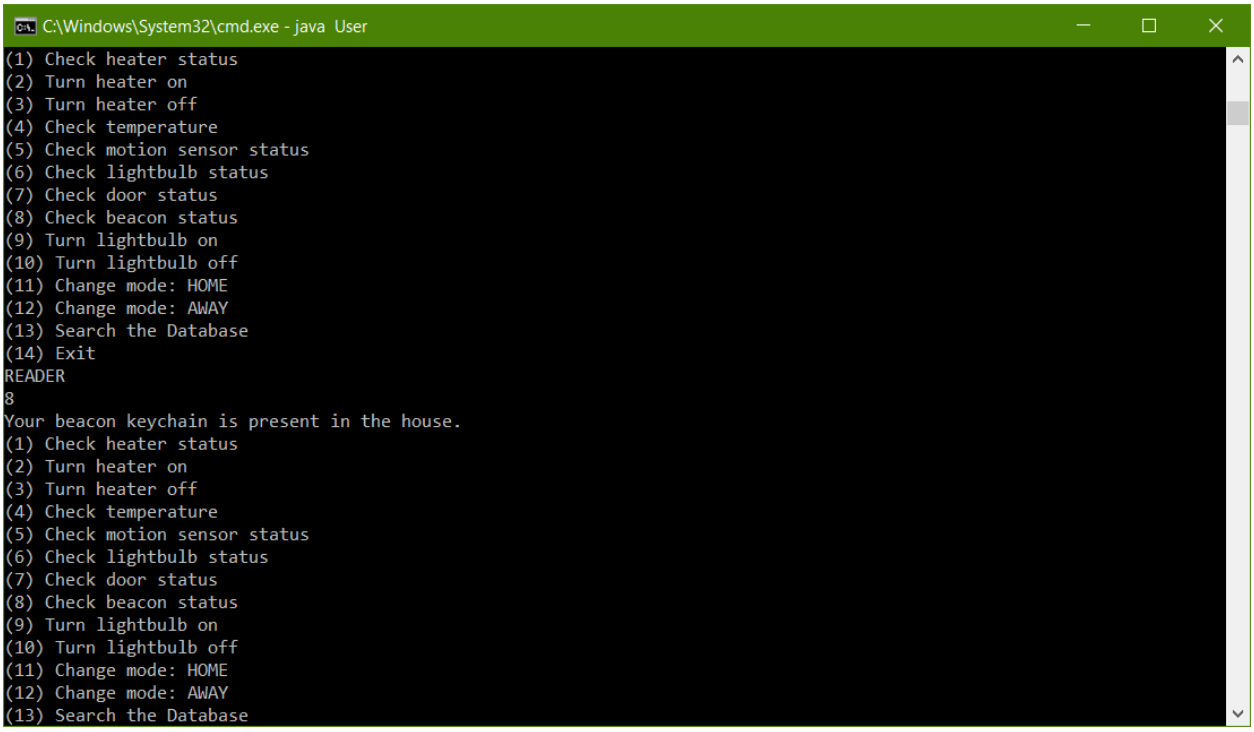
Test Case 3:

Input: User chooses option 8, “Check beacon status”

Pre-requisites: The server is running, User.java and the beacon are connected to it, and the beacon is in the house.

Expected result: User receives a text message saying, "Your beacon keychain is present in the house."

Actual result: User receives a text message saying, "Your beacon keychain is present in the house."



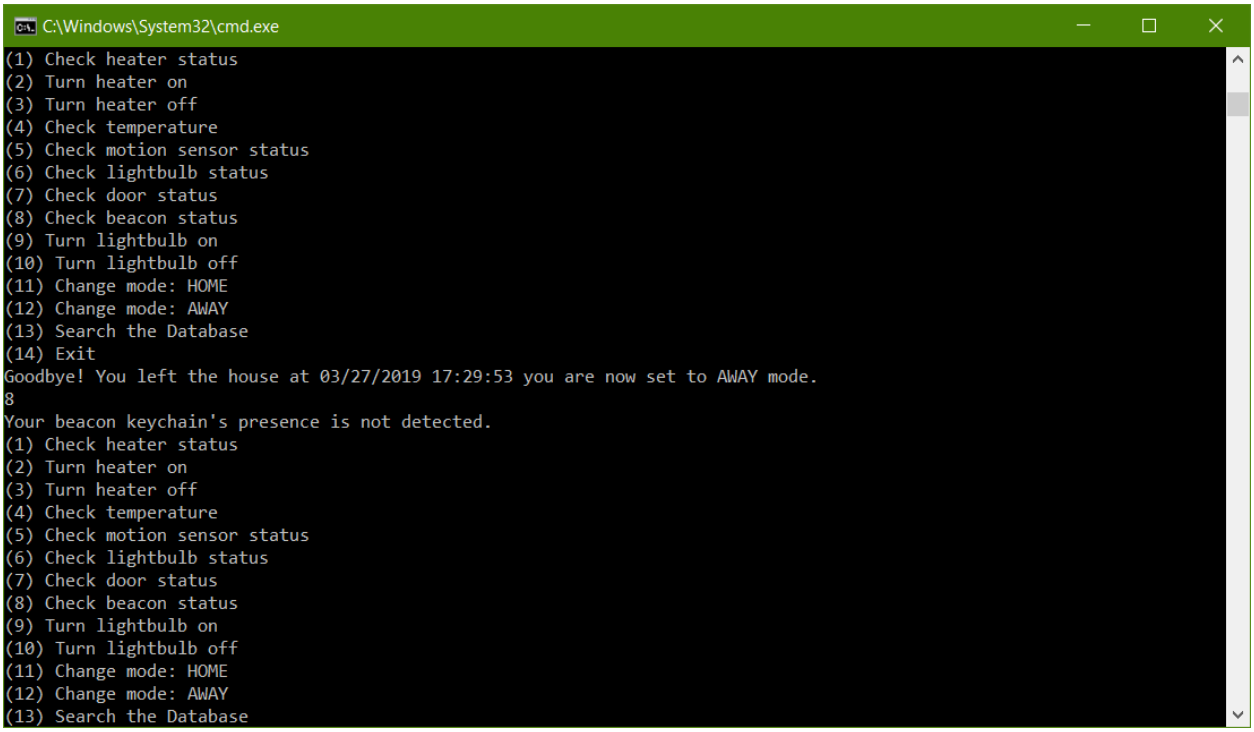
Test Case 4:

Input: User chooses option 8, “Check beacon status”

Pre-requisites: The server is running, User.java and the beacon are connected to it, and the beacon is not in the house.

Expected result: User receives a text message saying, "Your beacon keychain's presence is not detected."

Actual result: User receives a text message saying, "Your beacon keychain's presence is not detected."



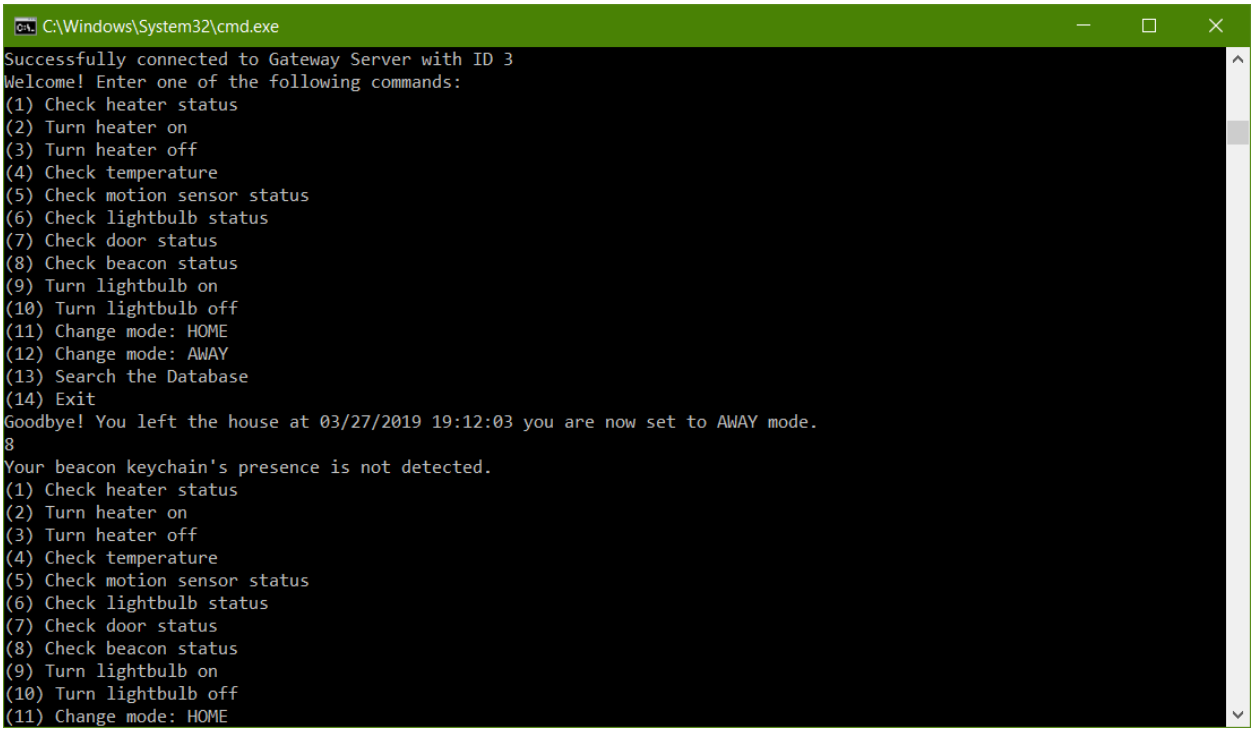
Test Case 5:

Input: Motion is sensed first then the door is opened

Pre-requisites: The server is running and User.java and the door are connected to it.

Expected result: User receives a text message saying, "Goodbye! You left the house at x time. You are now set to AWAY mode." and the security system is set to AWAY.

Actual result: User receives a text message saying, "Goodbye! You left the house at x time. You are now set to AWAY mode." and the security system is set to AWAY.



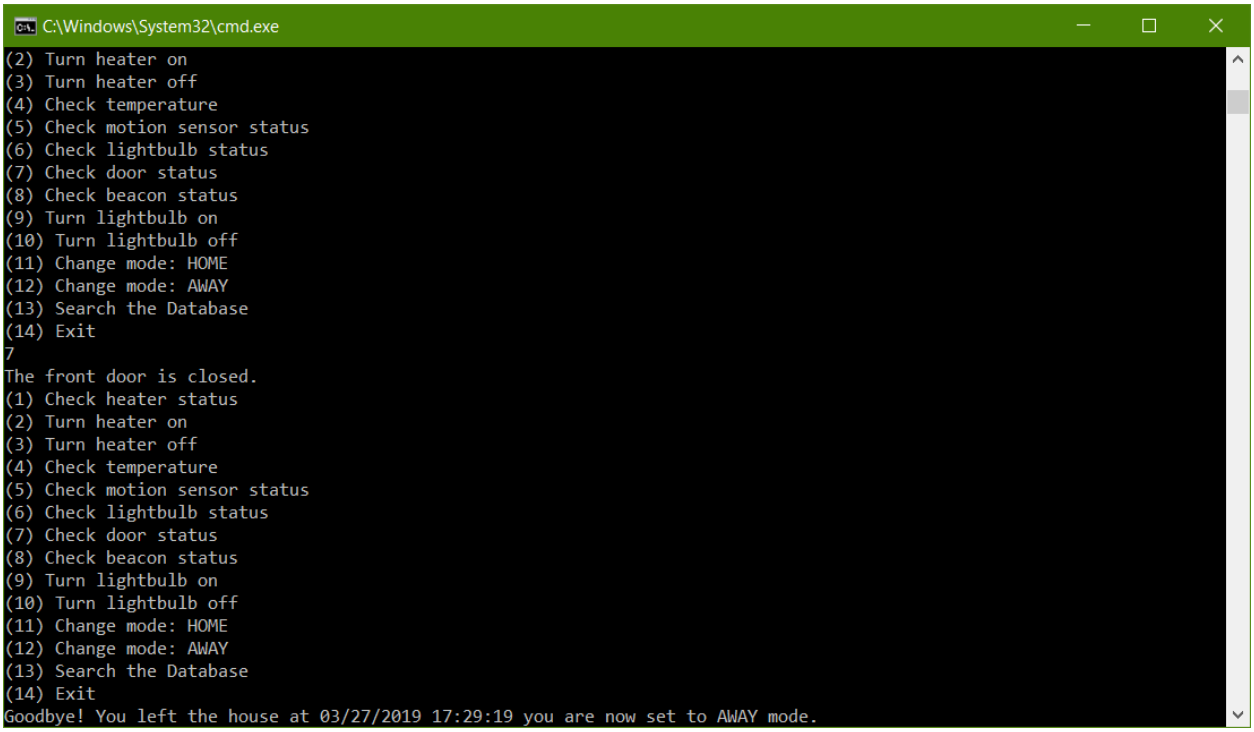
Test Case 6:

Input: Motion is sensed first then the door is opened and the beacon is present.

Pre-requisites: The server is running and User.java, the door, and the beacon are connected to it.

Expected result: User receives a text message saying, "Goodbye! You left the house at x time. You are now set to AWAY mode." and the security system is set to AWAY.

Actual result: User receives a text message saying, "Goodbye! You left the house at x time. You are now set to AWAY mode." and the security system is set to AWAY.



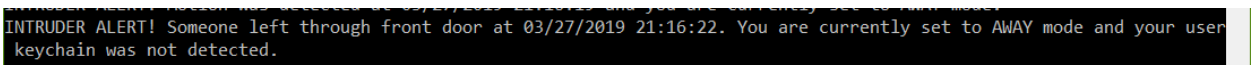
Test Case 7:

Input: Motion is sensed first then the door is opened and the beacon is not present.

Pre-requisites: The server is running and User.java, the door, and the beacon are connected to it.

Expected result: User receives a text message saying INTRUDER ALERT! Someone left through front door at x time. You are currently set to AWAY mode and your user keychain was not detected."

Actual result: User receives a text message saying INTRUDER ALERT! Someone left through front door at x time. You are currently set to AWAY mode and your user keychain was not detected."



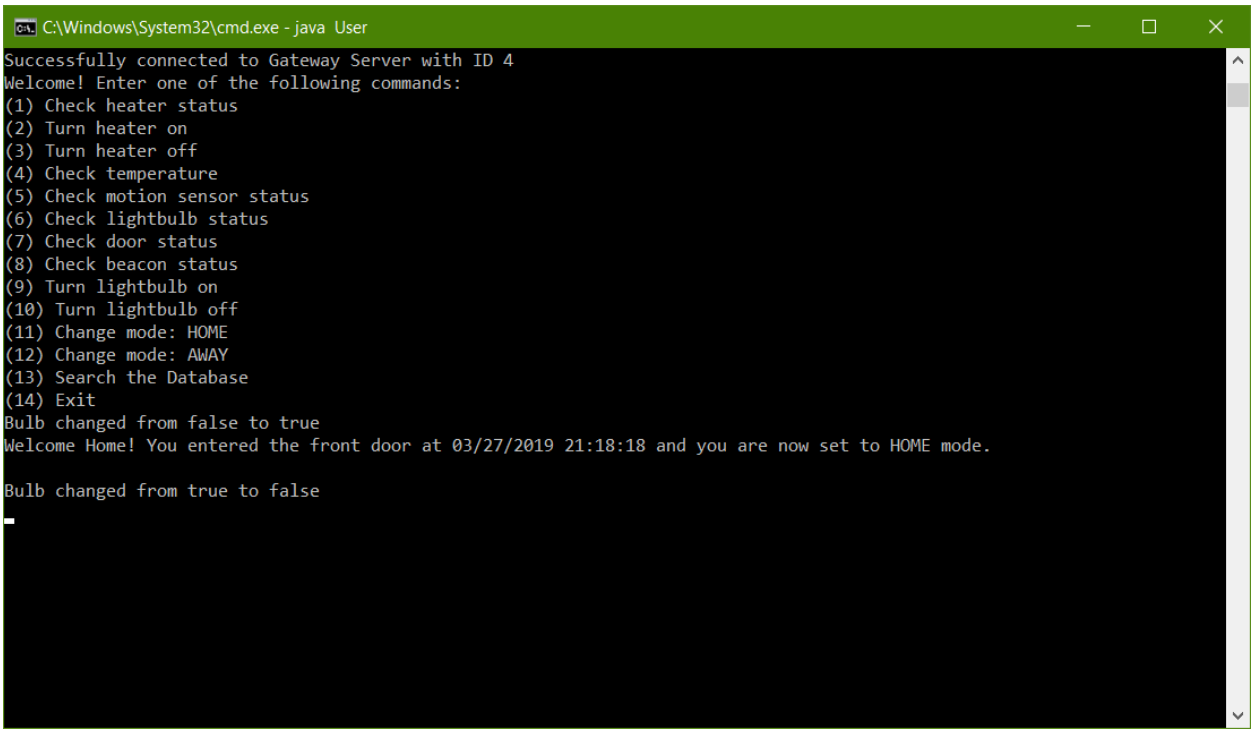
Test Case 8:

Input: Door opens first then motion is sensed.

Pre-requisites: The server is running and User.java and the door are connected to it. The security system is set to AWAY.

Expected result: User receives a text message saying, "Welcome Home! You entered the front door at x time and you are now set to HOME mode.", the security system is set to HOME, and the lights turn on.

Actual result: User receives a text message saying, "Welcome Home! You entered the front door at x time and you are now set to HOME mode.", the security system is set to HOME, and the lights turn on.



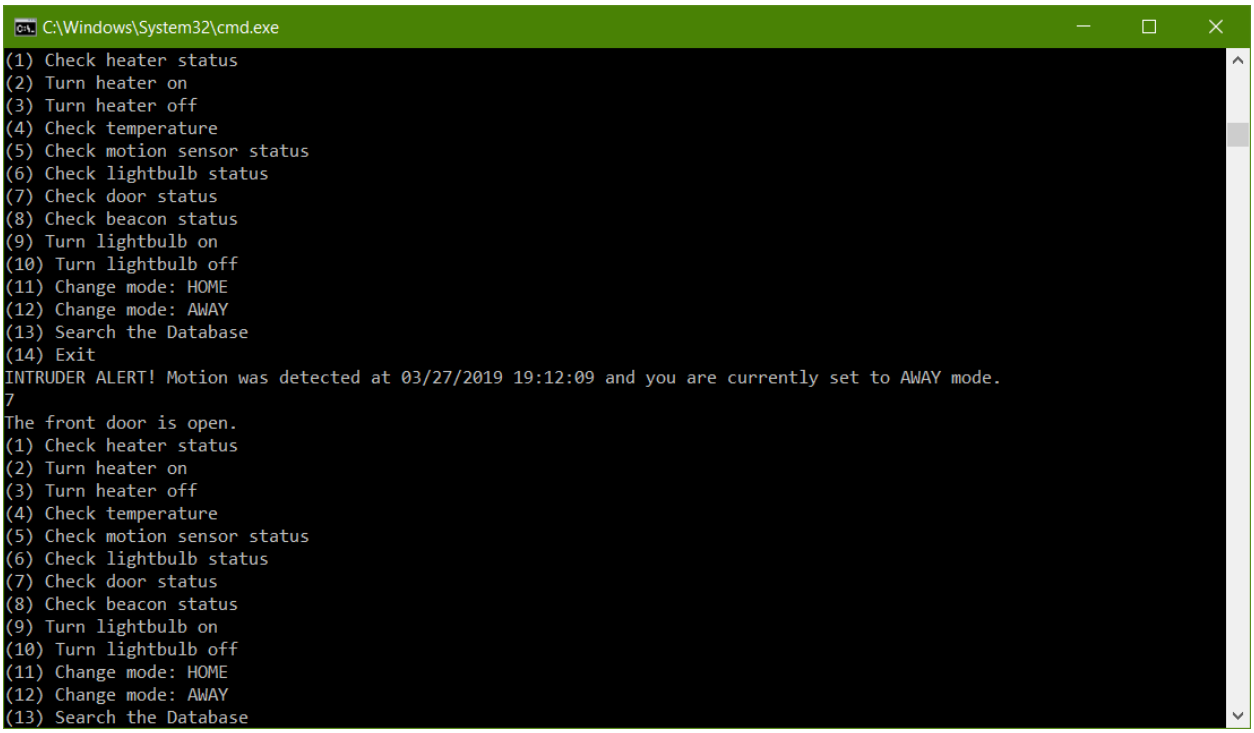
Test Case 9:

Input: Door opens first then motion is sensed but no beacon is detected.

Pre-requisites: The server is running and User.java and the door are connected to it. The security system is set to AWAY.

Expected result: User receives a text message saying "INTRUDER ALERT! Motion was detected at x time and you are currently set to AWAY mode."

Actual result: User receives a text message saying "INTRUDER ALERT! Motion was detected at x time and you are currently set to AWAY mode."



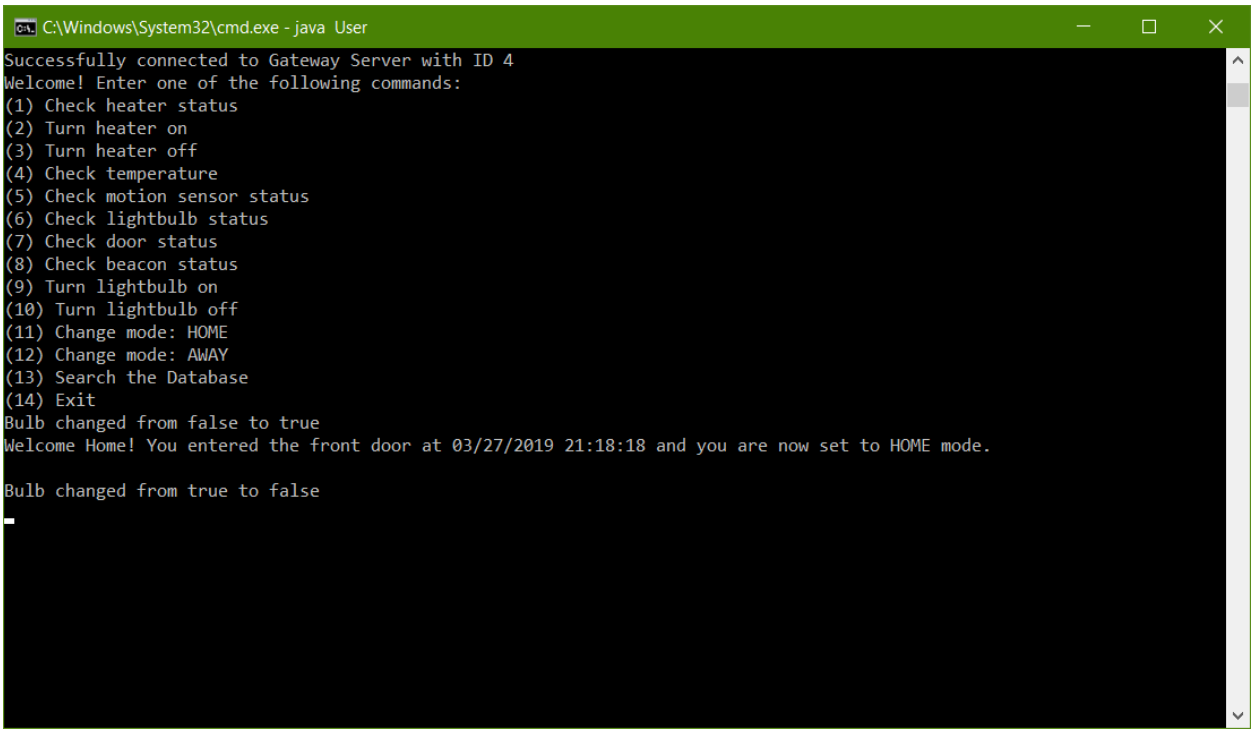
Test Case 10:

Input: Door opens first then motion is sensed and beacon is detected.

Pre-requisites: The server is running and User.java and the door are connected to it. The security system is set to AWAY.

Expected result: User receives a text message saying, "Welcome Home! You entered the front door at x time and you are now set to HOME mode.", the security system is set to HOME, and the lights turn on.

Actual result: User receives a text message saying, "Welcome Home! You entered the front door at x time and you are now set to HOME mode.", the security system is set to HOME, and the lights turn on.



Test Case 11:

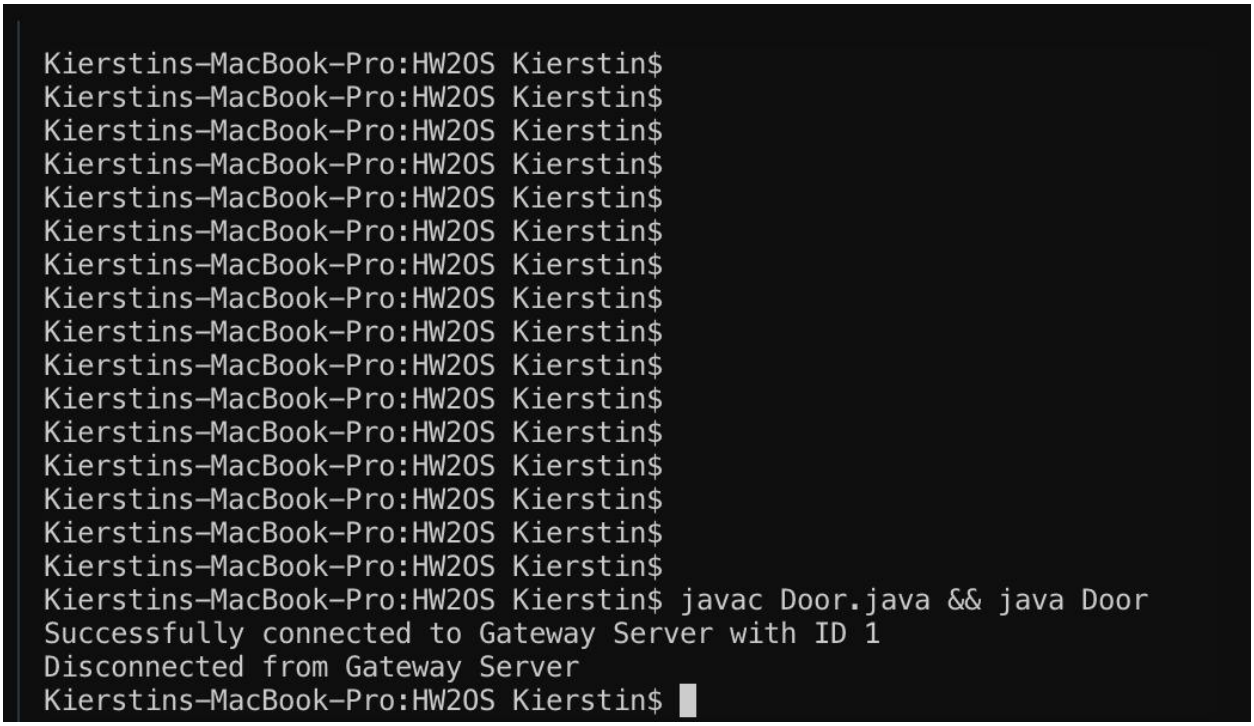
Input: Door stops running unexpectedly.

Pre-requisites: The server and the door are running.

Expected result: Door successfully stops, the server keeps running with no error, and server prints out the message “Disconnected from Gateway Server”, and that thread stops.

Actual result: Door successfully stops, the server keeps running with no error, and server prints

out the message “Disconnected from Gateway Server”, and that thread stops.



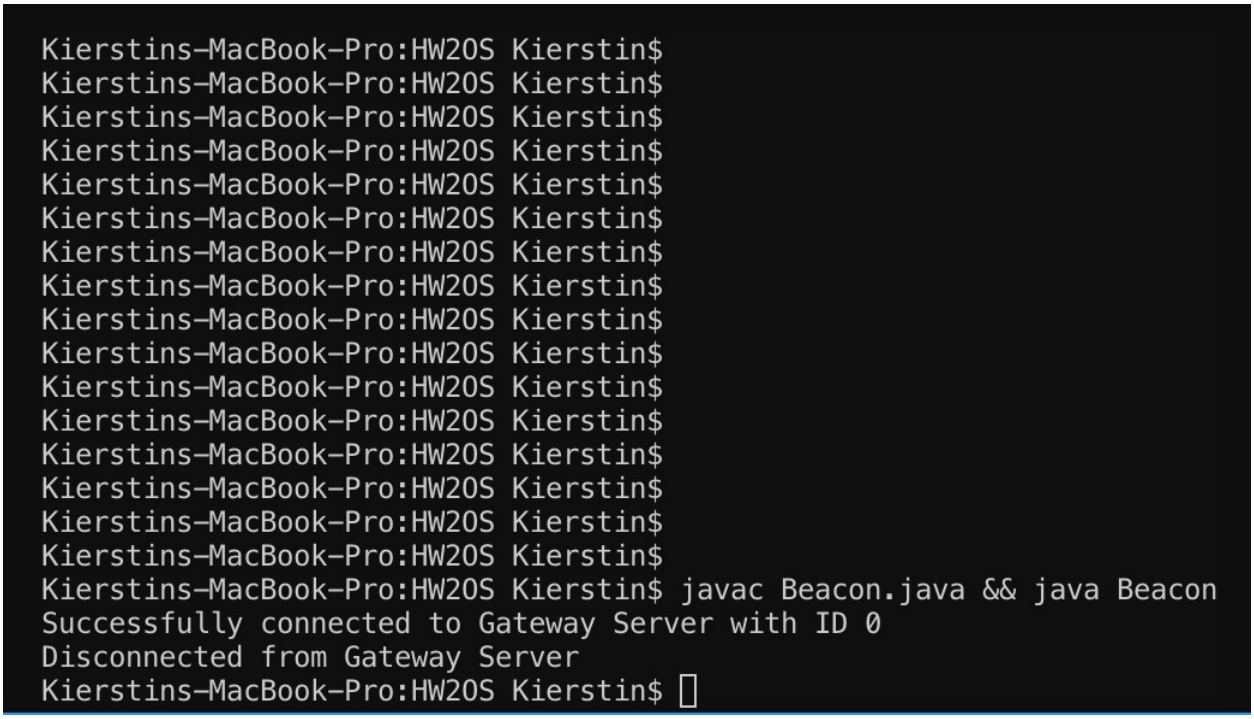
Test Case 12:

Input: Beacon stops running unexpectedly.

Pre-requisites: The server and the beacon are running.

Expected result: Beacon successfully stops, the server keeps running with no error, and server prints out the message “Disconnected from Gateway Server”, and that thread stops.

Actual result: Beacon successfully stops, the server keeps running with no error, and server prints out the message “Disconnected from Gateway Server”, and that thread stops.



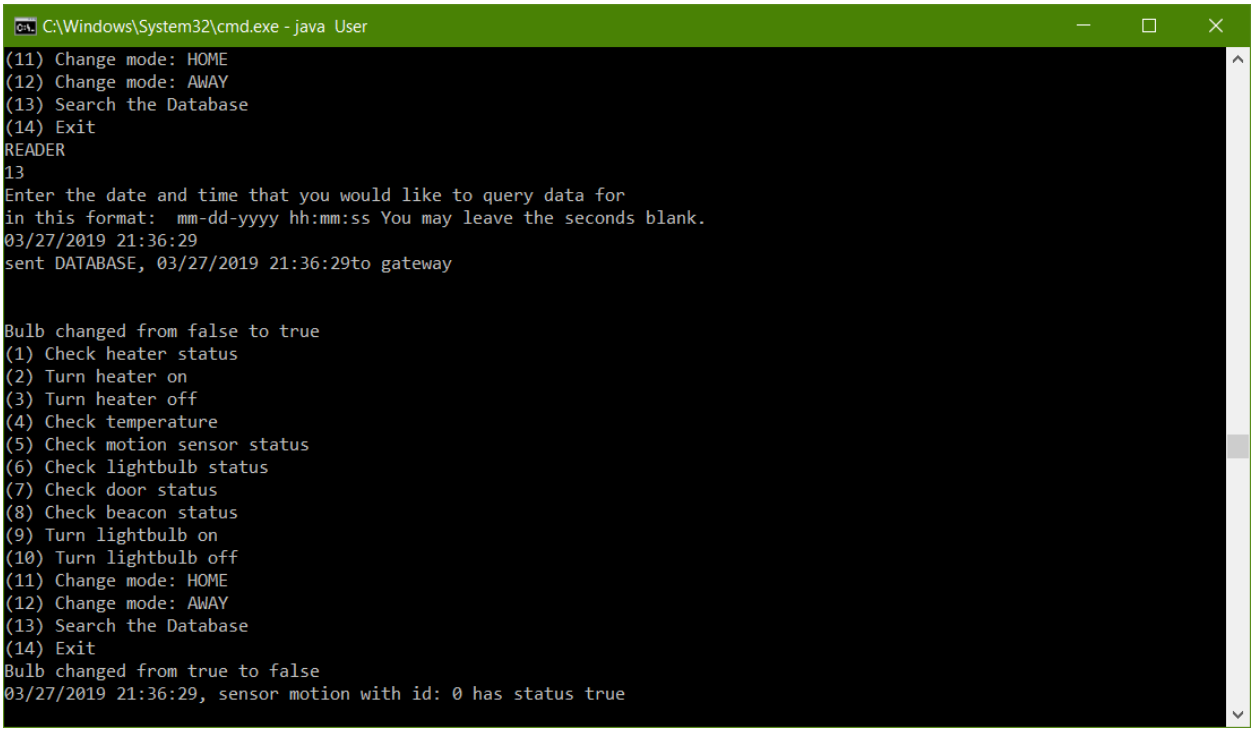
Test Case 13:

Input: User chooses option 13, “Search the Database”

Pre-requisites: The server is running and User.java is connected to it. The user inputs the date and time in the specified format.

Expected result: User receives a line of text with all events that happened during that time and the status of every device and sensor connected at the time.

Actual result: User receives a line of text with all events that happened during that time and the status of every device and sensor connected at the time.



Test Case 14:

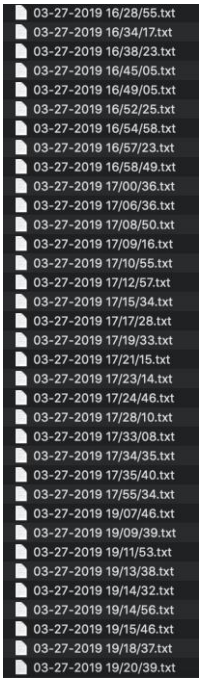
Input: Database server crashes, then user reboots it.

Pre-requisites: The server is running and User.java is connected to it.

Expected result: Database server keeps all past log files despite crashing.

Actual result: Database server keeps all past log files despite crashing.

```
Kevins-MacBook-Pro-2:HW2OS kingkev$ javac DatabaseServer.java && java DatabaseServer
Successfully connected to Gateway Server
Disconnected from Gateway Server
Finished Writing to files
Kevins-MacBook-Pro-2:HW2OS kingkev$
```



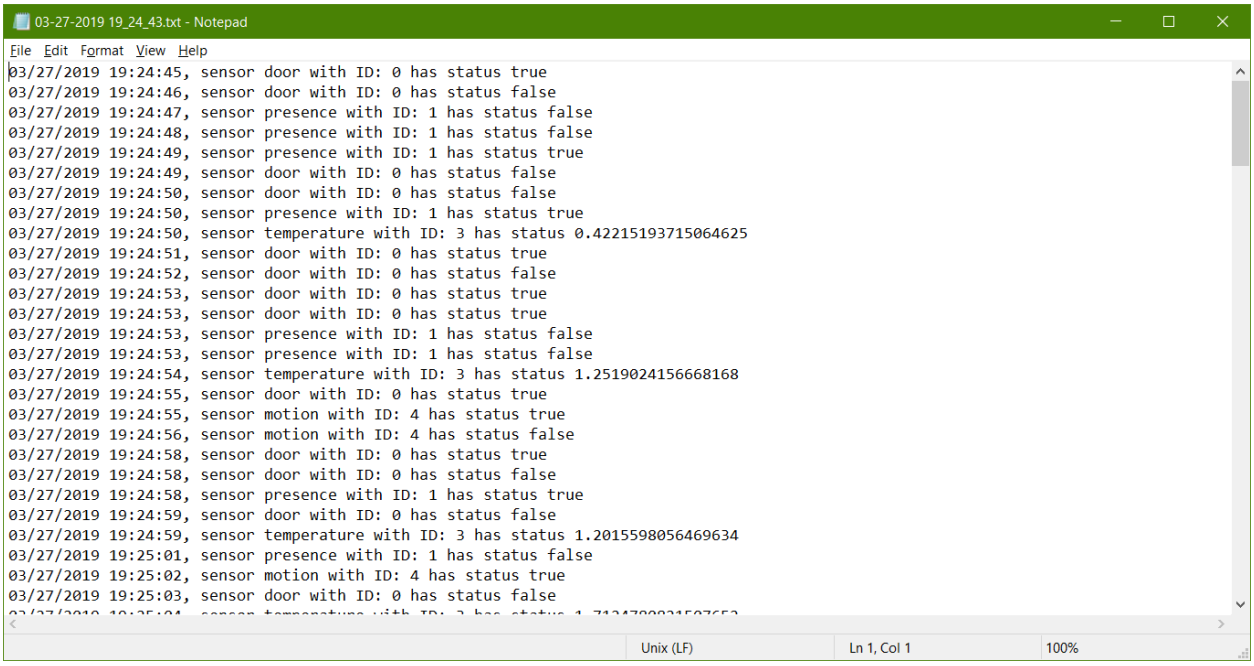
Test Case 15:

Input: Door has opened.

Pre-requisites: The server is running and door is connected to it. The door is initially closed.

Expected result: The server stores that information into a text file titled the time and date the sensor recorded at. It is stored as “x time, sensor door with ID: x is true”.

Actual result: The server stores that information into a text file titled the time and date the sensor recorded at. It is stored as “x time, sensor door with ID: x is true”.



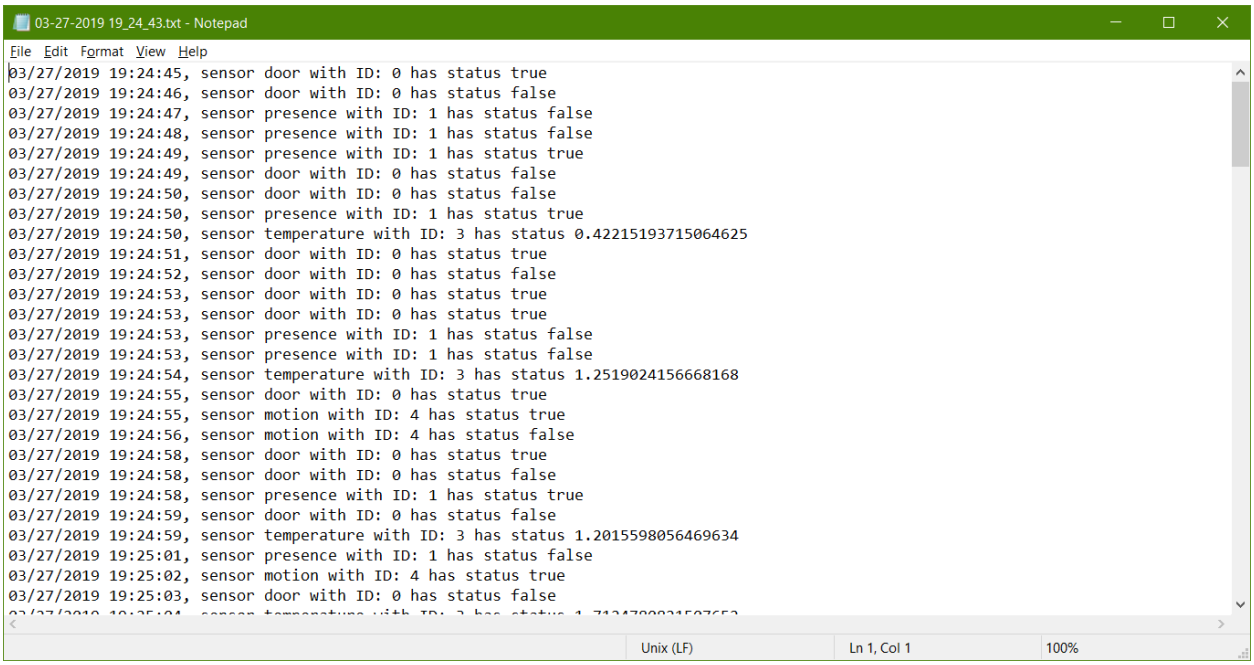
Test Case 16:

Input: Temperature sensor recorded temperature of 1.2015598056469634 degrees Celsius.

Pre-requisites: The server and temperature sensor are running.

Expected result: The server stores that information into a text file titled the time and date the sensor recorded at. It is stored as “x time, sensor temperature with ID: x has status 1.2015598056469634 degrees”.

Actual result: The server stores that information into a text file titled the time and date the sensor recorded at. It is stored as “x time, sensor temperature with ID: x has status 1.2015598056469634 degrees”.



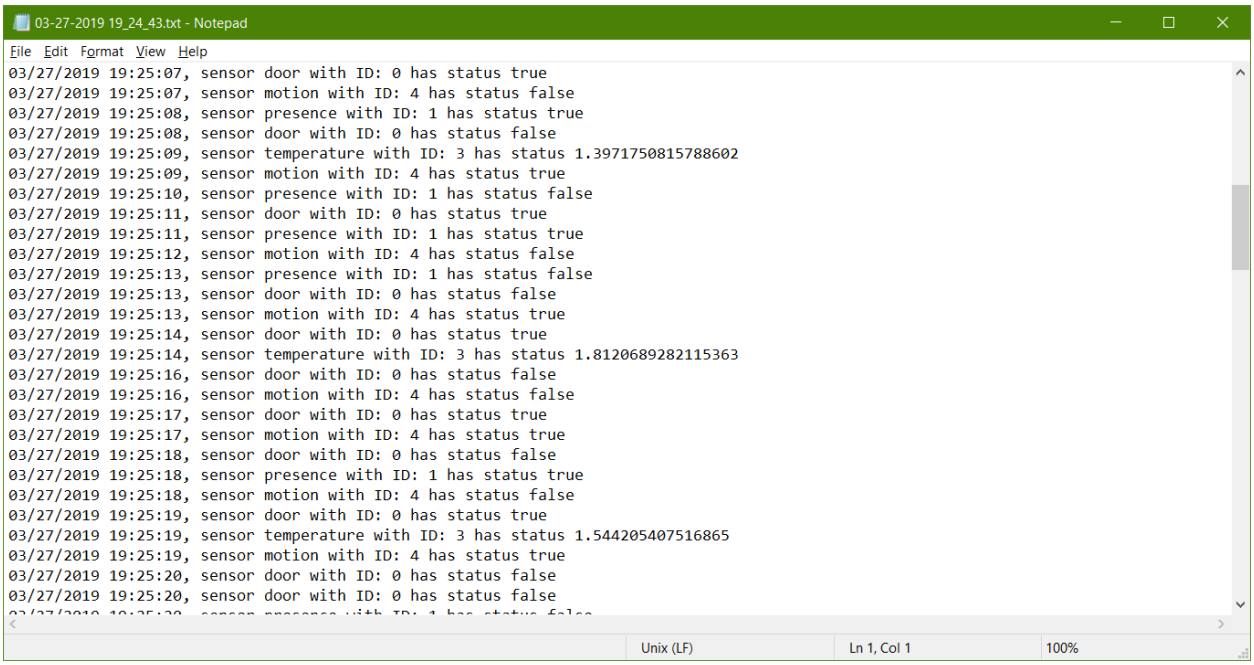
Test Case 17:

Input: The beacon is present.

Pre-requisites: The server and beacon are running and beacon is connected to it.

Expected result: The server stores that information into a text file titled the time and date the sensor recorded at. It is stored as “x time, sensor presence with ID: x has status true”.

Actual result: The server stores that information into a text file titled the time and date the sensor recorded at. It is stored as “x time, sensor presence with ID: x has status true”.



Test Case 18:

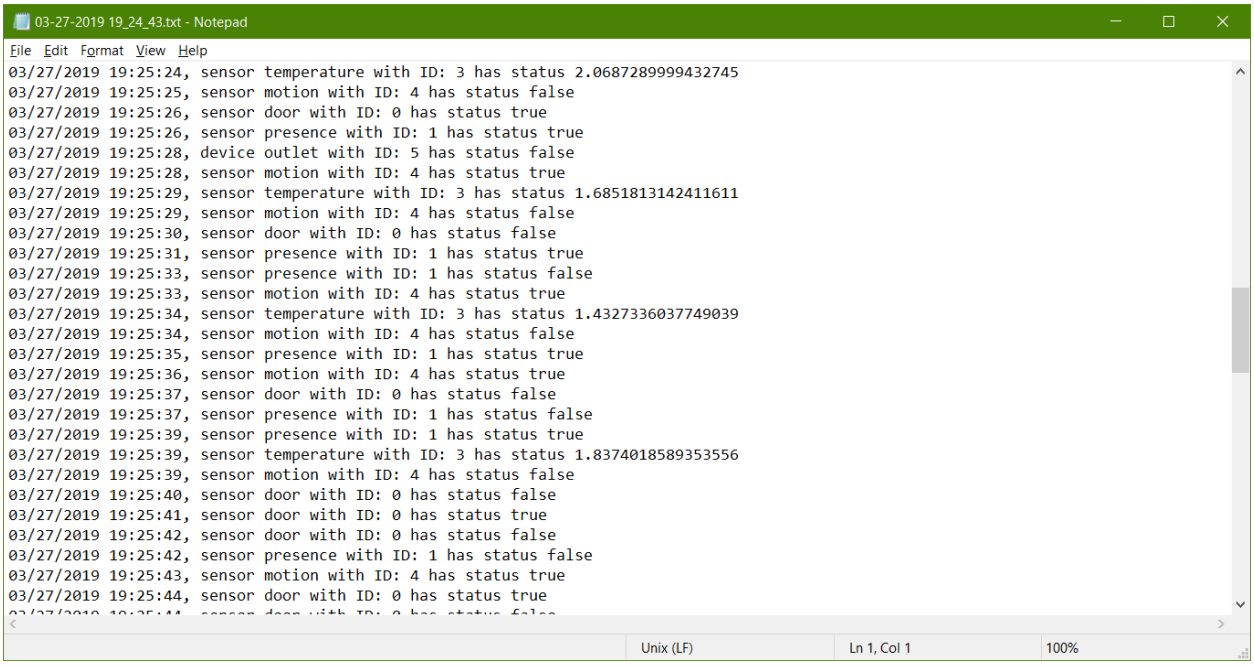
Input: Motion was sensed in the room.

Pre-requisites: The server and motion sensor are running and motion sensor is connected to it.

Expected result: The server stores that information into a text file titled the time and date the sensor recorded at. It is stored as “x time, sensor motion with ID: x is true”.

Actual result: The server stores that information into a text file titled the time and date the sensor

recorded at. It is stored as “x time, sensor motion with ID: x is true”.



Test Case 19:

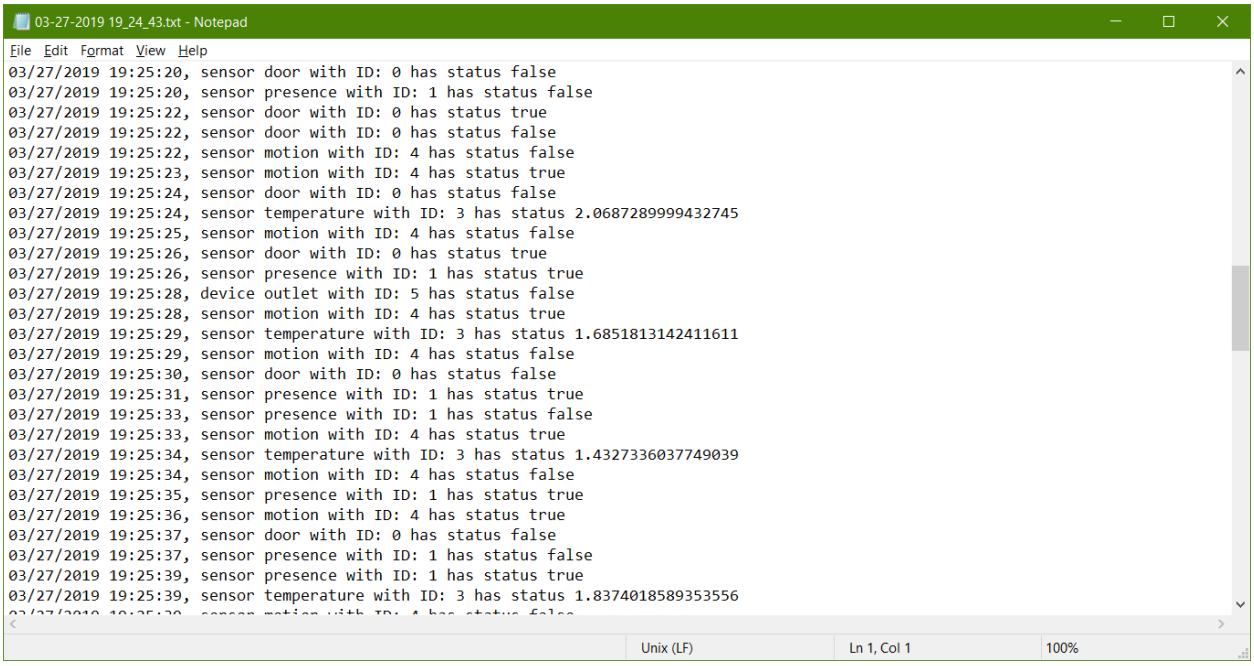
Input: outlet was turned off.

Pre-requisites: The server and outlet device are running and temperature sensor is connected to it.

Expected result: The server stores that information into a text file titled the time and date the sensor recorded at. It is stored as “x time, device outlet with ID: x has status false”.

Actual result: The server stores that information into a text file titled the time and date the sensor

recorded at. It is stored as “x time, device outlet with ID: x has status false”.



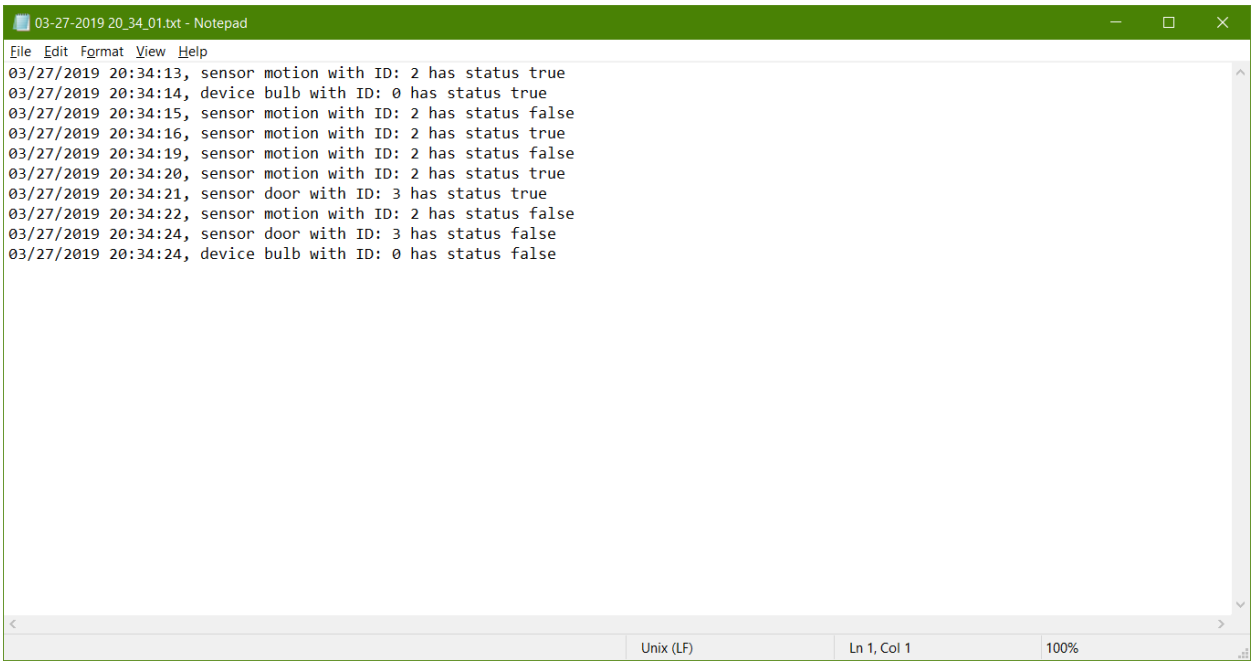
Test Case 20:

Input: The smart bulb turns on.

Pre-requisites: The server and smart bulb are running and smart bulb is connected to it.

Expected result: The server stores that information into a text file titled the time and date the data was recorded at. It is stored as “device bulb with ID: x has status true”.

Actual result: The server stores that information into a text file titled the time and date the data was recorded at. It is stored as “device bulb with ID: x has status true”.



Test Case 21:

Input: Database server quits from Gateway server.

Pre-requisites: The Gateway server and Database server are running.

Expected result: The database server will continue writing to a file until all data has been processed.

Actual result: The database server will continue writing to a file until all data has been processed.

```
Kevins-MacBook-Pro-2:HW2OS kingkev$ javac DatabaseServer.java && java DatabaseServer
Successfully connected to Gateway Server
Disconnected from Gateway Server
Finished Writing to files
Kevins-MacBook-Pro-2:HW2OS kingkev$
```

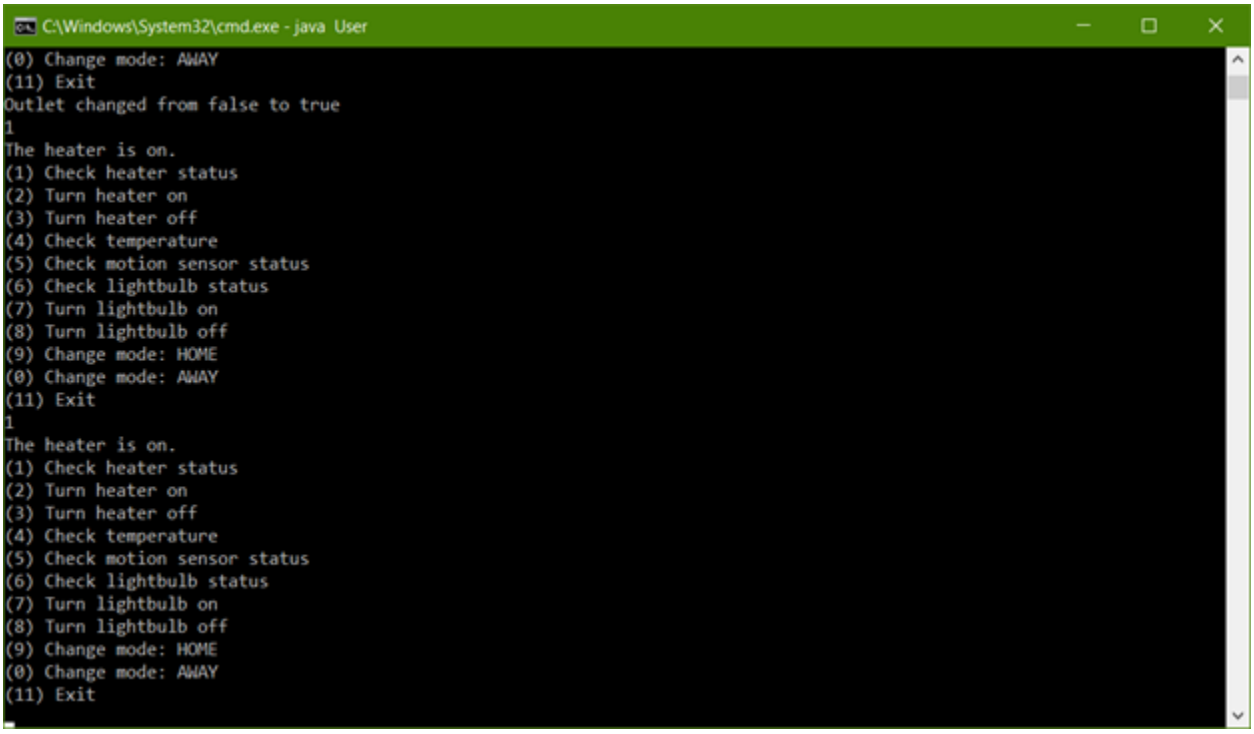
Test Case 22:

Input: User chooses option 1, “Check heater status”.

Pre-requisites: The server is running, User.java is connected to it, and the temperature is less than 1 degree Celsius.

Expected result: User receives a text message saying, “The heater is on”.

Actual result: User receives a text message saying, “The heater is on”.



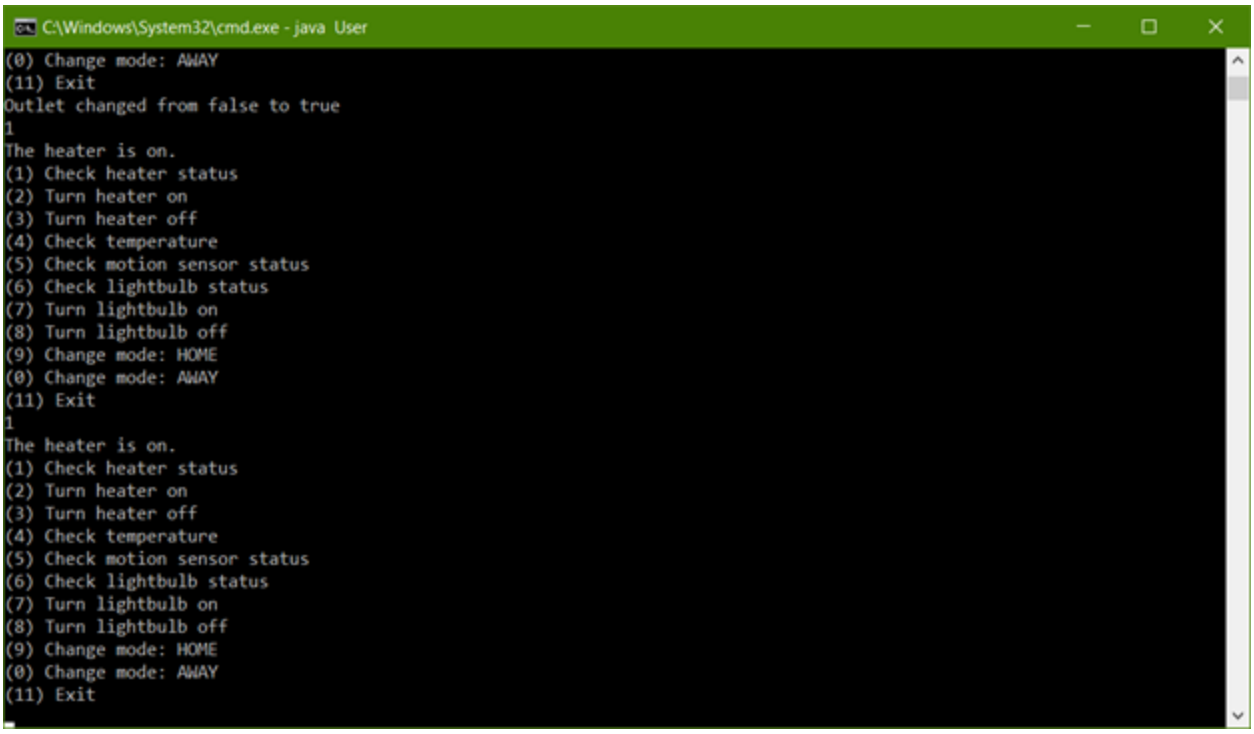
Test Case 23:

Input: User chooses option 1, “Check heater status”.

Pre-requisites: The server is running, User.java is connected to it, and the temperature is between 1 degree Celsius and 2 degrees Celsius.

Expected result: User receives a text message saying, “The heater is on”.

Actual result: User receives a text message saying, “The heater is on”.



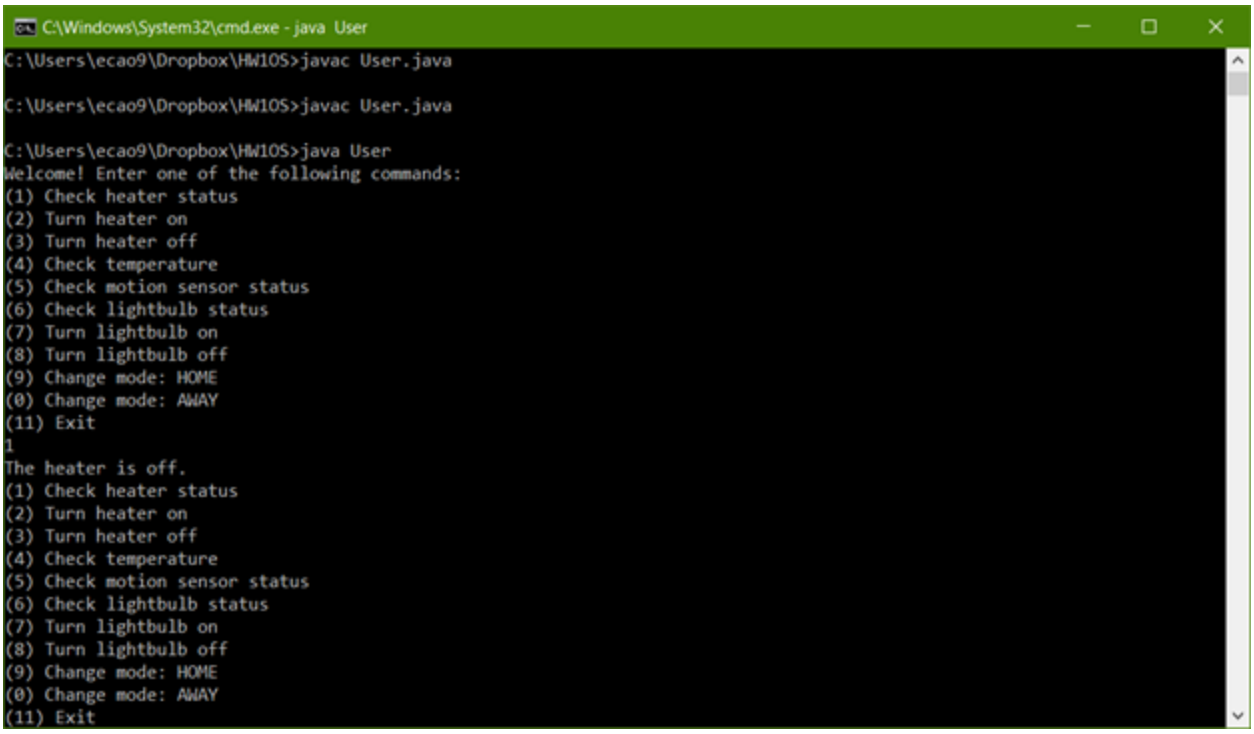
Test Case 24:

Input: User chooses option 1, “Check heater status”.

Pre-requisites: The server is running, User.java is connected to it, and the temperature is greater than 2 degrees Celsius.

Expected result: User receives a text message saying, “The heater is off”.

Actual result: User receives a text message saying, “The heater is off”.



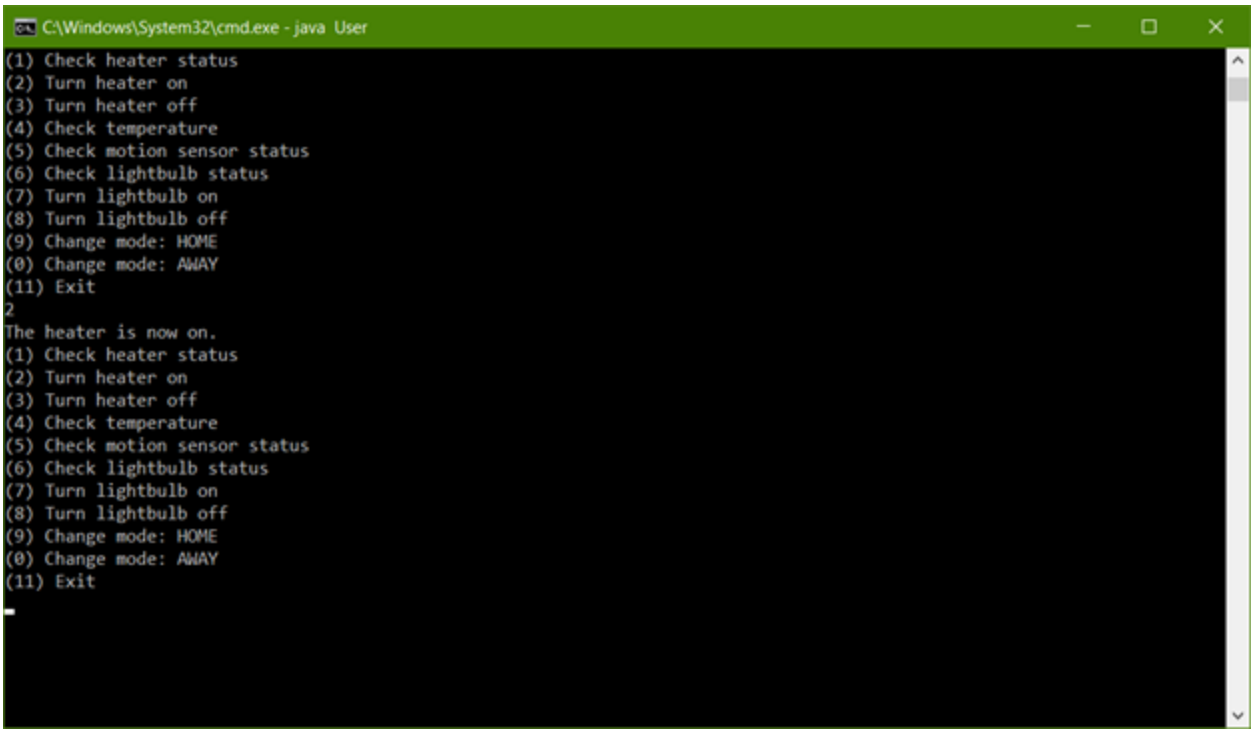
Test Case 25:

Input: User chooses option 2, “Turn heater on”.

Pre-requisites: The server is running, User.java is connected to it, and the heater is off.

Expected result: The heater is turned on and User receives a text message saying, “the heater is now on”.

Actual result: The heater is turned on and User receives a text message saying, “the heater is now on”.



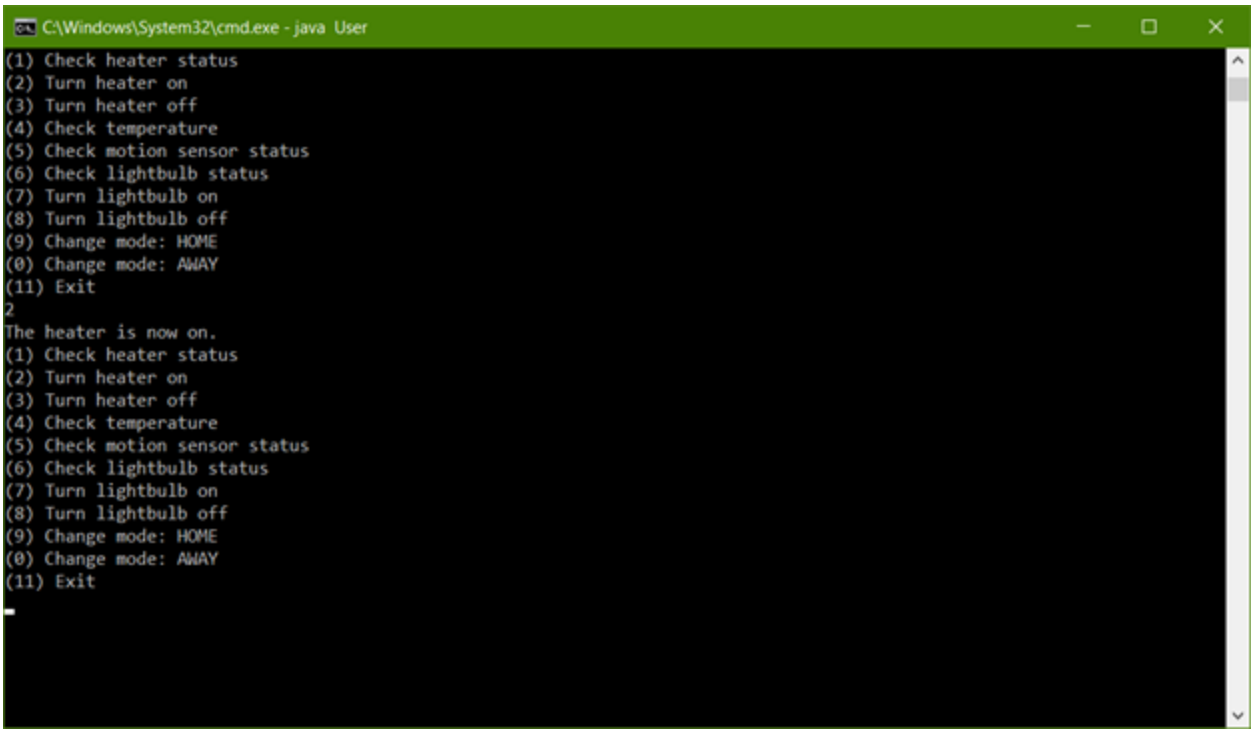
Test Case 26:

Input: User chooses option 2, “Turn heater on”.

Pre-requisites: The server is running, User.java is connected to it, and the heater is on.

Expected result: The heater is turned on and User receives a text message saying, “the heater is now on”.

Actual result: The heater is turned on and User receives a text message saying, “the heater is now on”.



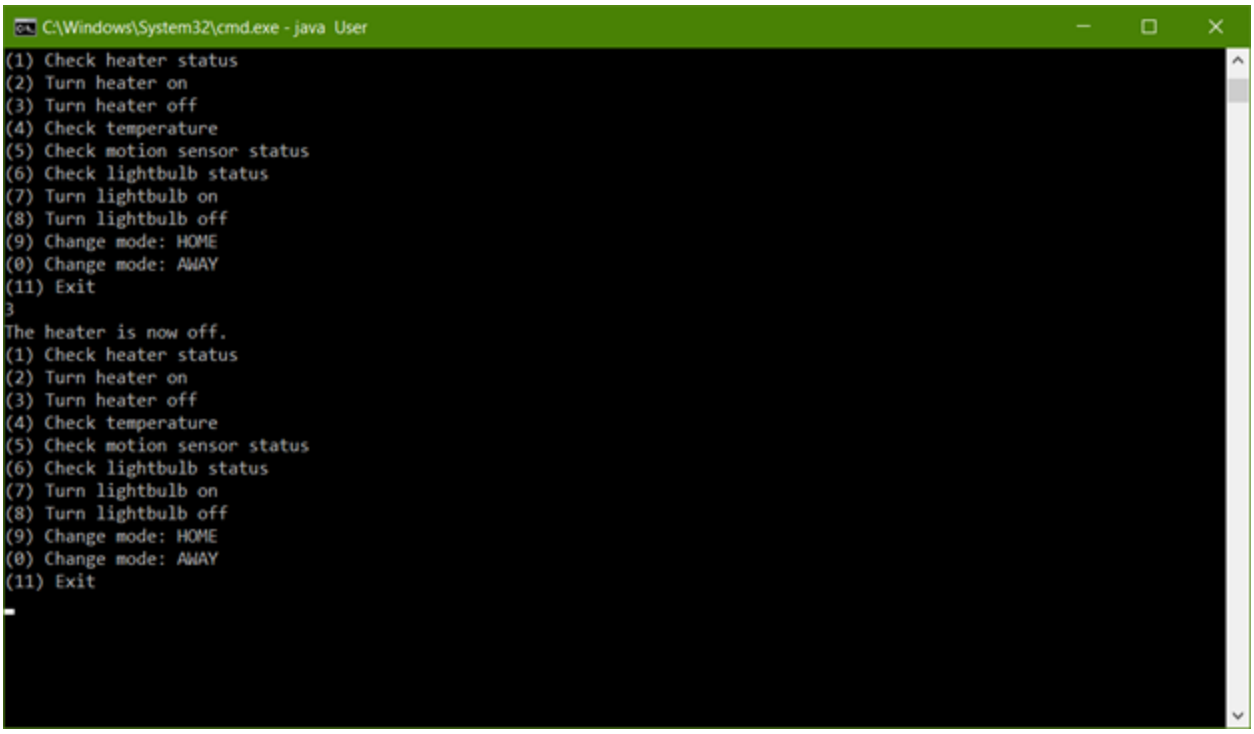
Test Case 27:

Input: User chooses option 3, “Turn heater off”.

Pre-requisites: The server is running, User.java is connected to it, and the heater is on.

Expected result: The heater is turned off and User receives a text message saying, “The heater is now off”.

Actual result: The heater is turned off and User receives a text message saying, “The heater is now off”.



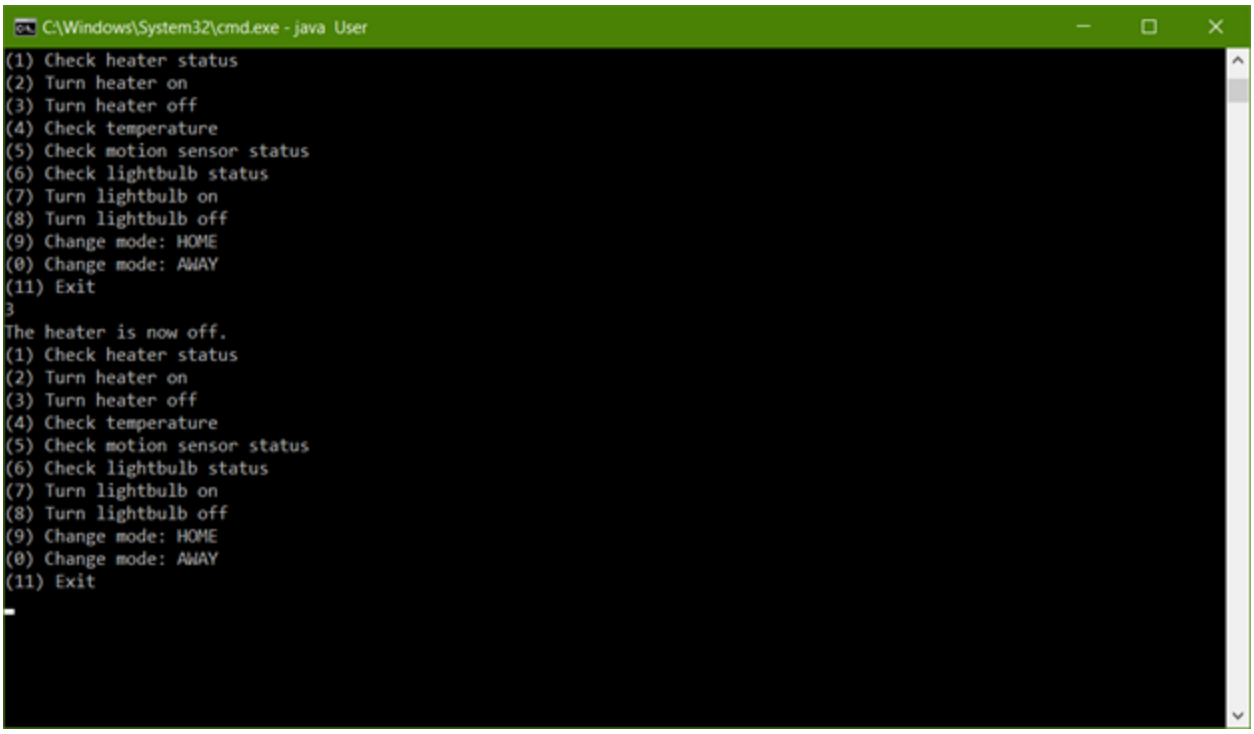
Test Case 28:

Input: User chooses option 3, “Turn heater off”.

Pre-requisites: The server is running, User.java is connected to it, and the heater is off.

Expected result: The heater is turned off and User receives a text message saying, “The heater is now off”.

Actual result: The heater is turned off and User receives a text message saying, “The heater is now off”.



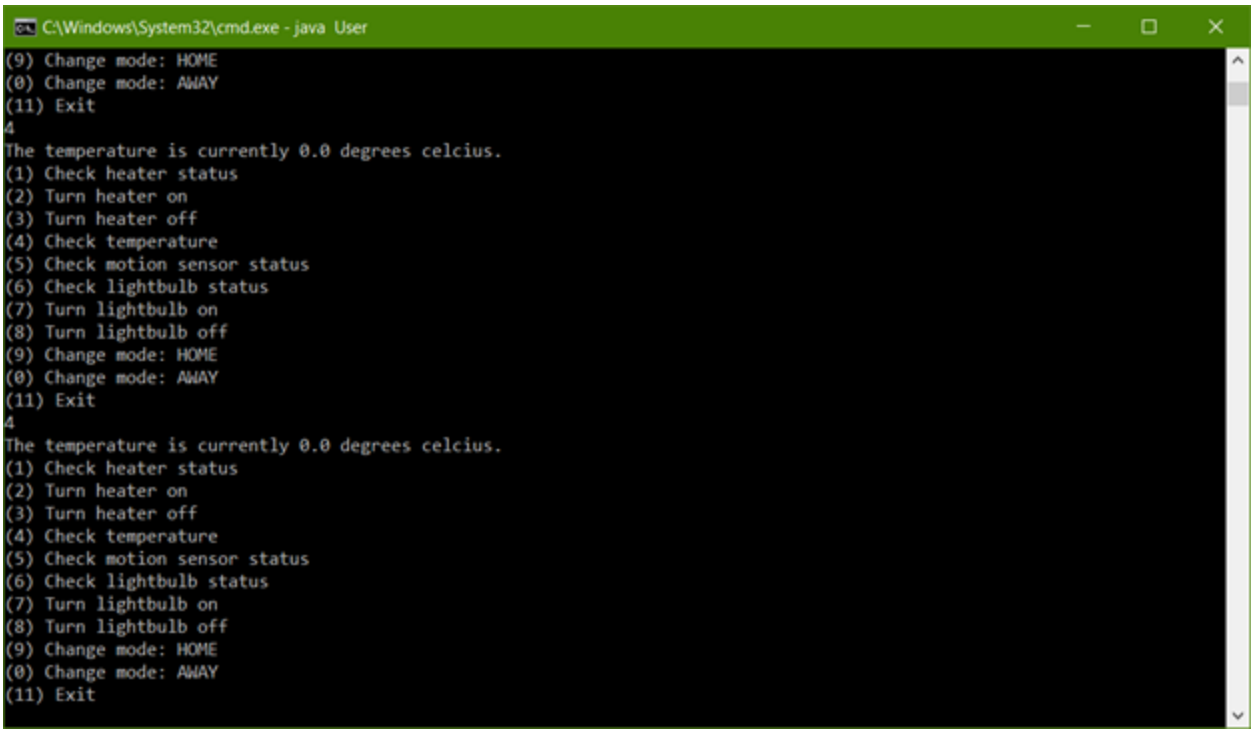
Test Case 29:

Input: User chooses option 4, “Check temperature”.

Pre-requisites: The server is running, User.java is connected to it, and it is 0 degrees Celsius.

Expected result: User receives a text message saying, “The temperature is currently 0.0 degrees Celsius”.

Actual result: User receives a text message saying, “The temperature is currently 0.0 degrees Celsius”.



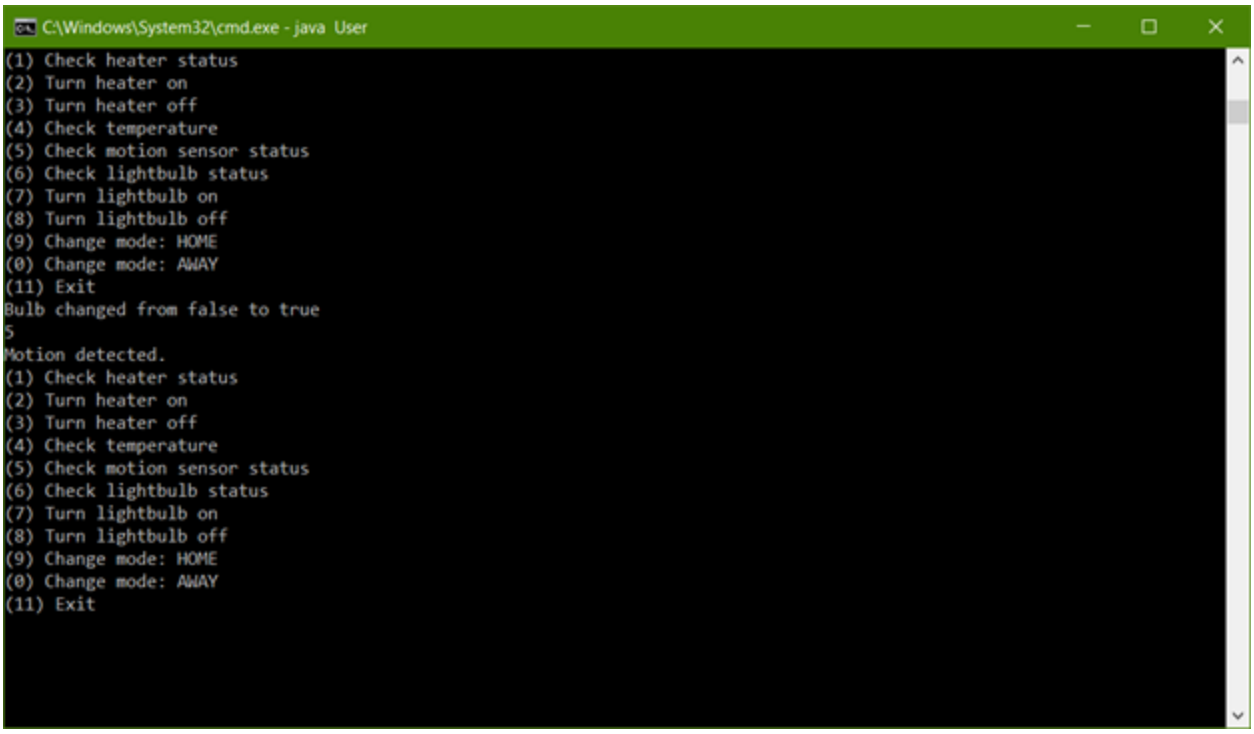
Test Case 30:

Input: User chooses option 5, “Check motion sensor status”.

Pre-requisites: The server is running, User.java is connected to it, and there is motion.

Expected result: User receives a text message saying, “Motion detected”.

Actual result: User receives a text message saying, “Motion detected”.



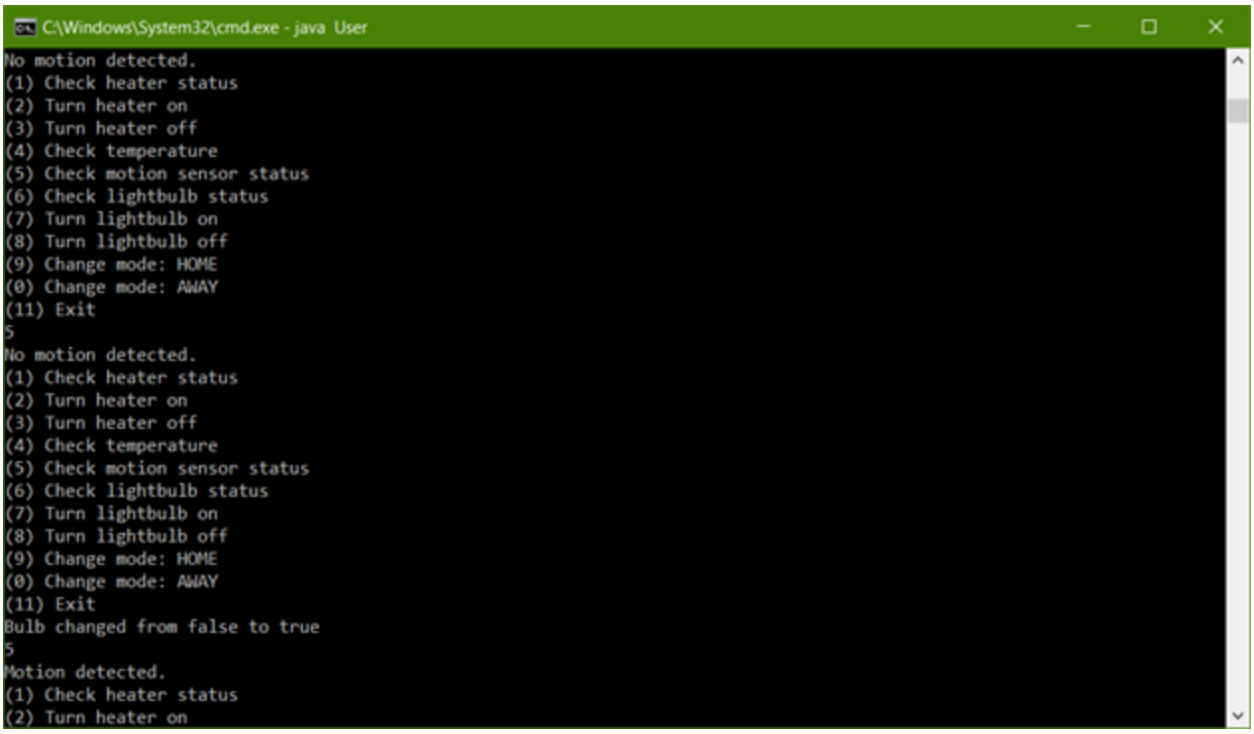
Test Case 31:

Input: User chooses option 5, “Check motion sensor status”.

Pre-requisites: The server is running, User.java is connected to it, and there is no motion.

Expected result: User receives a text message saying, “No motion detected”.

Actual result: User receives a text message saying, “No motion detected”.



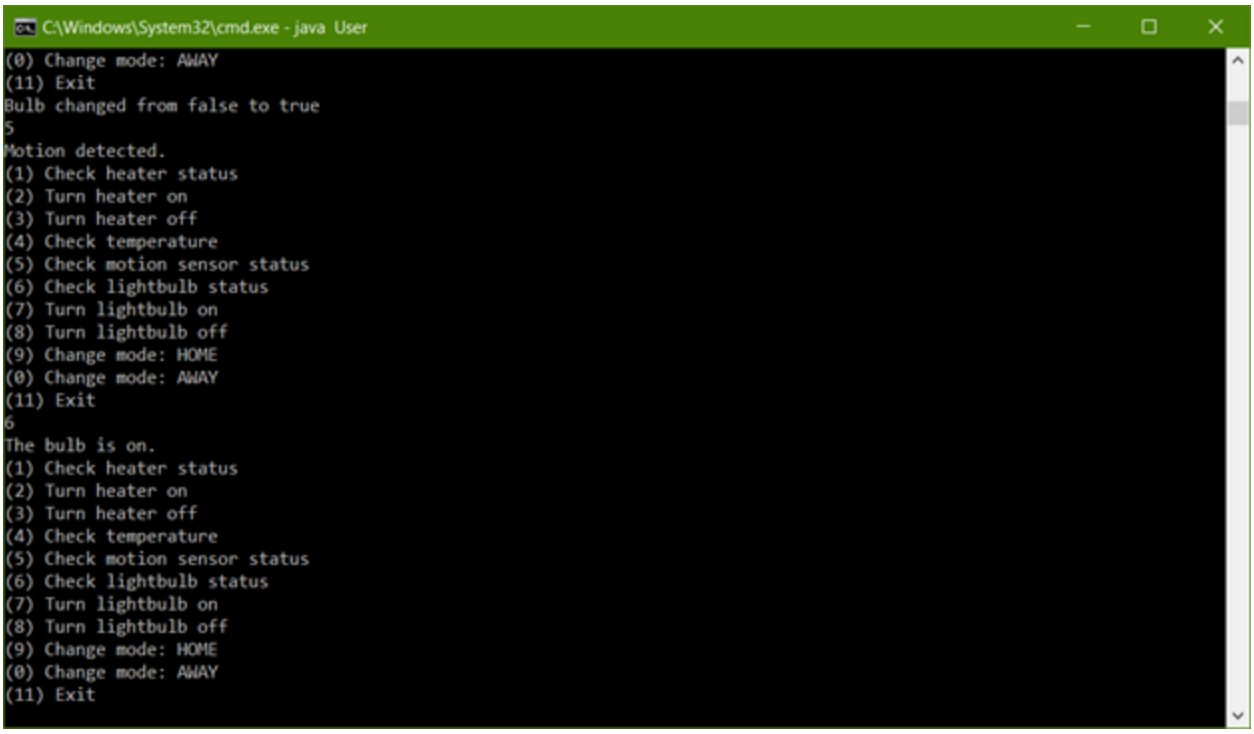
Test Case 32:

Input: User chooses option 6, “Check lightbulb status”.

Pre-requisites: The server is running, User.java is connected to it, and there is motion.

Expected result: User receives a text message saying, “The bulb is on”.

Actual result: User receives a text message saying, “The bulb is on”.



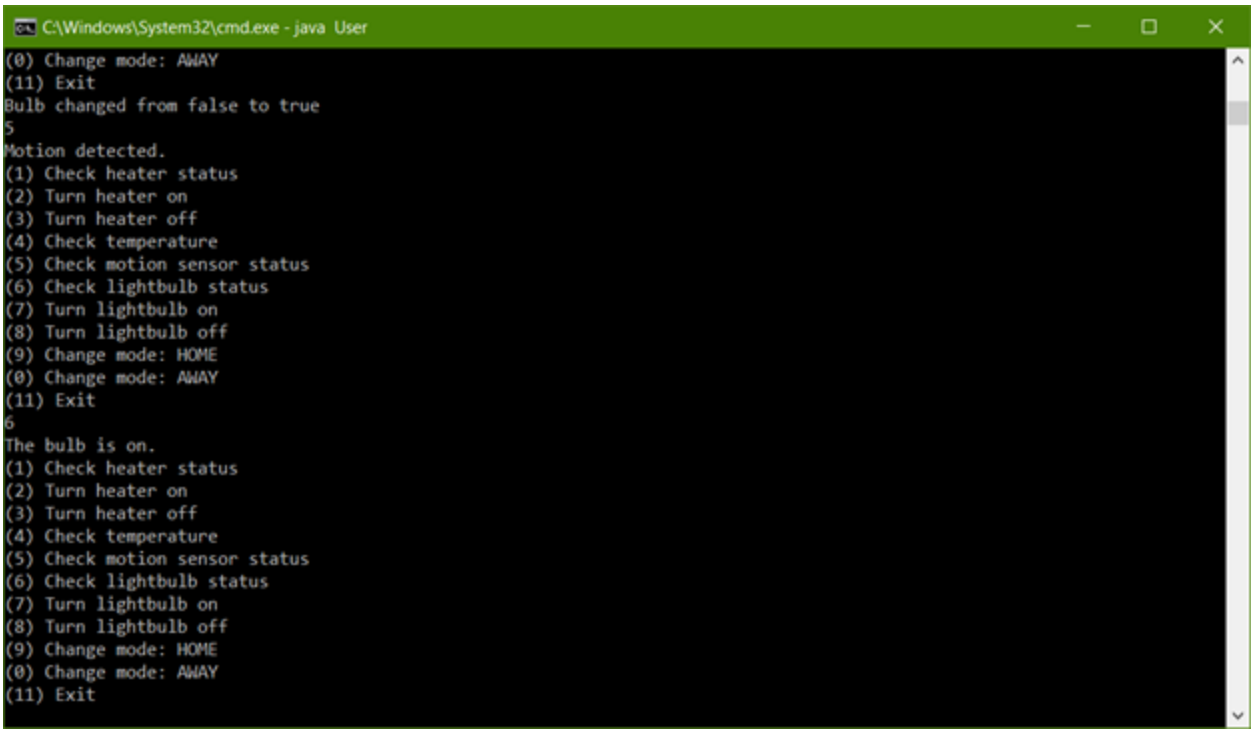
Test Case 33:

Input: User chooses option 6, “Check lightbulb status”.

Pre-requisites: The server is running, User.java is connected to it, and there has been no motion for three minutes.

Expected result: User receives a text message saying, “Bulb is on”.

Actual result: User receives a text message saying, “Bulb is on”.



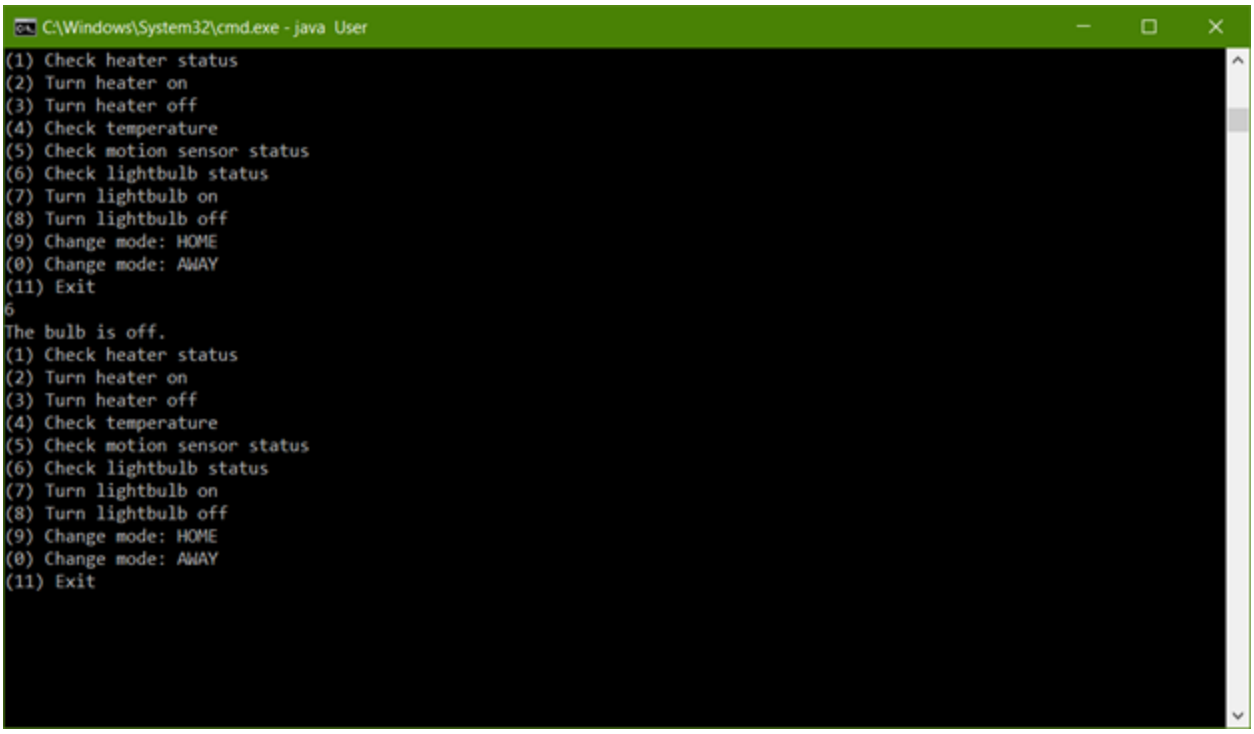
Test Case 34:

Input: User chooses option 6, “Check lightbulb status”.

Pre-requisites: The server is running, User.java is connected to it, and there has been no motion for six minutes.

Expected result: User receives a text message saying, “Bulb is off”.

Actual result: User receives a text message saying, “Bulb is off”.



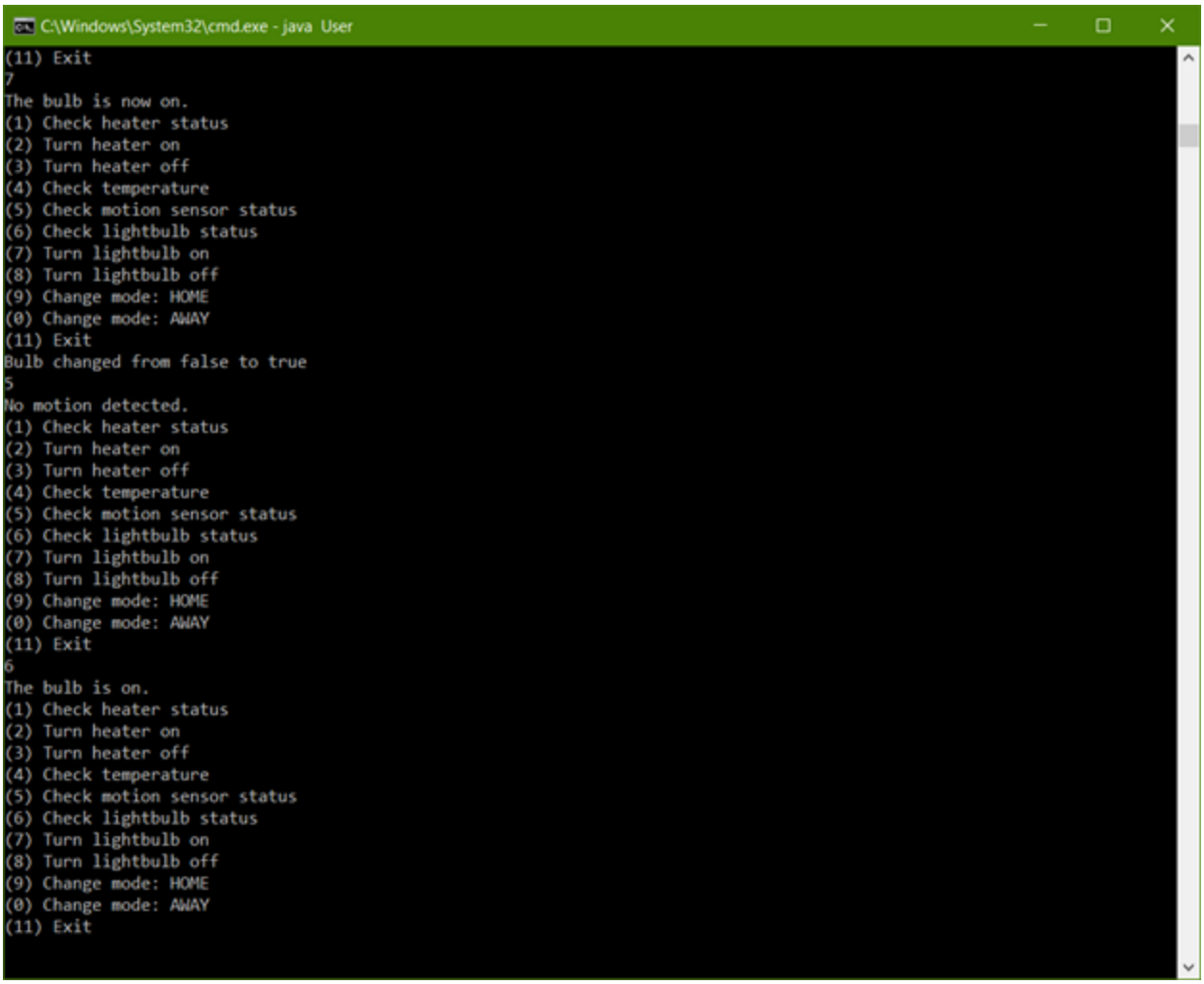
Test Case 35:

Input: User chooses option 9, “Turn lightbulb on”.

Pre-requisites: The server is running, User.java is connected to it, and there has been no motion for five minutes.

Expected result: User receives a text message saying, “The bulb is now on”.

Actual result: User receives a text message saying, “The bulb is now on”.



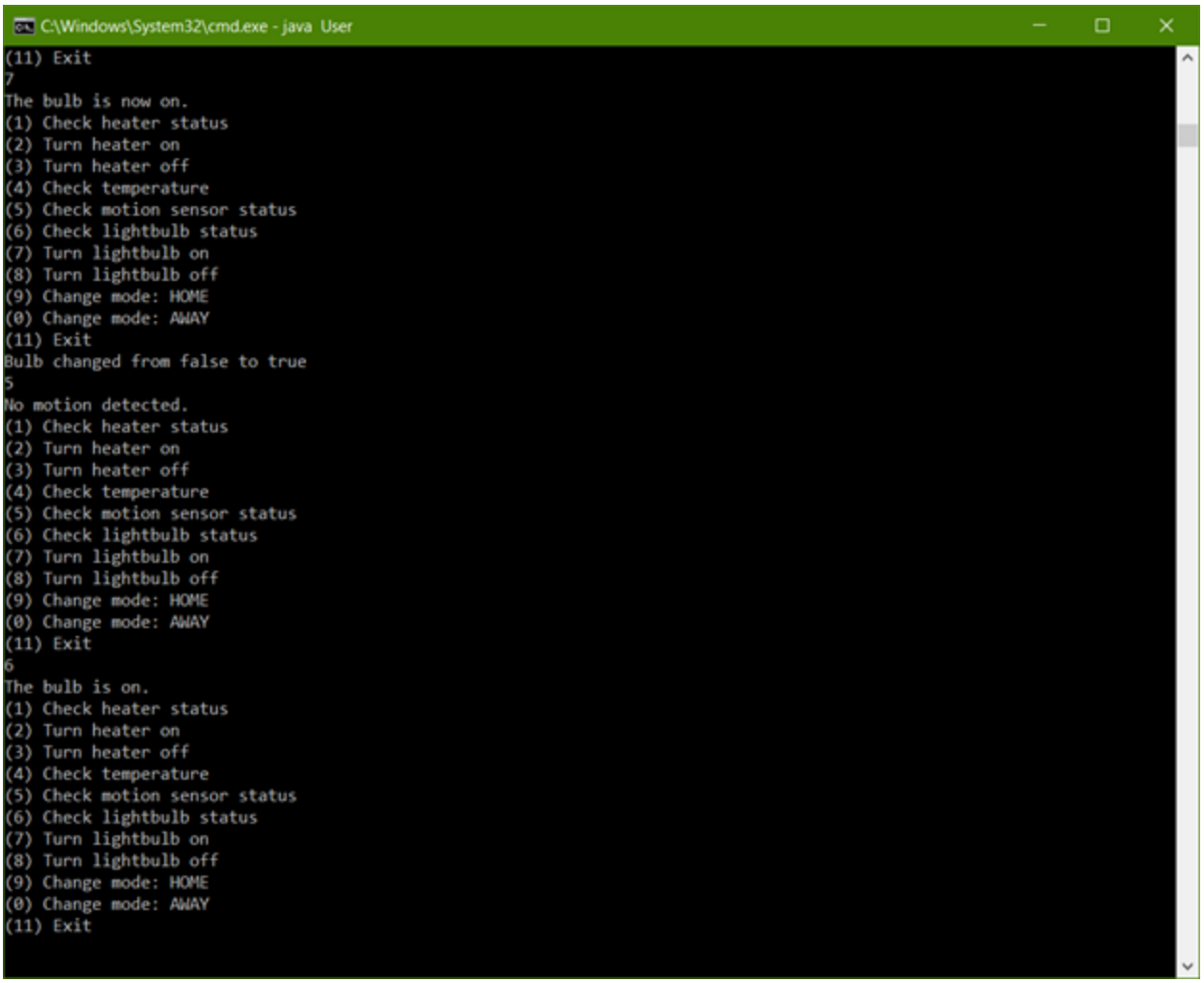
Test Case 36:

Input: User chooses option 9, “Turn lightbulb on”.

Pre-requisites: The server is running, User.java is connected to it, and there has been no motion for three minutes.

Expected result: User receives a text message saying, “The bulb is now on”.

Actual result: User receives a text message saying, “The bulb is now on”.



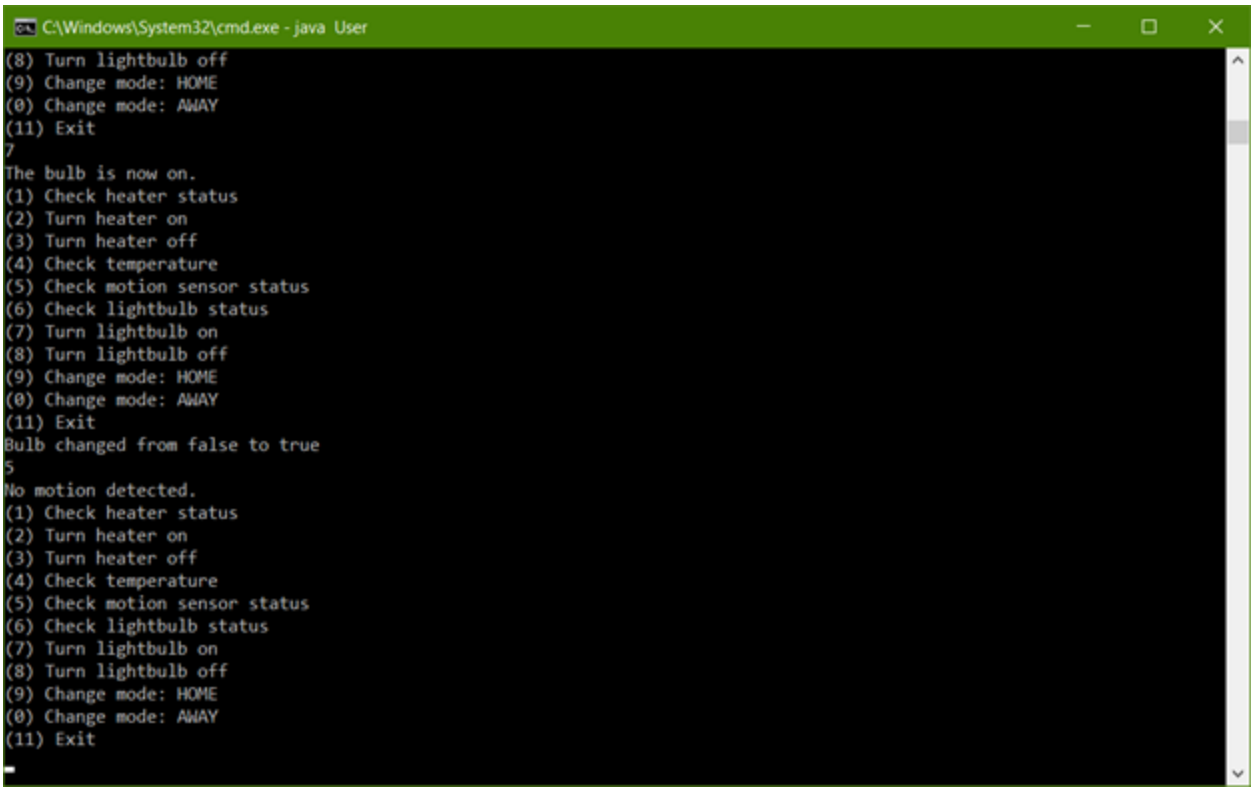
Test Case 37:

Input: User chooses option 9, “Turn lightbulb on”.

Pre-requisites: The server is running, User.java is connected to it, and the lightbulb is already on.

Expected result: User receives a text message saying, “The bulb is now on”.

Actual result: User receives a text message saying, “The bulb is now on”.



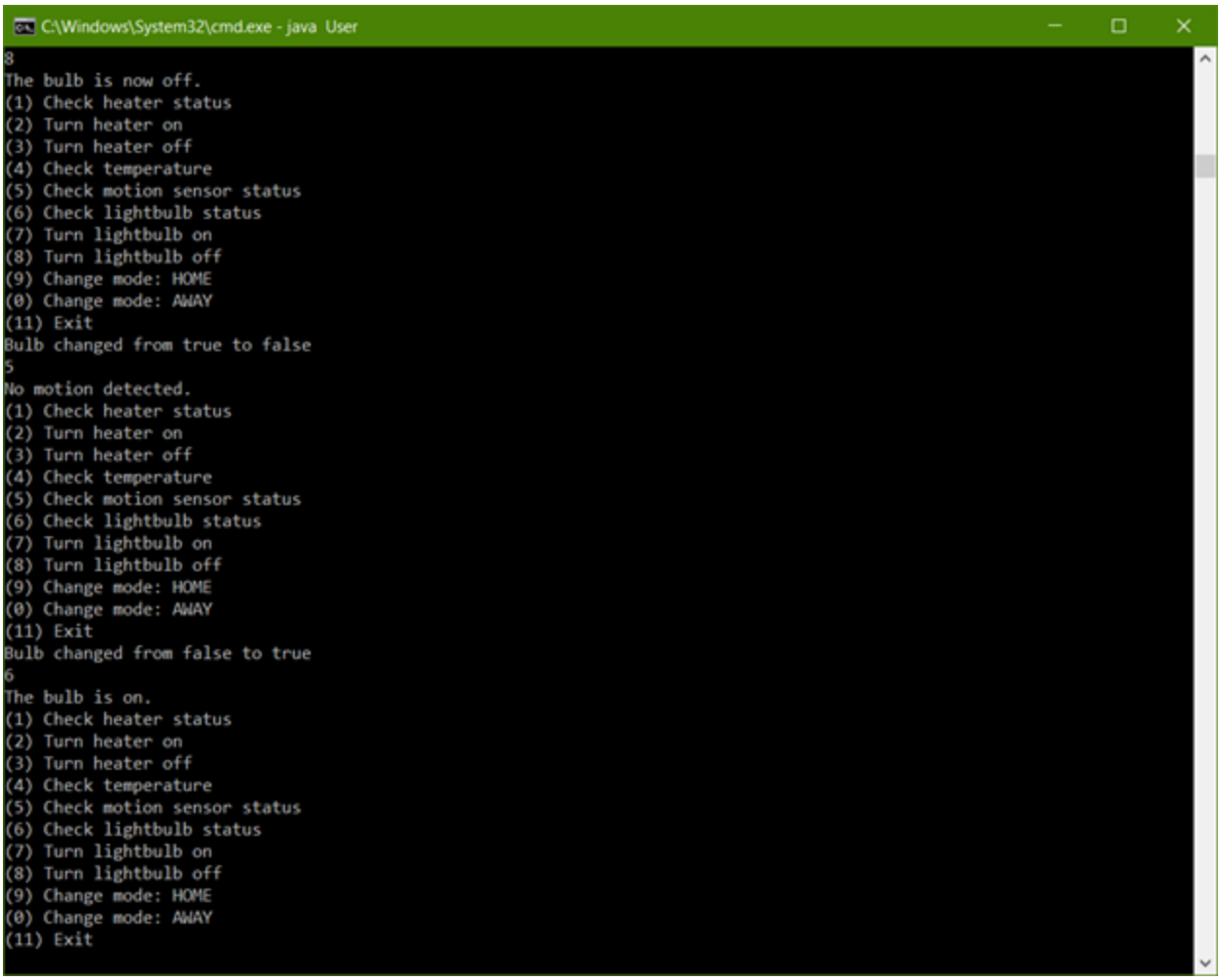
Test Case 38:

Input: User chooses option 10, “Turn lightbulb off”.

Pre-requisites: The server is running, User.java is connected to it, and there has been no motion for three minutes.

Expected result: User receives a text message saying, “The bulb is now on”.

Actual result: User receives a text message saying, “The bulb is now on”.



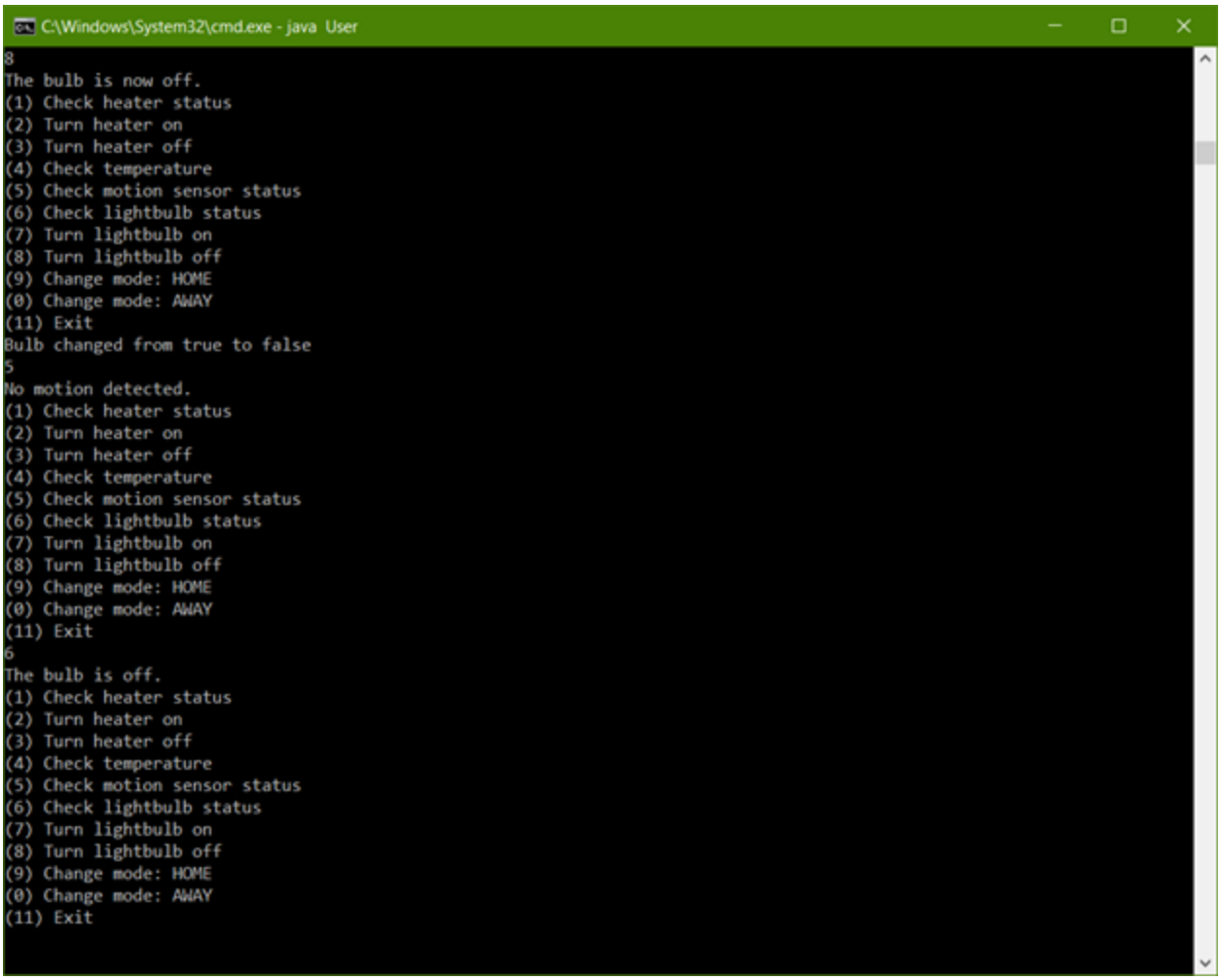
Test Case 39:

Input: User chooses option 10, “Turn lightbulb off”.

Pre-requisites: The server is running, User.java is connected to it, and there has been no motion for five minutes.

Expected result: User receives a text message saying, “The bulb is now off”.

Actual result: User receives a text message saying, “The bulb is now off”.



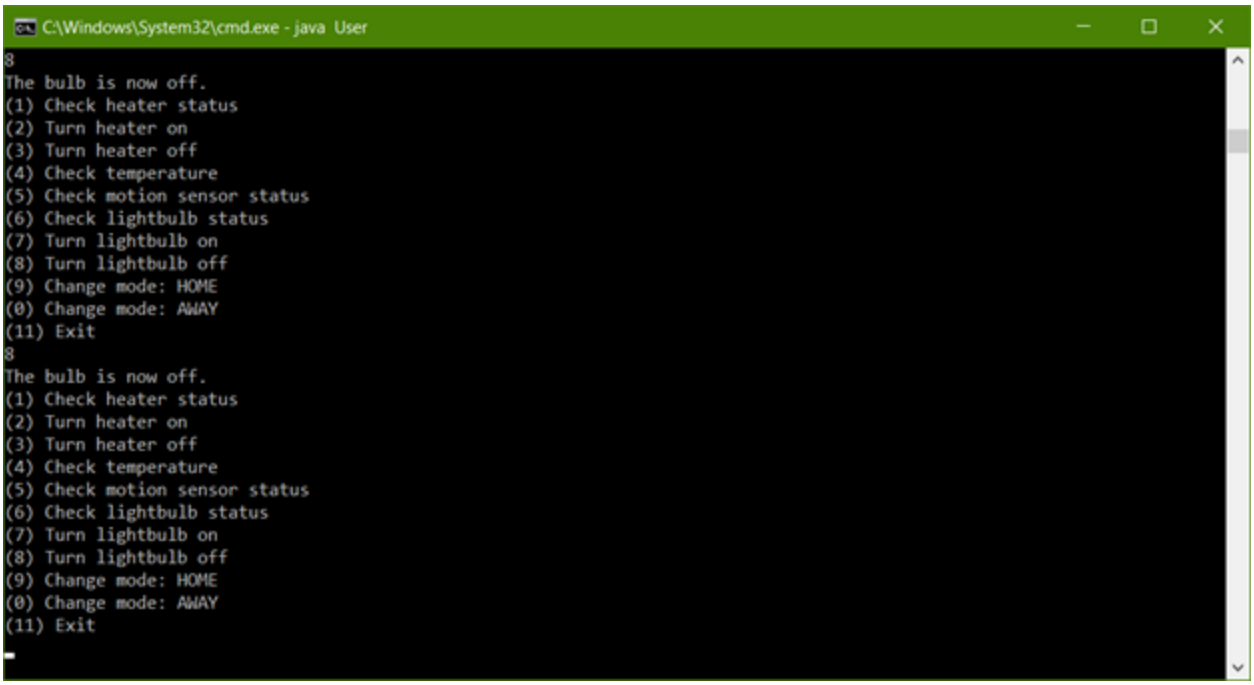
Test Case 40:

Input: User chooses option 10, “Turn lightbulb off”.

Pre-requisites: The server is running, User.java is connected to it, and the lightbulb is already off.

Expected result: User receives a text message saying, “The bulb is now off”.

Actual result: User receives a text message saying, “The bulb is now off”.



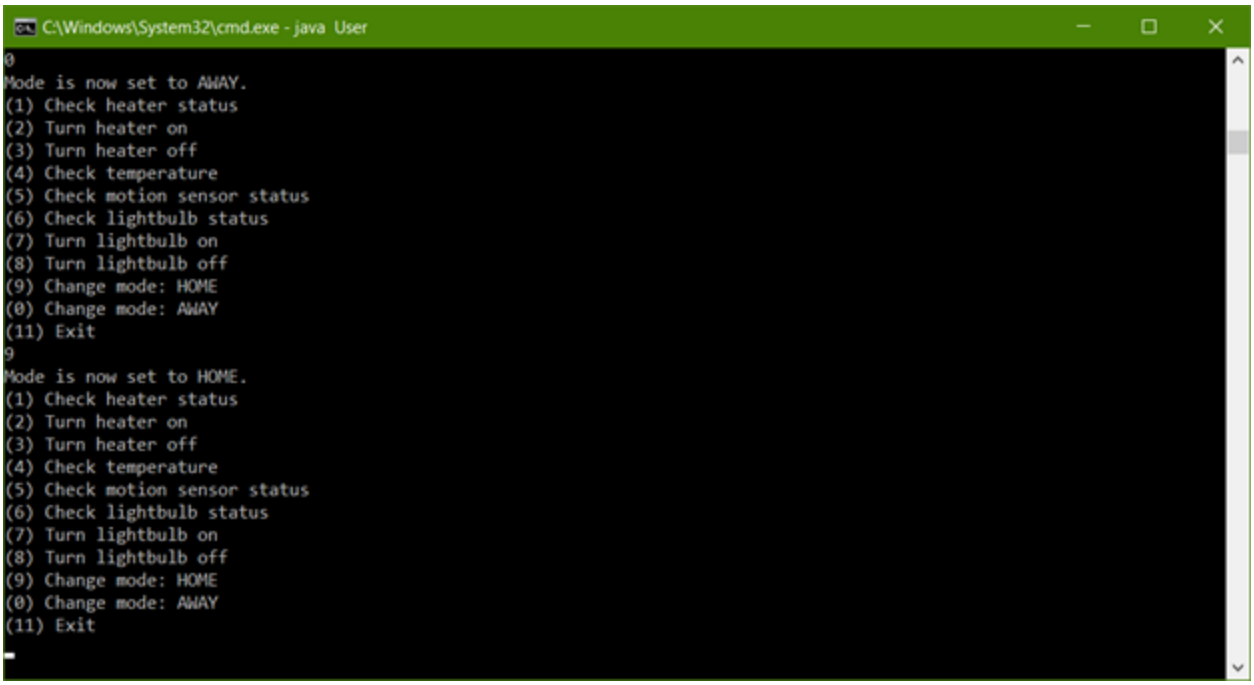
Test Case 41:

Input: User chooses option 11, “Change mode: HOME”.

Pre-requisites: The server is running, User.java is connected to it, and mode is currently set to AWAY.

Expected result: User receives a text message saying, “Mode is now set to HOME”.

Actual result: User receives a text message saying, “Mode is now set to HOME”.



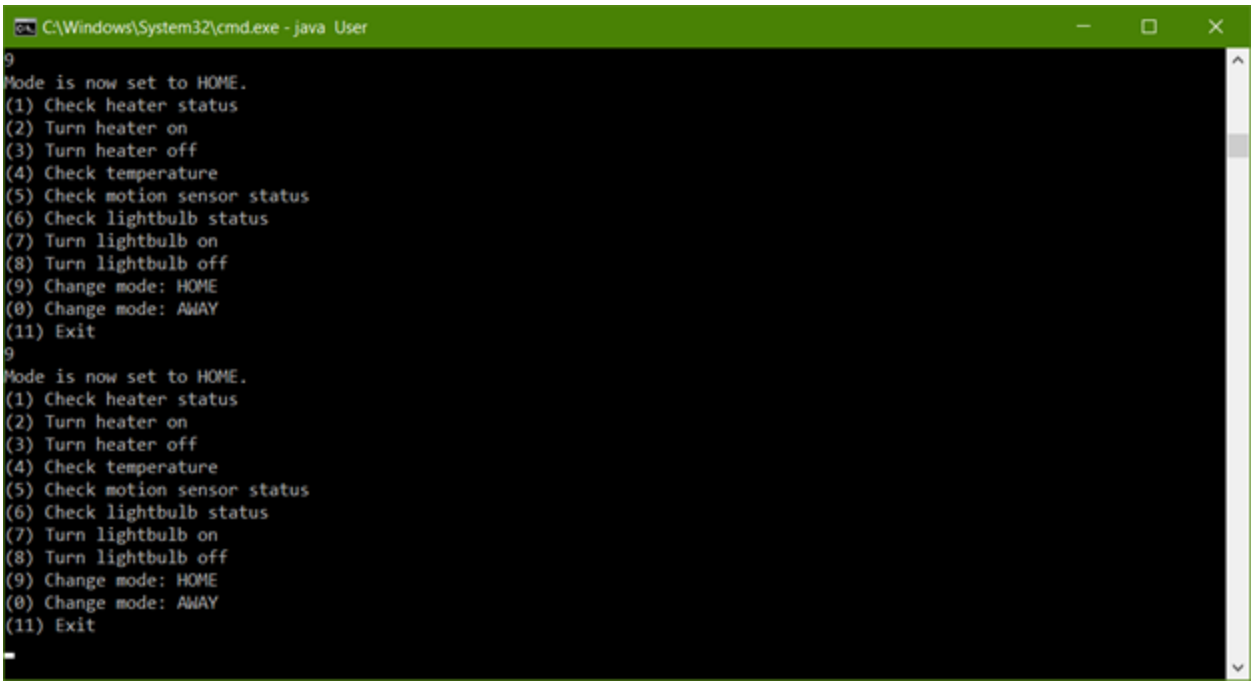
Test Case 42:

Input: User chooses option 11, “Change mode: HOME”.

Pre-requisites: The server is running, User.java is connected to it, and mode is currently set to HOME.

Expected result: User receives a text message saying, “Mode is now set to HOME”.

Actual result: User receives a text message saying, “Mode is now set to HOME”.



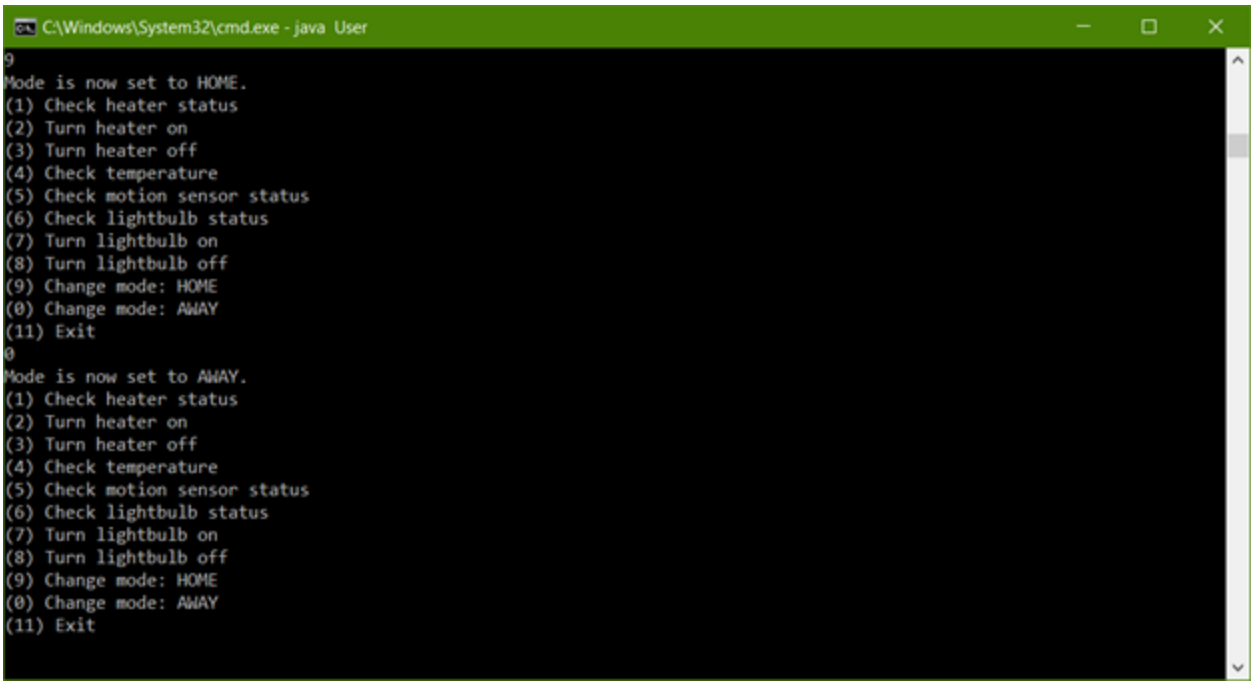
Test Case 43:

Input: User chooses option 12, “Change mode: AWAY”.

Pre-requisites: The server is running, User.java is connected to it, and mode is currently set to HOME.

Expected result: User receives a text message saying, “Mode is now set to AWAY”.

Actual result: User receives a text message saying, “Mode is now set to AWAY”.



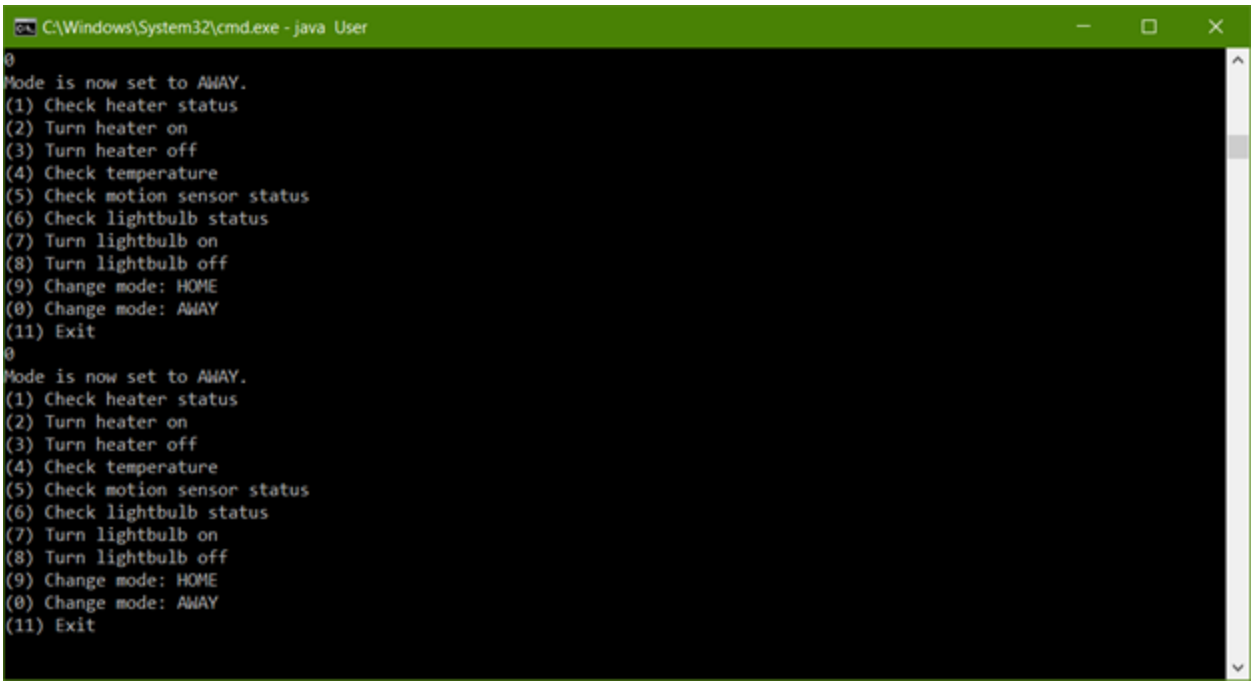
Test Case 44:

Input: User chooses option 12, “Change mode: AWAY”.

Pre-requisites: The server is running, User.java is connected to it, and mode is currently set to AWAY.

Expected result: User receives a text message saying, “Mode is now set to AWAY”.

Actual result: User receives a text message saying, “Mode is now set to AWAY”.



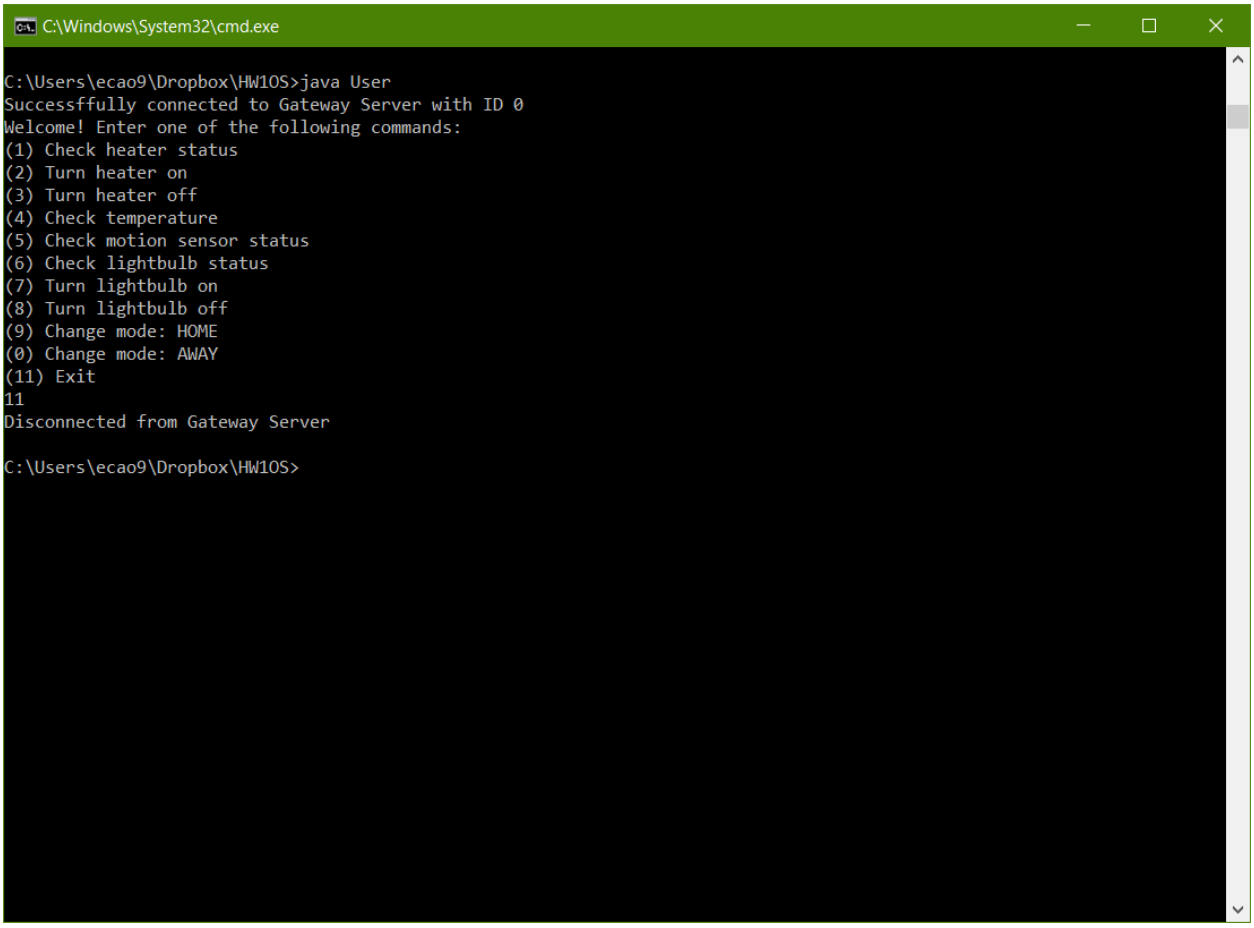
Test Case 45:

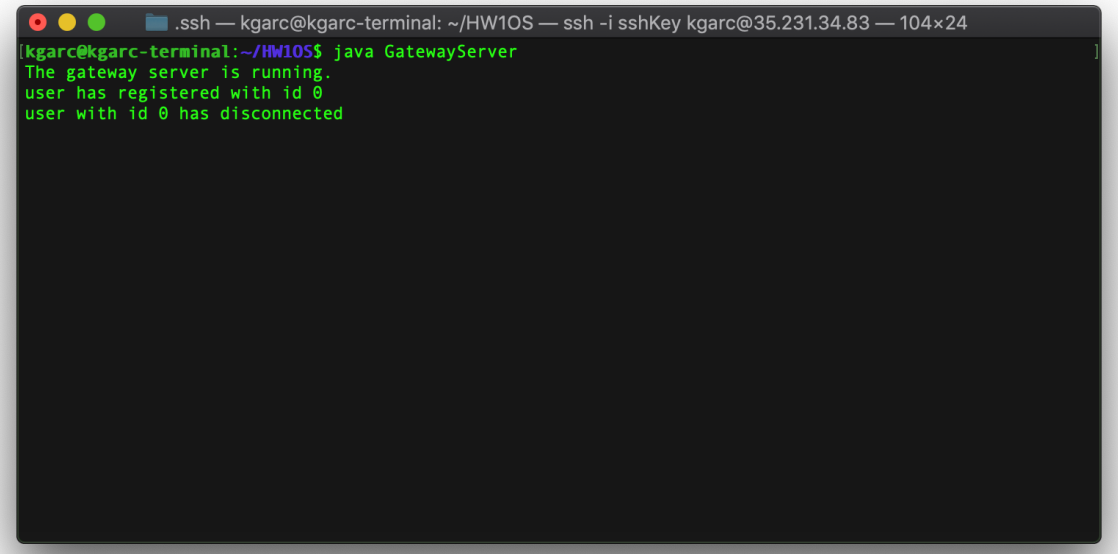
Input: User chooses option 14, “Exit”.

Pre-requisites: The server is running and User.java is connected to it.

Expected result: User.java successfully disconnects with message “Disconnected from Gateway Server”, and the server continues running and its prints the message “User with id x has disconnected” and stops the thread.

Actual result: User.java successfully disconnects with message “Disconnected from Gateway Server”, and the server continues running and its prints the message “User with id x has disconnected” and stops the thread.


C:\Windows\System32\cmd.exe
C:\Users\ecao9\Dropbox\HW1OS>java User
Successffully connected to Gateway Server with ID 0
Welcome! Enter one of the following commands:
(1) Check heater status
(2) Turn heater on
(3) Turn heater off
(4) Check temperature
(5) Check motion sensor status
(6) Check lightbulb status
(7) Turn lightbulb on
(8) Turn lightbulb off
(9) Change mode: HOME
(0) Change mode: AWAY
(11) Exit
11
Disconnected from Gateway Server
C:\Users\ecao9\Dropbox\HW1OS>


.ssh — kgarc@kgarc-terminal: ~/HW1OS — ssh -i sshKey kgarc@35.231.34.83 — 104×24
[kgarc@kgarc-terminal:~/HW1OS$ java GatewayServer
The gateway server is running.
user has registered with id 0
user with id 0 has disconnected

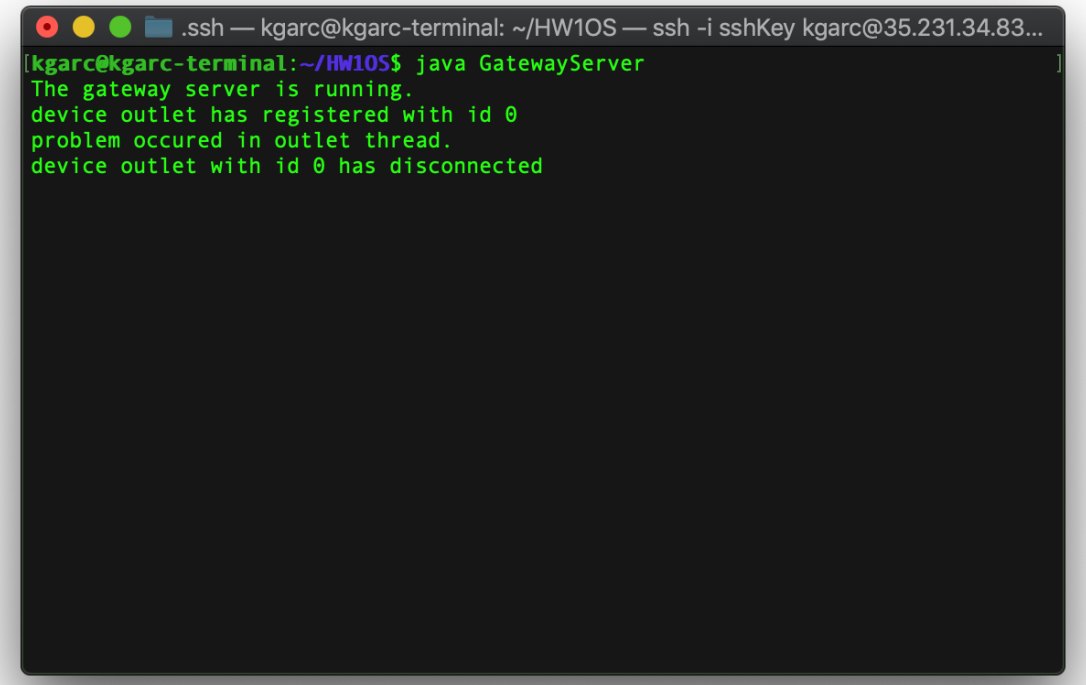
Test Case 46:

Input: Outlet stops running unexpectedly.

Pre-requisites: The server and the heater are running.

Expected result: Outlet successfully stops, the server keeps running with no error, and server prints out the message “Device outlet with id x has disconnected”, and that thread stops.

Actual result: Outlet successfully stops, the server keeps running with no error, and server prints out the message “Device outlet with id x has disconnected”, and that thread stops.



Test Case 47:

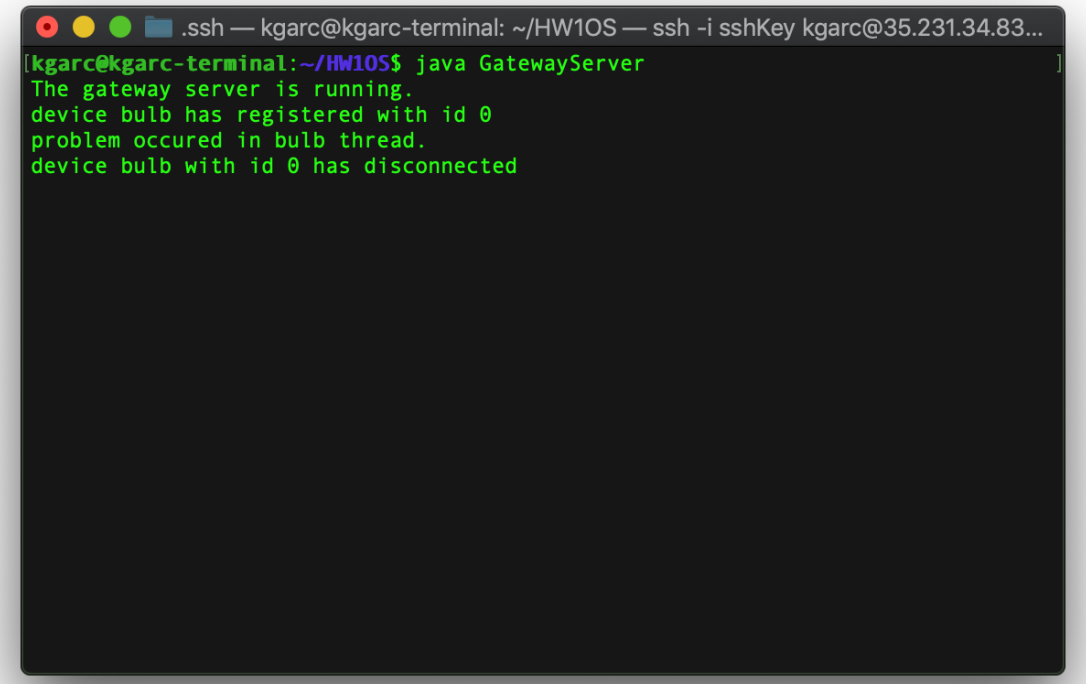
Input: SmartBulb stops running unexpectedly.

Pre-requisites: The server and the lightbulb are running.

Expected result: SmartBulb successfully stops, the server keeps running with no error, and server

prints out the message “Device bulb with id x has disconnected”, and that thread stops.

Actual result: SmartBulb successfully stops, the server keeps running with no error, and server prints out the message “Device bulb with id x has disconnected”, and that thread stops.



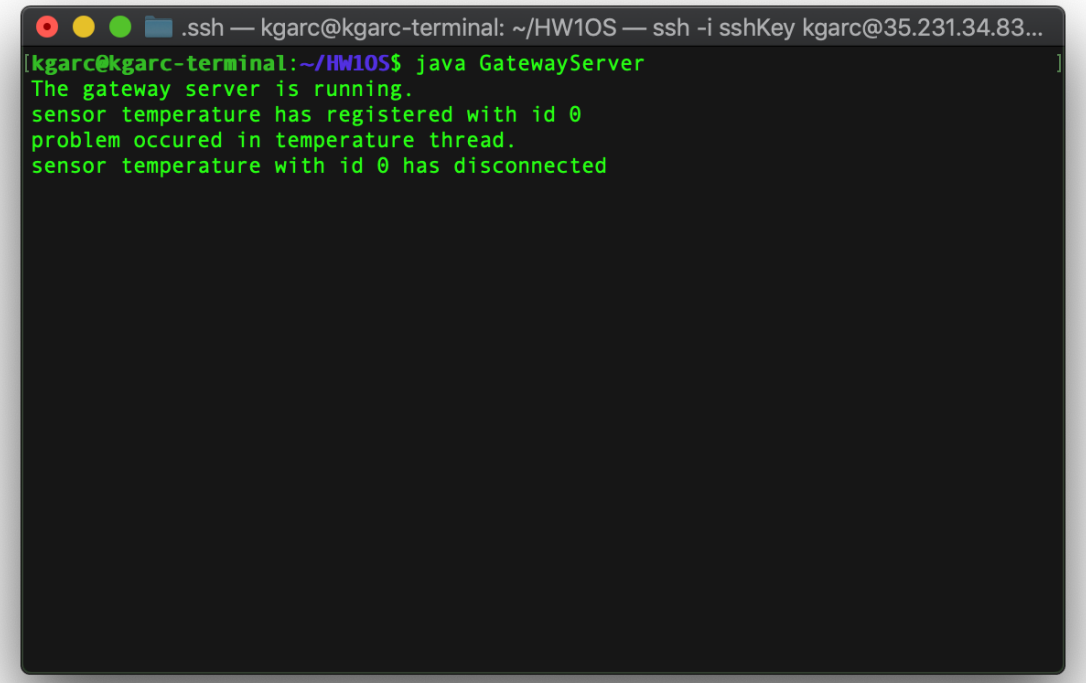
Test Case 48:

Input: Temperature stops running unexpectedly.

Pre-requisites: The server and the temperature sensor are running.

Expected result: Temperature successfully stops, the server keeps running with no error, and server prints out the message “Sensor temperature with id x has disconnected”, and that thread stops.

Actual result: Temperature successfully stops, the server keeps running with no error, and server prints out the message “Sensor temperature with id x has disconnected”, and that thread stops.



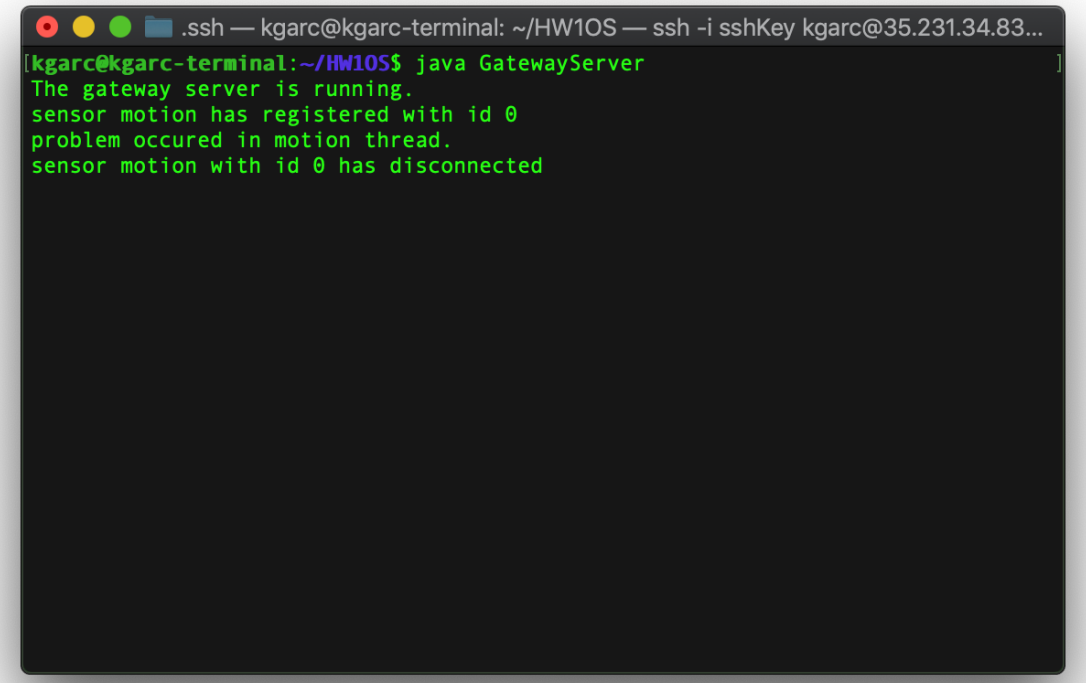
Test Case 49:

Input: Motion stops running unexpectedly.

Pre-requisites: The server and the motion sensor are running.

Expected result: Motion successfully stops, the server keeps running with no error, and server prints out the message “Sensor motion with id x has disconnected”, and that thread stops.

Actual result: Motion successfully stops, the server keeps running with no error, and server prints out the message “Sensor motion with id x has disconnected”, and that thread stops.



Test Case 50:

Input: The server stops running unexpectedly.

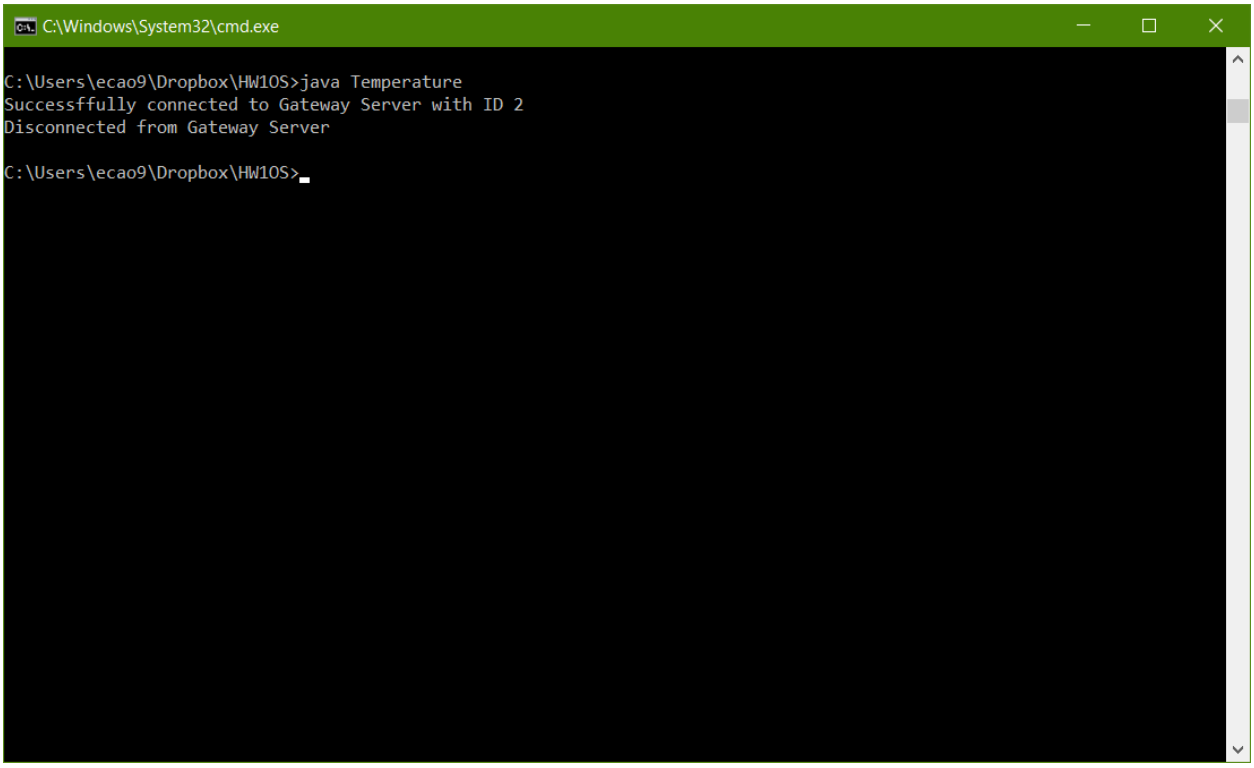
Pre-requisites: The server is running.

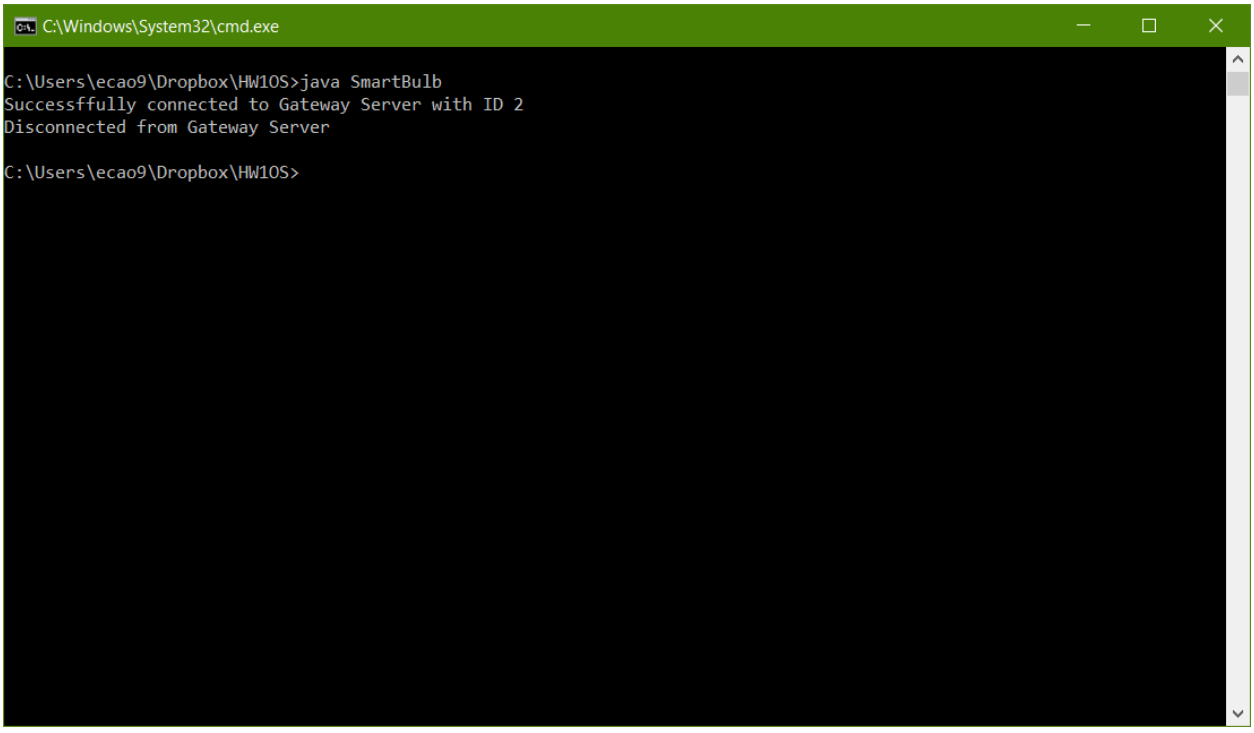
Expected result: The server successfully stops, and any running clients stop with message

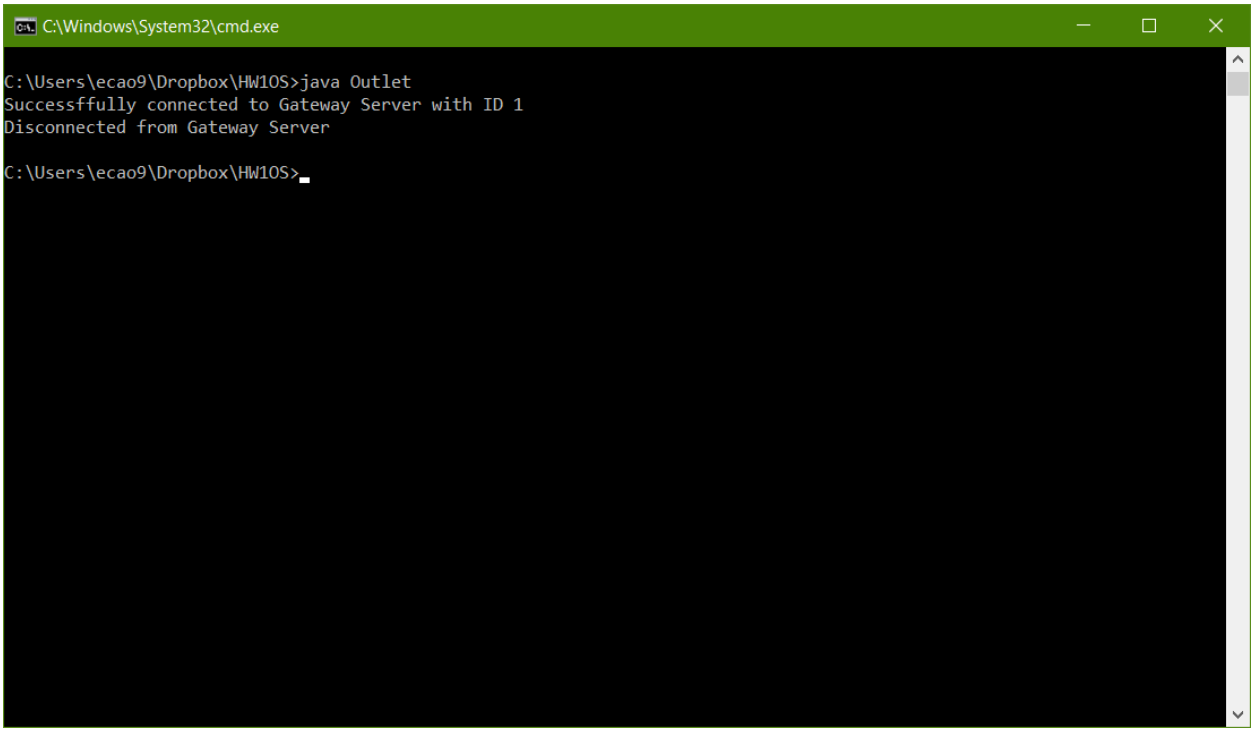
“Disconnected from Gateway Server.”

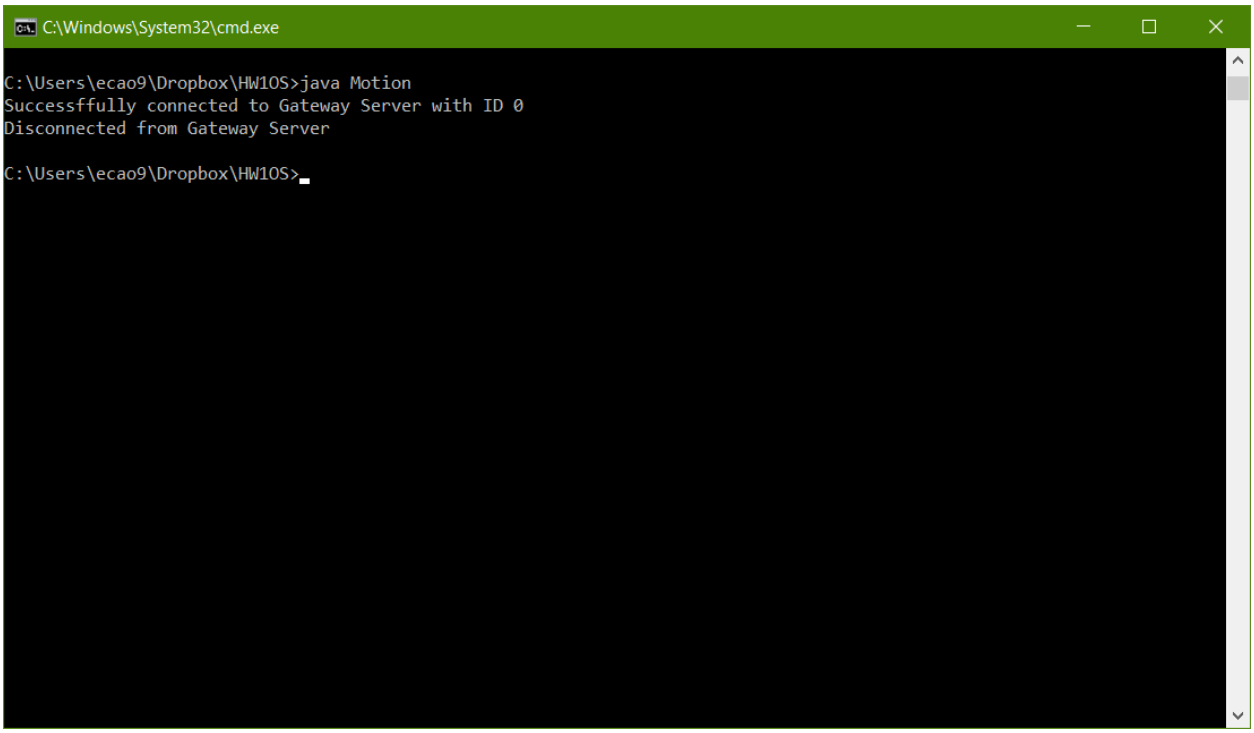
Actual result: The server successfully stops, and any running clients stop with no error

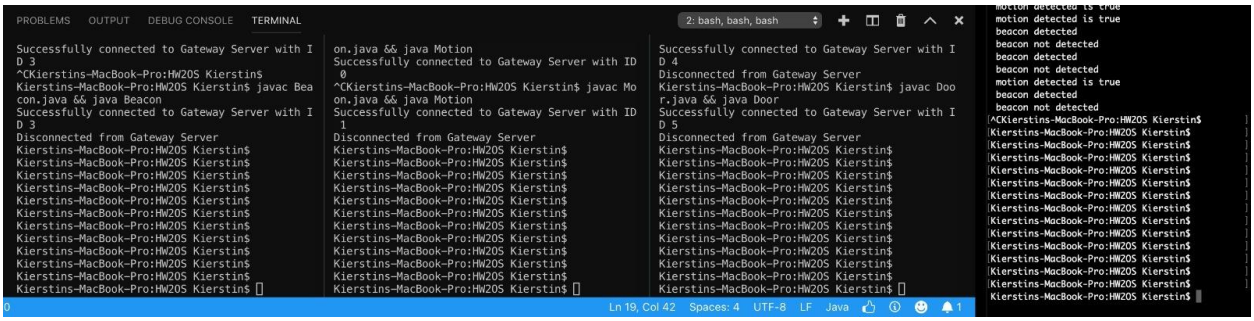
“Disconnected from Gateway Server.”


C:\Windows\System32\cmd.exe
C:\Users\ecao9\Dropbox\HW1OS>java Temperature
Successffully connected to Gateway Server with ID 2
Disconnected from Gateway Server
C:\Users\ecao9\Dropbox\HW1OS>


C:\Windows\System32\cmd.exe
C:\Users\ecao9\Dropbox\HW1OS>java SmartBulb
Successffully connected to Gateway Server with ID 2
Disconnected from Gateway Server
C:\Users\ecao9\Dropbox\HW1OS>


C:\Windows\System32\cmd.exe
C:\Users\ecao9\Dropbox\HW1OS>java Outlet
Successffully connected to Gateway Server with ID 1
Disconnected from Gateway Server
C:\Users\ecao9\Dropbox\HW1OS>


C:\Windows\System32\cmd.exe
C:\Users\ecao9\Dropbox\HW1OS>java Motion
Successffully connected to Gateway Server with ID 0
Disconnected from Gateway Server
C:\Users\ecao9\Dropbox\HW1OS>



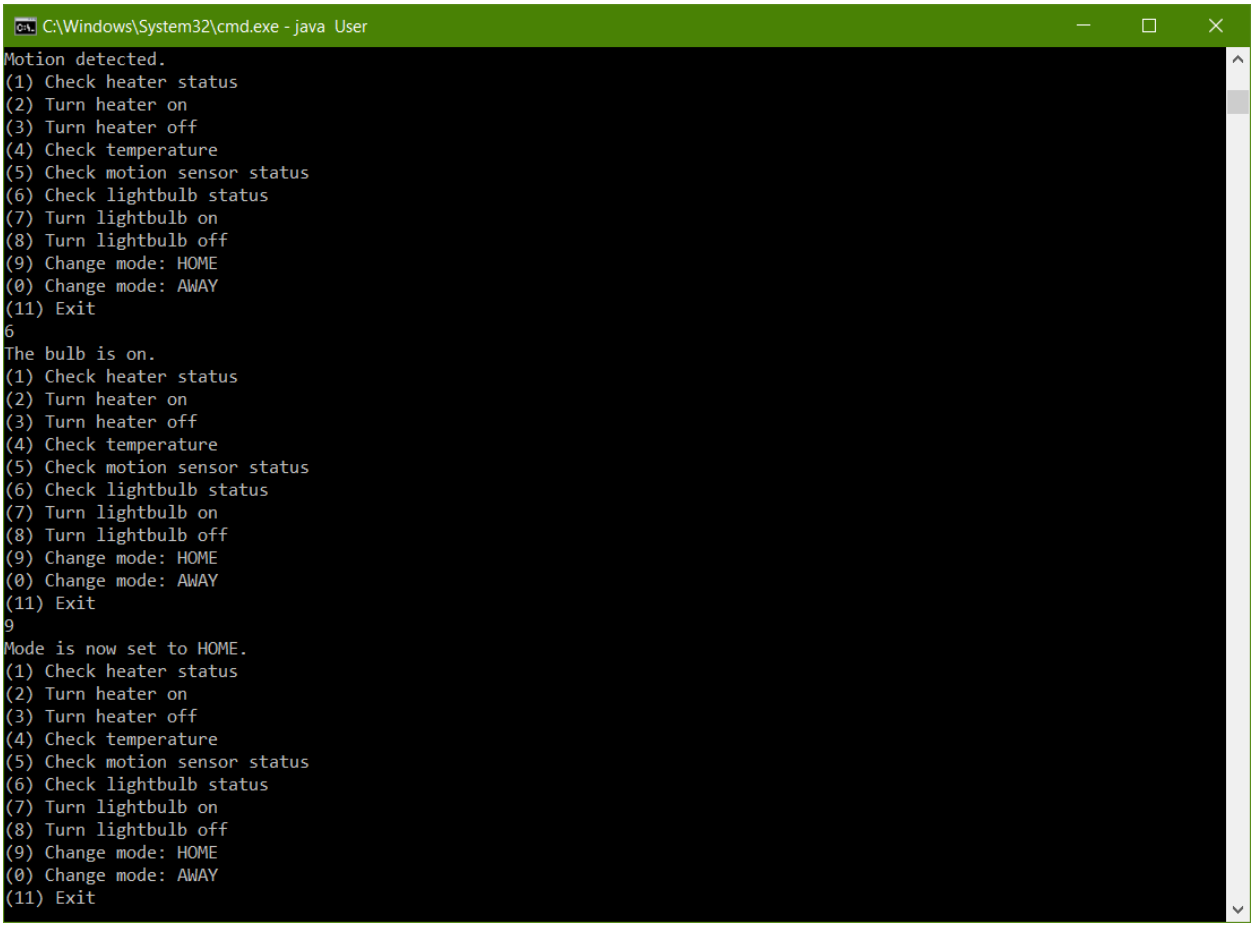
Test Case 51:

Input: Motion has been detected.

Pre-requisites: Mode has been set to HOME and everything is running.

Expected result: The lightbulb is turned on and the user does not receive a text message.

Actual result: The lightbulb is turned on and the user does not receive a text message.



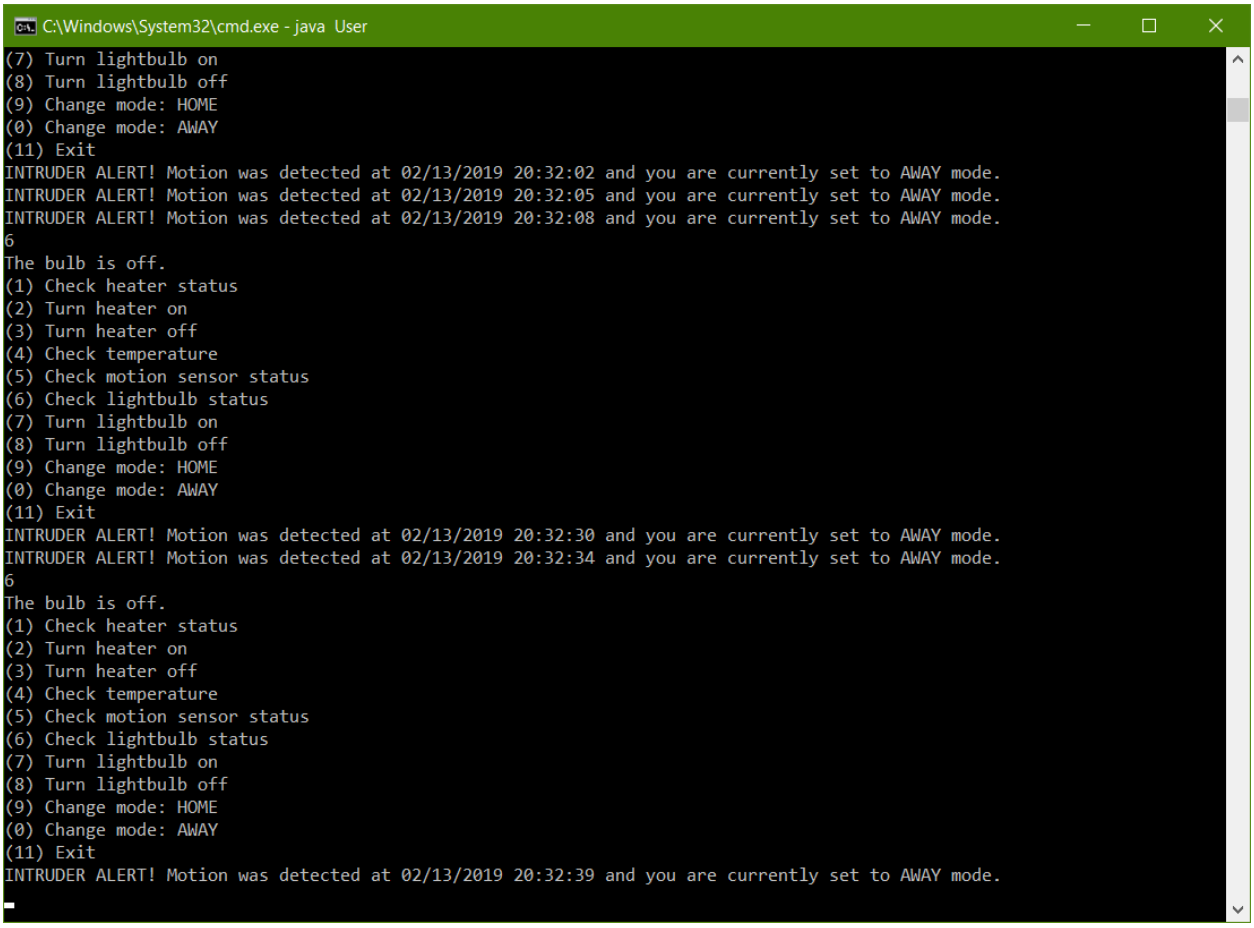
Test Case 52:

Input: Motion has been detected.

Pre-requisites: Mode has been set to AWAY and everything is running.

Expected result: The lightbulb does not change its status, and the user receives a text message warning about a possible intruder.

Actual result: The lightbulb does not change its status, and the user receives a text message warning about a possible intruder.

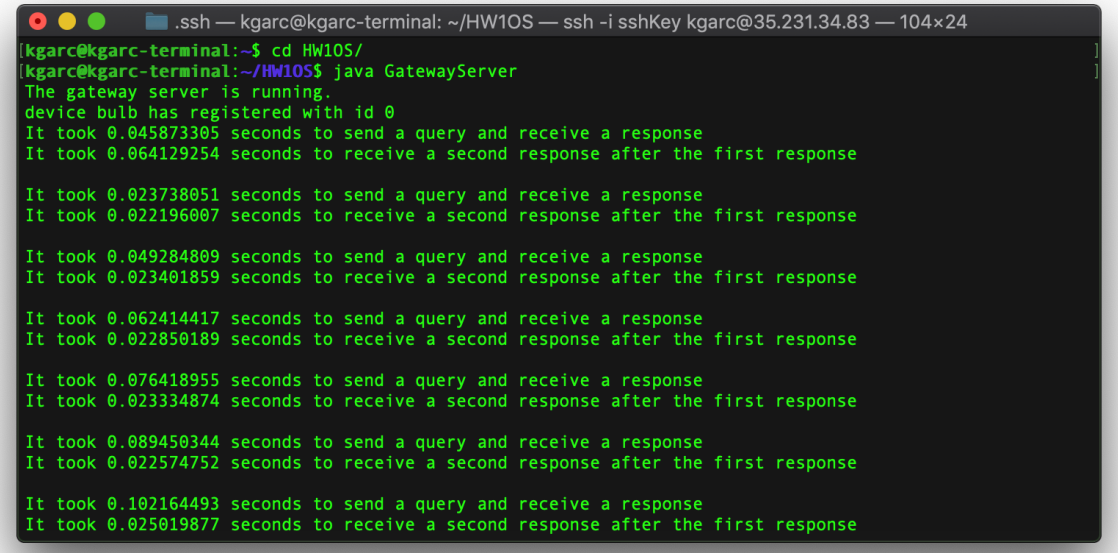

C:\Windows\System32\cmd.exe - java User
(7) Turn lightbulb on
(8) Turn lightbulb off
(9) Change mode: HOME
(0) Change mode: AWAY
(11) Exit
INTRUDER ALERT! Motion was detected at 02/13/2019 20:32:02 and you are currently set to AWAY mode.
INTRUDER ALERT! Motion was detected at 02/13/2019 20:32:05 and you are currently set to AWAY mode.
INTRUDER ALERT! Motion was detected at 02/13/2019 20:32:08 and you are currently set to AWAY mode.
6
The bulb is off.
(1) Check heater status
(2) Turn heater on
(3) Turn heater off
(4) Check temperature
(5) Check motion sensor status
(6) Check lightbulb status
(7) Turn lightbulb on
(8) Turn lightbulb off
(9) Change mode: HOME
(0) Change mode: AWAY
(11) Exit
INTRUDER ALERT! Motion was detected at 02/13/2019 20:32:30 and you are currently set to AWAY mode.
INTRUDER ALERT! Motion was detected at 02/13/2019 20:32:34 and you are currently set to AWAY mode.
6
The bulb is off.
(1) Check heater status
(2) Turn heater on
(3) Turn heater off
(4) Check temperature
(5) Check motion sensor status
(6) Check lightbulb status
(7) Turn lightbulb on
(8) Turn lightbulb off
(9) Change mode: HOME
(0) Change mode: AWAY
(11) Exit
INTRUDER ALERT! Motion was detected at 02/13/2019 20:32:39 and you are currently set to AWAY mode.

LATENCIES

```
Kevins-MacBook-Pro-2:HW2OS kingkev$ java GatewayServer
The gateway server is running.
user has registered with id 0
It took 0.011104482 to send a query to DATABASE and receive a response from DATABASE
^CKevins-MacBook-Pro-2:HW2OS kingkev$
```

```
Kevins-MacBook-Pro-2:HW2OS kingkev$ java GatewayServer
The gateway server is running.
sensor door has registered with id 0
It took 0.003674384 seconds to send and receive a message from door
^CKevins-MacBook-Pro-2:HW2OS kingkev$
```

```
Kevins-MacBook-Pro-2:HW2OS kingkev$ java GatewayServer
The gateway server is running.
sensor presence has registered with id 0
It took 0.007654669 seconds to send and receive a message from beacon
^CKevins-MacBook-Pro-2:HW2OS kingkev$
```



On average it took less than 1/10th of a second to send and receive messages.
On average it took about 0.02 seconds to receive a message without sending one.
We had the server query all the machines without a timer and the server ran into problems with synchronization. We found under our experiments that the server runs best on a timer of five seconds. However, the system still works very well at two seconds. We also found that the rate at which motion generated runs well at once a minute, however this might lead to the bulb not turning off.