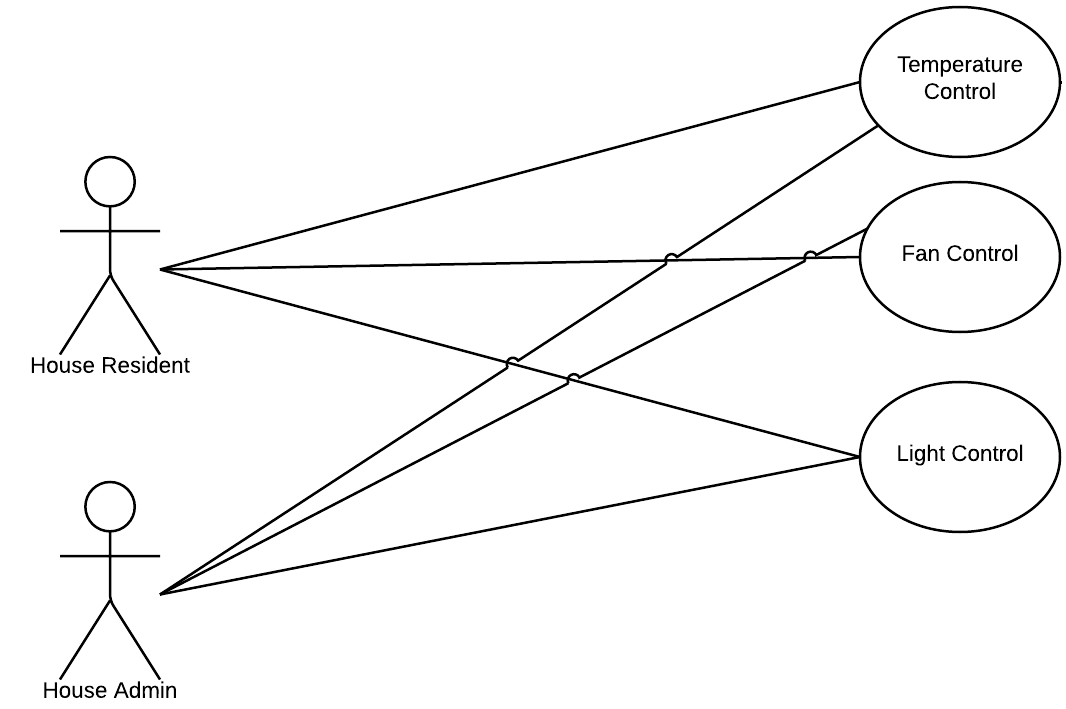
Room Subsystem

Michael Garrison

Team Members: Stephen Gondring (Door Subsystem), Sean Henshaw (House Admin Subsystem)

UML use cases diagram



Use case descriptions

Name: Temperature Control  
Actors: House Resident, House Admin  
Goals: Control the temperature in a specific room  
Preconditions: System must accept user’s secret code and user must have room privileges  
Summary: If the system accepts the user’s secret code and the user has room privileges, then the user will be able to turn the room temperature up, down, or turn the temperature control off.   
Related use cases: N/A  
Steps:   
1. User enters secret code into the system.  
2. System will verify secret code and check to see if the user has room privileges.  
3. If the secret code is accepted and user has room privileges, then the use case is successful and the system will allow the user to turn the room temperature up, down, or turn the temperature control off. Otherwise, the use case ends.   
4. If the user attempts to raise the temperature higher than the set maximum or lower than the set minimum the temperature up/down buttons will no longer increment/decrement the current temperature as a safe guard. This means that the temperature up button will not function if the current temperature is already set to the maximum temperature allowed, and the temperature down button will not function if the current temperature is already set to the minimum temperature allowed.  
Postconditions: The Room Subsystem GUI will return to the start state (Keypad) and a log with the new settings (variable) values will be dumped after the user has exited the room settings window.  
  
Scenario A)   
1. House resident John Doe wants to change the temperature in his room. John Doe walks up to the system where he is presented with the Room Subsystem GUI.  
2. John Doe uses the keypad in the Room Subsystem GUI and enters his secret code.  
3. The system goes to authenticate the secret code which turns out to be a valid code. John Doe also has set permissions for this room. The GUI now displays options for controlling the room temperature, fan, and lights.  
4. John Doe turns the temperature control up a couple of degrees but makes sure to stay below the maximum temperature that is displayed, which was previously set by the house admin.   
5. The GUI is then cleared and reset back to the keypad screen where the secret code must be entered again and authenticated to make any further changes.

Scenario B)   
1. House resident John Doe wants to change the temperature in the living room. John Doe walks up to the system where he is presented with the Room Subsystem GUI.  
2. John Doe uses the keypad in the Room Subsystem GUI and enters his secret code.  
3. The system goes to authenticate the secret code which turns out to be a valid code, however John Doe does not have set permissions for this room which causes an error code to be displayed.  
4. The GUI is then cleared and reset back to the keypad screen where the secret code must be entered again and authenticated to make any further changes.

Name: Fan Control  
Actors: House Resident, House Admin  
Goals: Control the fan in a specific room  
Preconditions: System must accept user’s secret code and user must have room privileges  
Summary: If the system accepts the user’s secret code and the user has room privileges, then the user will be able to turn the room fan on high, low, or turn the room fan off.  
Related use cases: N/A  
Steps:   
1. User enters secret code into the system.  
2. System will verify secret code and check to see if the user has room privileges.  
3. If the secret code is accepted and user has room privileges, then the use case is successful and the system will allow the user to turn the room fan on high, low, or turn the room fan off. Otherwise, the use case ends.   
Postconditions: The Room Subsystem GUI will return to the start state (Keypad) and a log with the new settings (variable) values will be dumped after the user has exited the room settings window.

Scenario A)   
1. House resident Jane Doe wants to control the fan in the living room. Jane Doe walks up to the system where she is presented with the Room Subsystem GUI.  
2. Jane Doe uses the keypad in the Room Subsystem GUI and enters her secret code.  
3. The system goes to authenticate the secret code which turns out to be a valid code. Jane Doe also has set permissions for this room. The GUI now displays options for controlling the room temperature, fan, and lights.  
4. Jane Doe turns the fan off.  
5. The GUI is then cleared and reset back to the keypad screen where the secret code must be entered again and authenticated to make any further changes.

Scenario B)   
1. House resident Jane Doe decides to turn the fan back on to low in the living room. Jane Doe walks up to the system where she is presented with the Room Subsystem GUI.  
2. Jane Doe uses the keypad in the Room Subsystem GUI and enters her secret code, but accidently puts in the wrong code.  
3. The system goes to authenticate the secret code which turns out to be an invalid code which causes an error code to be displayed.  
4. The GUI is then cleared and reset back to the keypad screen where the secret code must be entered again and authenticated to make any further changes.

Name: Light Control  
Actors: House Resident, House Admin  
Goals: Control the light in a specific room  
Preconditions: System must accept user’s secret code and user must have room privileges  
Summary: If the system accepts the user’s secret code and the user has room privileges, then the user will be able to turn the room light up, down, or turn the room light off.  
Related use cases: N/A  
Steps:   
1. User enters secret code into the system.  
2. System will verify secret code and check to see if the user has room privileges.  
3. If the secret code is accepted and user has room privileges, then the use case is successful and the system will allow the user to turn the room light up, down, or turn the room light off. Otherwise, the use case ends.   
Postconditions: The Room Subsystem GUI will return to the start state (Keypad) and a log with the new settings (variable) values will be dumped after the user has exited the room settings window.

Scenario A)   
1. House resident Jane Doe wants to control the light in the living room. Jane Doe walks up to the system where she is presented with the Room Subsystem GUI.  
2. Jane Doe uses the keypad in the Room Subsystem GUI and enters her secret code.  
3. The system goes to authenticate the secret code which turns out to be a valid code. Jane Doe also has set permissions for this room. The GUI now displays options for controlling the room temperature, fan, and lights.  
4. Jane Doe turns the light up to an optimum setting for reading.   
5. The GUI is then cleared and reset back to the keypad screen where the secret code must be entered again and authenticated to make any further changes.

Scenario B)   
1. House resident Jonnie Doe decides to turn the light off in the living room to play a trick on his mom. Jonnie Doe walks up to the system where he is presented with the Room Subsystem GUI.  
2. Jonnie Doe uses the keypad in the Room Subsystem GUI and enters his secret code.  
3. The system goes to authenticate the secret code which turns out to be a valid code, however Jonnie Doe does not have set permissions for this room which causes an error code to be displayed.  
4. The GUI is then cleared and reset back to the keypad screen where the secret code must be entered again and authenticated to make any further changes.