Practical Guidelines for Tests through Safehome Example

Moonzoo Kim

SoC KAIST

General Guidelines for Tests

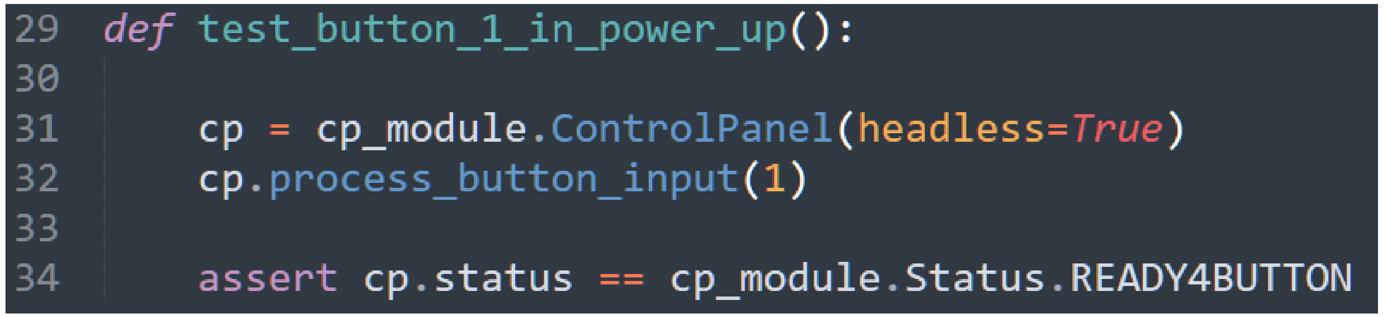
- Tests should be carefully designed based on SRS, SDS, and the implementation.
- Design tests of various levels based on SRS/SDS as follows (but, not limited to)
  - Unit tests: based on state diagrams, class diagrams (and implicitly based on CRC cards and sequence diagrams)
  - Integration tests: class diagrams, CRC cards, and sequence diagrams
  - System-level tests: based on CRC cards, sequence diagrams and use case scenarios
- Test documents should refer to corresponding requirements and designs, if any.
- Usually, test code file (e.g., test_xx.py) should be separated from target source code file for better maintenance.

General Guidelines for Unit Tests

- You are expected to generate at least one unit test for every method in your target system
  - but not for a method in 3rd party library (e.g., the virtual device in safehome)

- A unit test usually means a test method that
  - does not have parameters (i.e., 1 unit test per 1 specific input value), and
  - has `assert` statement(s) at (the end of) the test method

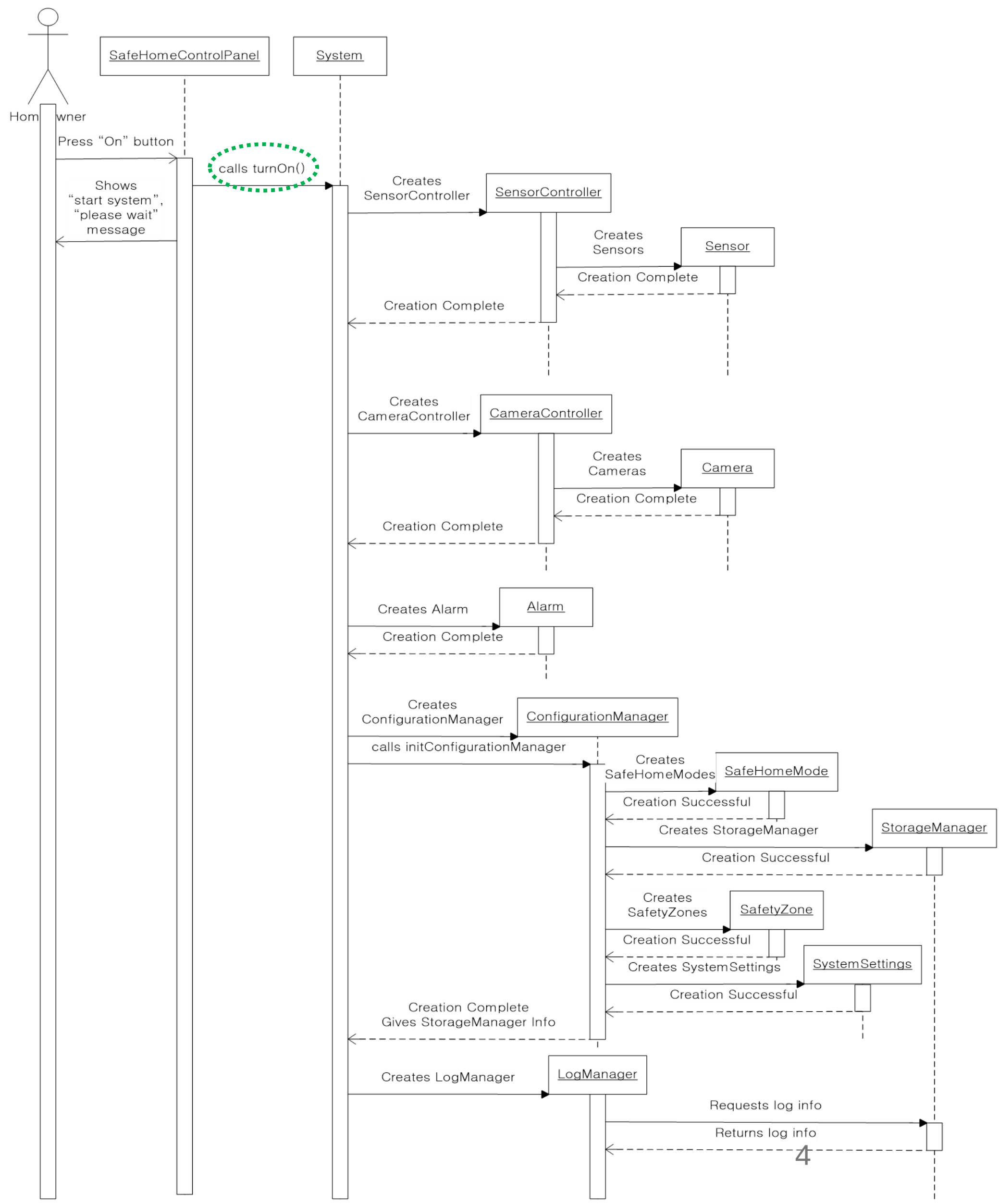
Unit Test Example



```python
29  def test_button_1_in_power_up():
30
31      cp = cp_module.ControlPanel(headless=True)
32      cp.process_button_input(1)
33
34      assert cp.status == cp_module.Status.READY4BUTTON
```

- **Test_**`button_1_in_power_up()`
  - Unit test to test `ControlPanel.process_button_input` w/ a parameter 1
  - Context/precondition: `ControlPanel`'s constructor waits for a user to press a power button (i.e. button 1) and the button value 1 is passed as a parameter to `process_button_input`
  - Test oracle (see `assert` at line 34): after pressing button 1, control panel's status should be `READY4BUTTON`
- Corresponding state diagrams, CRC cards, sequence diagrams should be referenced in the test document
  - Ex. A target method under test `process_button_input` is a part of "Turn the system on" sequence diagram on page 50 of SDS (see the green oval in the sequence diagram on the right), although it is not explicitly shown in the diagram.
- Unit test may not have corresponding diagrams, since the granularity of a unit is very small
  - Ex. Ex. A target method under test `process_button_input` does not have a corresponding state diagram in SDS

Sequence Diagram for Turn the System On(1)

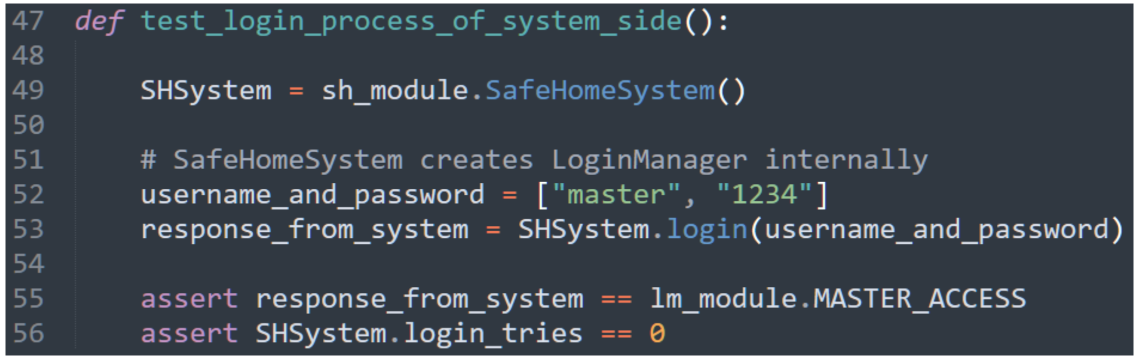


Example of Unit Test Document

1. **External Communication Management**
   A. **Control Panel Management**
      1. `ControlPanel` class
         A. `process_button_input()` (handling key input from CP)
            A. `test_button_1_in_power_up` (UT-CP-btn1-PW-U)

Class	Method	Author	Date	Version
`ControlPanel`	`process_button_input()`			1.0.1
Test Case Description				
Verifies that pressing a button “1” results in `status == READY4BUTTON`, when the system is turned off.				
Context/Precondition/Input Specifications				
Call `process_button_input(1)` while the safehome system is powered off (which is performed by the constructor of `ControlPanel` class).				
Expected Result				
CP’s status remains `Status.READY4BUTTON`				
Actual Result (Pass/Fail/Exception)				
Pass: status remains `Status.READY4BUTTON`				
Comment				
Reference: ”Turn the system on” sequence diagram on page 50 of SDS				

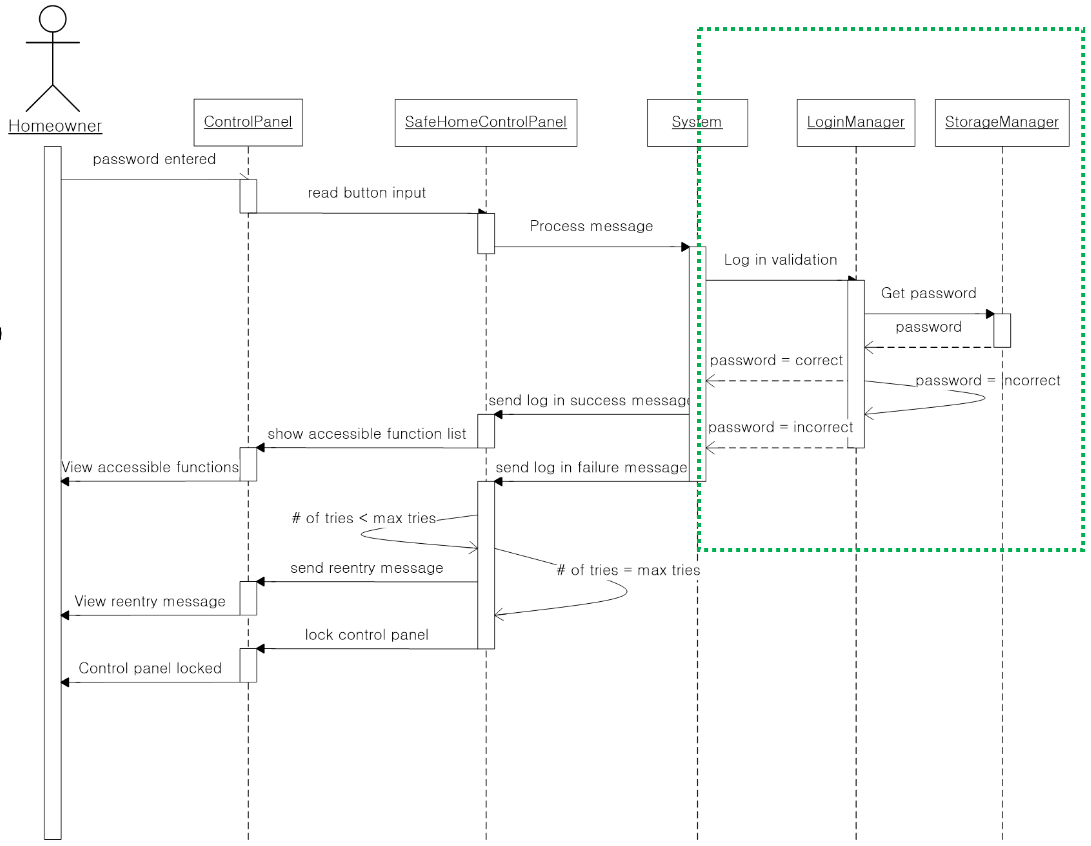
Integration Test Example



```
47  def test_login_process_of_system_side():
48  
49      SHSystem = sh_module.SafeHomeSystem()
50  
51      # SafeHomeSystem creates LoginManager internally
52      username_and_password = ["master", "1234"]
53      response_from_system = SHSystem.login(username_and_password)
54  
55      assert response_from_system == lm_module.MASTER_ACCESS
56      assert SHSystem.login_tries == 0
```

- **Test_**`login_process_of_system_side()`
  - Integration test to test login process in system side (i.e., not including user interaction through control panel)
  - Context/precondition:
    - a homeowner (by default, its user name is "master") entered a password (i.e., "1234") through control panel, which is stored in `username_and_password` (see line 52). We assume that "1234" is the correct password.
    - Then, the homeowner's password is passed as a parameter to `login`
  - Test oracle (see `assert` at line 55 and 56):
    - LoginManager should accept the password (i.e., `login` returns `lm_module.MASTER_ACCESS`)
    - A number of login trial should not be increased, since the given password is correct
- Corresponding CRC cards, sequence diagrams should be referenced in the test document
  - Ex. The integration testing scenario is a part of "Log onto the system through control panel" sequence diagram on page 47 of SDS (see the green box in the sequence diagram on the right)

a. Log onto the system through control panel



Example of Integration Test Document

A. Log onto system

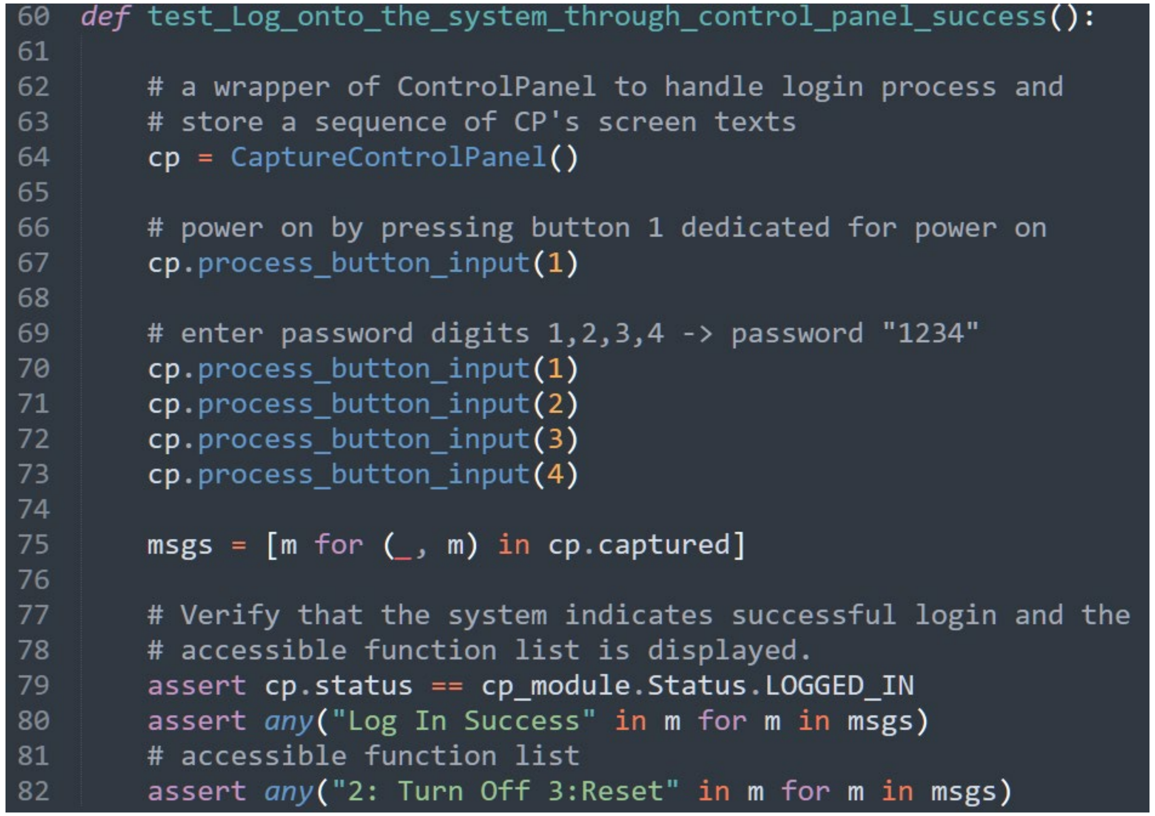
1. test_login_process_of_system_side (IT-Login-Sys-Su)

Class	Function	Author	Date

`System`
`LoginManager`
`StrageManager`
	Log onto the system		

Test Case Description			
Validate that `SafeHomeSystem` successfully authenticates the master user when valid credentials are provided and `StorageManager` returns the matching login record.			
Input Specifications			
Database has predefined username and password: “master” and “1234” Put list username and password: (["master", "1234”])			
Detailed Step			
1. Initialize `SafeHomeSystem`, which internally creates an instance of `LoginManager` and prepares `StorageManager`. 2. Call `SafeHomeSystem.login()` with "master" as the username and "1234" as the password. 3. Allow `LoginManager` to validate the provided credentials by comparing them against the stored credentials retrieved from `StorageManager`. 4. After successful authentication, `SafeHomeSystem` resets the login attempt counter and returns the access code for master. 5. Confirm that `SafeHomeSystem` does not increment the login attempt count after the successful authentication response.			
Expected Result			
`login()` of `System` class returns the `MASTER_ACCESS` (0x01). `login_tries` of `System` equals to 0 after successful authentication.			
Actual Result (Pass/Fail/Exception)			
Pass: `login()` of `System` class returns the `MASTER_ACCESS` (0x01). Pass: `login_tries` of `System` equals to 0 after successful authentication.			
Comment (including references)			
The right half of Sequence Diagram on System, LoginManager, and StorageManager, page 47 of SDS			

System-level Test Example



```
60 def test_Log_onto_the_system_through_control_panel_success():
61
62     # a wrapper of ControlPanel to handle login process and
63     # store a sequence of CP's screen texts
64     cp = CaptureControlPanel()
65
66     # power on by pressing button 1 dedicated for power on
67     cp.process_button_input(1)
68
69     # enter password digits 1,2,3,4 -> password "1234"
70     cp.process_button_input(1)
71     cp.process_button_input(2)
72     cp.process_button_input(3)
73     cp.process_button_input(4)
74
75     msgs = [m for (_, m) in cp.captured]
76
77     # Verify that the system indicates successful login and the
78     # accessible function list is displayed.
79     assert cp.status == cp_module.Status.LOGGED_IN
80     assert any("Log In Success" in m for m in msgs)
81     # accessible function list
82     assert any("2: Turn Off 3:Reset" in m for m in msgs)
```

- **`Test_`**`log_onto_the_system_through_control_panel_success()`
  - Automated system-level test to test whole login process including user interaction through control panel (see lines 64-73), which is based on the sequence diagram on page 47 of SDS (see the previous slide).
  - Context/precondition:
    - `CaptureControlPanel` (see line 64) is a wrapper of `ControlPanel` to handle the login process and store its screen output
    - The homeowner's password is "1234"
  - Test oracles (see `assert` at line 79-82):
    - A homeowner should login correctly (i.e., `cp.status` should be `cp_module.Status.LOGGED_IN`) (line 79)
    - A screen output should contain "Log In Success" message (line 80)
    - A screen output also should show accessible function list (line 82)
- Note that, to test end-to-end system including compatibility with hardware peripherals (e.g., physical button press, physical screen output, etc.), a system-level test can be manually performed
  - You may press buttons w/ your hands (mouse clicks) and see the screen output w/ your eyes instead of using `Test_log_onto_the_system_through_control_panel_success()`
- Corresponding CRC cards, sequence diagrams should be referenced in the test document
  - Ex. The system-level testing scenario corresponds to "Log onto the system through control panel" sequence diagram on page 47 of SDS

Example of System-level Test Document

A. Log onto the system through control panel

1. Log onto the system through control panel success (ST-Login-Using-CP-Su)

Class	Function	Author	Date
ControlPanel SafeHomeControlPanel System LoginManager StrageManager	Log onto the system through control panel		
Test Case Description			
Validate the successful login path and the ControlPanel display of available functions after the authentication.			
Input Specifications			
ControlPanel receives button-press sequence of "1234". StorageManager contains master credentials ("1234").			
Detailed Step			
1. Seed StorageManager with the master user and password "1234" (if not already present). Reset login attempt counter. 2. Initialize ControlPanel and Simulate powering on the panel 3. Simulate button presses sequence of "1234" 4. Verify that the system indicates successful login and the accessible function list is displayed.			
Expected Result			
SafeHomeSystem.login() returns success code for master. ControlPanel transitions to LOGGED_IN state. ControlPanel displays "Log In Success" followed by accessible functions message such as "2: Turn Off 3: Reset.			
Actual Result (Pass/Fail/Exception)			
Pass: SafeHomeSystem.login() returns success code for master. Pass: ControlPanel transitions to LOGGED_IN state. Pass: ControlPanel displays "Log In Success" followed by accessible functions message such as "2: Turn Off 3: Reset.			
Comment (including references)			
Sequence Diagram on ControlPanel, SafeHomeControlPanel, System, LoginManager, and StorageManager, page 47 of SDS			