1 User Interface

User interface is useful for taking input from user and giving results to the user. First this module asks user to load an input file and passes it to parser module. After initialization of the PMs and VMs by the PM modifier module it displays a screen with buttons specifying the PMs and labels specifying their respective VMs.
It also has buttons to specify functionalities of the the system such as addVM,deleteVM,reset and consolidate.

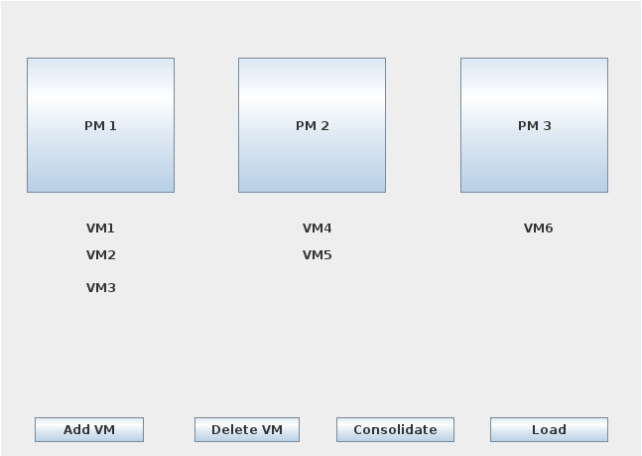

Figure 1: User interface

1.1 Internal Functions

callParser()

Initially user interface has a load button.On clicking, it opens a file chooser which helps the user to choose an input file.
The selected input file is passed to the parser module.

Input parameters	: Click event
Output parameters	: File
Returns	: UI is updated on success. If the file chosen is not in the specified format an error message is returned.

Pseudo code :

1: on clicking load

2: open file chooser
3: **if** file choosen **then**
4: call parser(file) in parser module
5: call updateUI()
6: **else**
7: no change
8: **end if**

Figure 2: Initial GUI

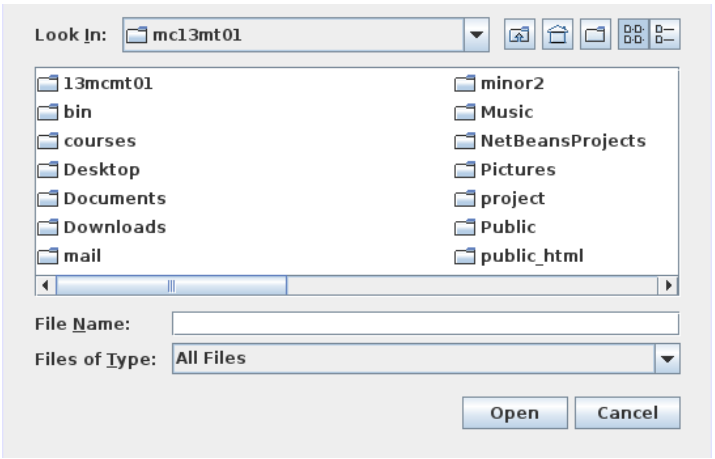

Figure 3: File chooser

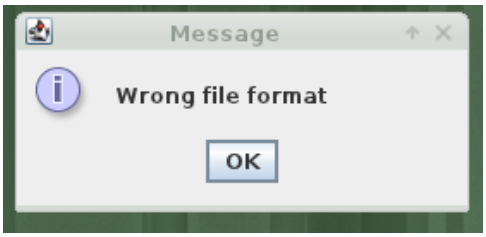

Figure 4: Error message

callAddVM()

On click addvm button ,a window asking the user to enter VM_ID and capacity opens. Once the user specifies VM_ID and its capacity it calls the addVM() in PM modifier module.And then calls updateUI() which is an internal function.

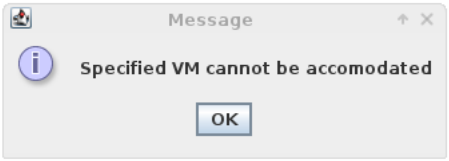

Figure 5: Error message

Input parameters	: Click event
Output parameters	: VM_ID ,Capacity
Returns	: UI is updated on success. Error message is returned if the specified VM cannot be accommodated by any of the PMs.
Error condition	: Entering negative numbers,alphanumerics,special characters, numbers greater than physical machine capacity can be avoided by data validation.

Pseudo code :

```
1: on clicking addVM
2: open dialog asking for VM_ID and capacity
3: input validation
4: if valid then
5:     call addVM(VM_ID,capacity) in PM modifier
6:     call updateUI()
```

7: **else**
8: show error message
9: **end if**

callDeleteVM()

On clicking the deleteVM button in user interface, it opens a window which prompts the user to choose a VM for deletion. This calls the deleteVM() in PM modifier module .Later calls an internal funtion updateUI().

Input parameters	: Click event
Output parameter	: VM_ID
Returns	: UI is updated on success.
Error conditions	: Choosing an incorrect VM_ID. But this case is avoided by listing all the VMs present in the system.

1: on clicking deleteVM
2: open dialog displaying list of VMs ,asking user to choose VM for deletion
3: **if** choosen **then**
4: call deleteVM(VM_ID)
5: call updateUI()
6: **else**
7: do nothing
8: **end if**

Figure 6: VM Deletion

callConsolidate()

On clicking consolidate button in UI it opens a window for confirmation from the user
If the user chooses not to consolidate the state of the system does not change.
If the user chooses to consolidate , then conolidate() funtion in PM modifier is called.
The user interface is updated using the updateUI() function.

Input parameters	: Click event
Output parameters	: None
Returns	: UI is updated and the color of the physical machines that are switched off changes.
Error condition	: None

Pseudo code :

```
1: on clicking consolidate
2: open dialog for confirmation
3: if confirmed then
4:     call consolidate()
5:     call updateUI()
6: else
7:     do nothing
8: end if
```

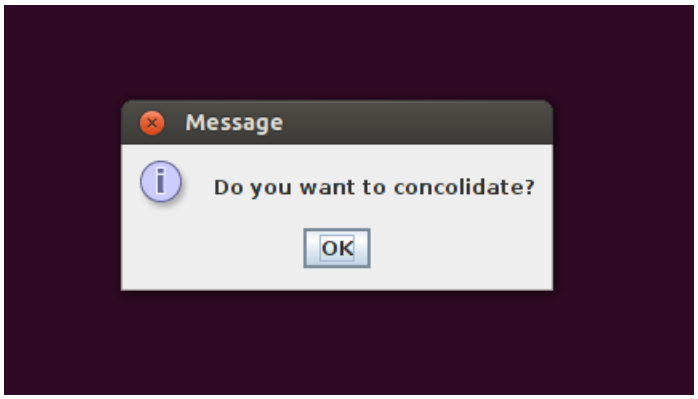

Figure 7: Consolidation

updateUI()

This function is called after every change performed on the system to reflect changes to the user.It calls the status() function in PM modifier module.

Input parameters	: None
Output parameters	: None
Returns	: Updated UI
Error conditions	: None

Pseudo code :

```
1: call status()
2: accordingly modify PMs and VMs
```

OnPM()

This function is used to switch on a PM manually by the user.When user clicks a PM which is in off state,changes its state to ON by calling the switchOnPM() function in PM modifier.Then calls updateUI(). If user chooses a PM that is already in on state ,a message is prompted to user if he/she wishes to turn off respective PM

Input parameters	: Click event
Output parameters	: PM_ID
Returns	: UI is updated on success.The color of the PM that is turned on changes.On failure returns an error message.

Pseudo code :

```
1: on clicking PM
2: if PM in OFF state then
3:    OPEN DIALOG FOR CONFIRMATION
4:    if confirmed then
5:       call switchOnPM() in PM modifier
6:    else
7:       no change
8:    end if
9: end if
```

OffPM()

This function is used to switch off a PM manually by the user.When user clicks

a PM which is turned on,changes its state to OFF by calling switchOffPM() function in PM modifier.

Input parameters	: Click event
Output parameters	: PM_ID
Returns	: UI is updated on success.The color of the PM that is turned off changes.On failure returns an error message.
Error conditions	: If the VMs in selected PM cannot be accommodated in another PMs.

Pseudo code :

```
1: on clicking PM
2: if PM is in ON state then
3:    open dialog for confirmation
4:    if confirmed then
5:       call switchOffPM()
6:       if no error then
7:          call updateUI()
8:       else
9:          show error message
10:      end if
11:   else
12:      no change
13:   end if
14: end if
```