1. **Functional Requirement**

A cloud orchestration is required which could provide IaaS to the user. The main objective to develop this product is to use institutes resources efficiently for research purposes, simulation purposes etc. Requirement of cost cutting on the purchase of hardwares because most of the resources goes unused. To make most out of the resources, they are integrated into one cloud (baadal) to provide VMs to users. Following are the requirements as perceived by different users:

Student as a user:

- Basic operation on VM
    - Start
    - suspend
    - delete
    - pause
    - resume
    - shutdown
    - Take,delete and configure snapshot of the VM
    - Clone VM
    - Attach extra HDD to the VM
    - Get VNC
    - Get VM history and performance

- Operation as a User
    - Register as a user
    - List all the VMs
    - Mailing service to Admin

Faculty as a User

- **All the functionality of User(Student)**
- See pending request (of VMs) from the students under him/her
- Approve, delete, edit the requests.
- Verify VM owner

OrgAdmin as a User

- **All the functionality of User(Student)**
- List all organization level VM
- Approve, Delete, Edit the requests.
- Reminding the faculty for the task to be done by that faculty

Admin as a User

- VM level functionality
    - List all VMs
    - Manage VM templates
    - Migrate, lock VM

    - Sync and deletion of VM
    - Get VM resource utilization
    - Verification of VM resource
    - Launch VM Image

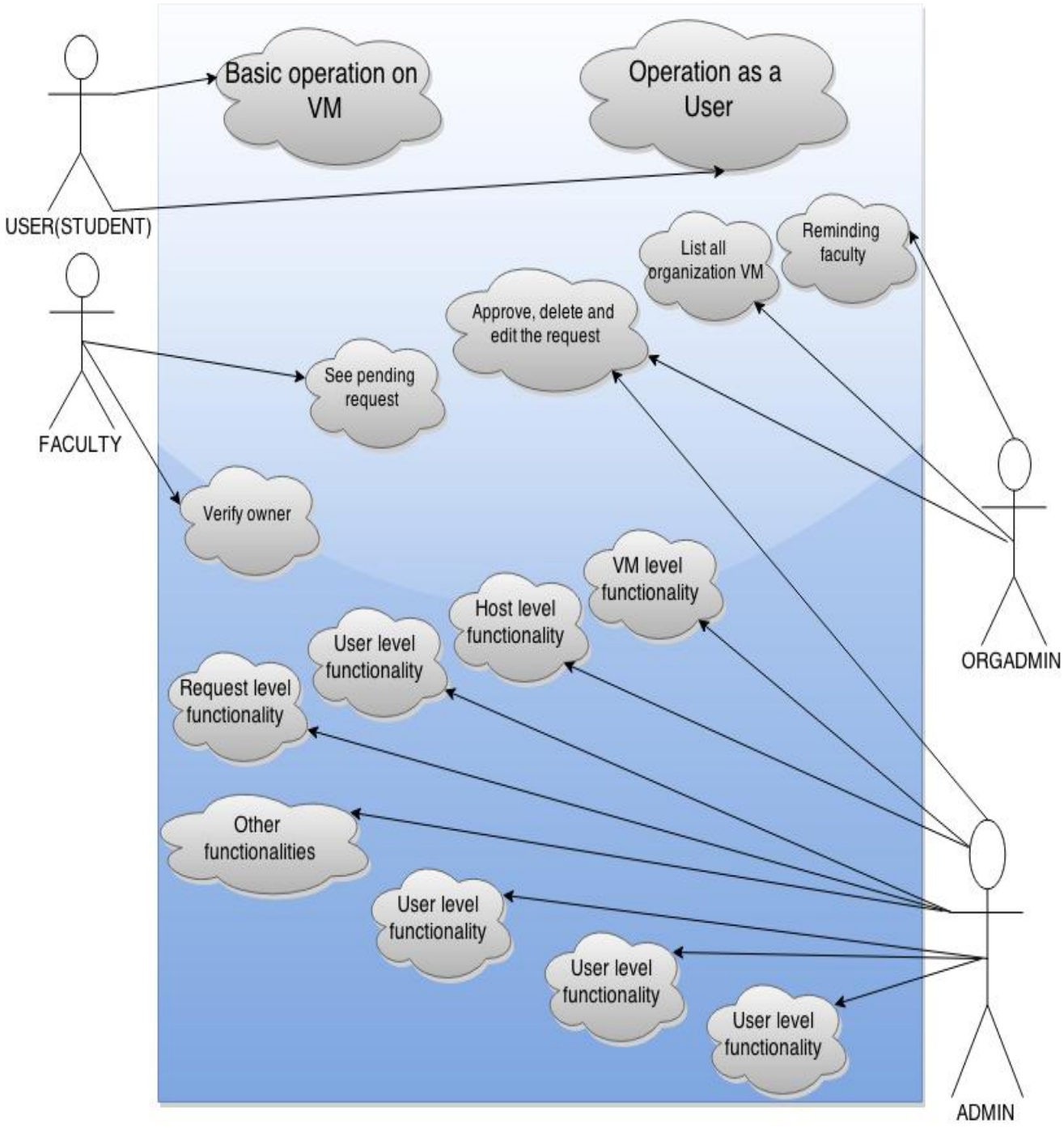
- Host level functionality
    - Get Host details and utilization data
    - Add host
    - Put host on maintenance
    - Boot, shutdown, delete host
    - Show host performance
    - Get updated host graph
    - Host configuration

- Request level functionalities
    - List all pending requests (of any type of owner)
    - Approve, delete the requests.

- User level functionalities
    - Modify user role and type of roles
    - Approve and mail user (of any type)
    - Get user details
    - Add, remove user to VM
    - Remind OrgAdmin about request under him/her
    - Add user with role define for him/her
    - Verify and Remove user
    - Verify extra disk requested by the user

- Other functionalities
    - Manage security domains
    - Manage datastore
    - Get task list from the TASK Queue
    - Ignore, retry any task
    - Perform Sanity check
    - Snapshot and sync sanity check
    - Manage and validate public and private ip pool and range

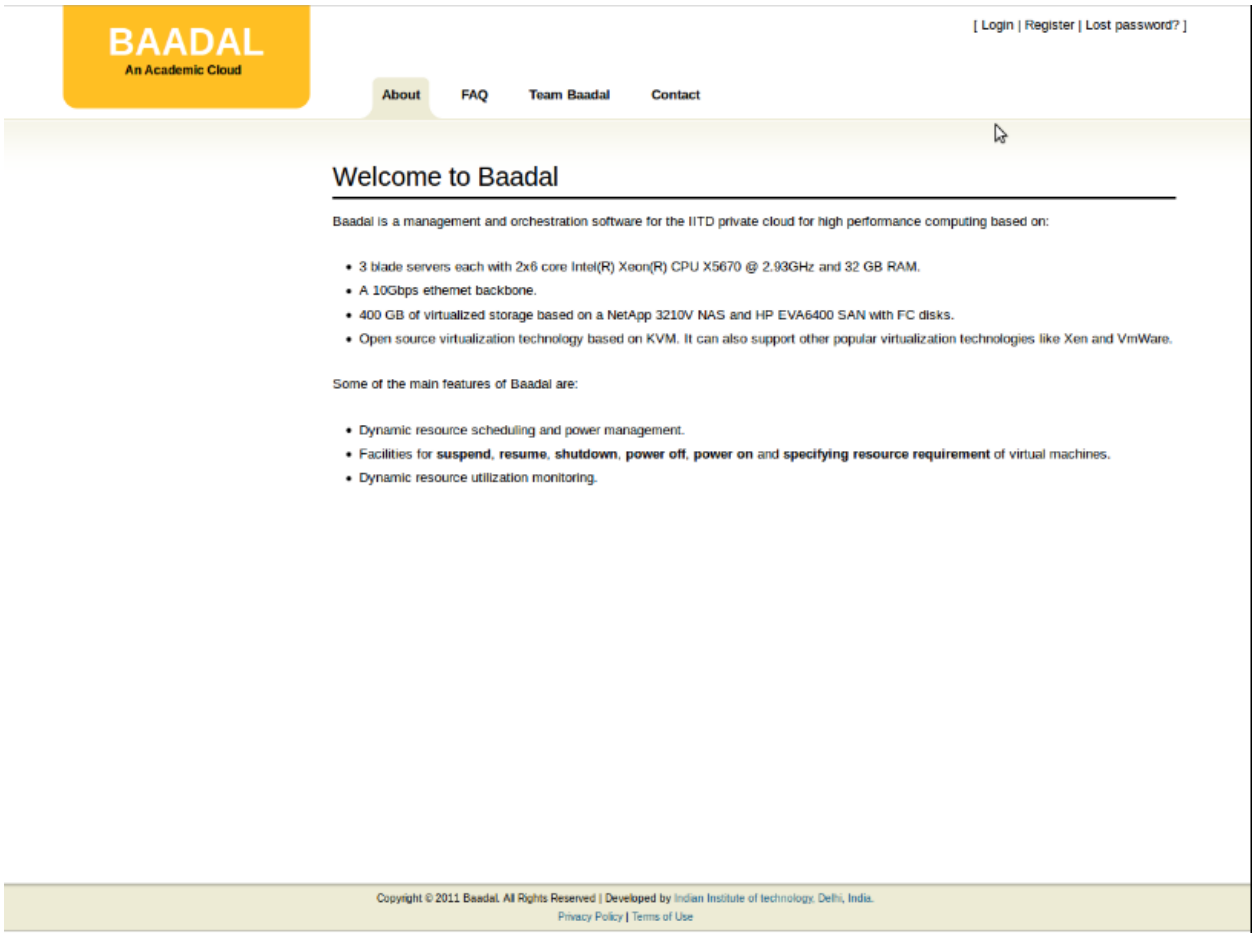
- Get baadal status
- Shutdown and bootup baadal
- Send shut down baadal mail to all the users


USER(STUDENT)
Basic operation on VM
Operation as a User
FACULTY
See pending request
Verify owner
Approve, delete and edit the request
List all organization VM
Reminding faculty
ORGADMIN
VM level functionality
Host level functionality
User level functionality
Request level functionality
Other functionalities
User level functionality
User level functionality
User level functionality
ADMIN

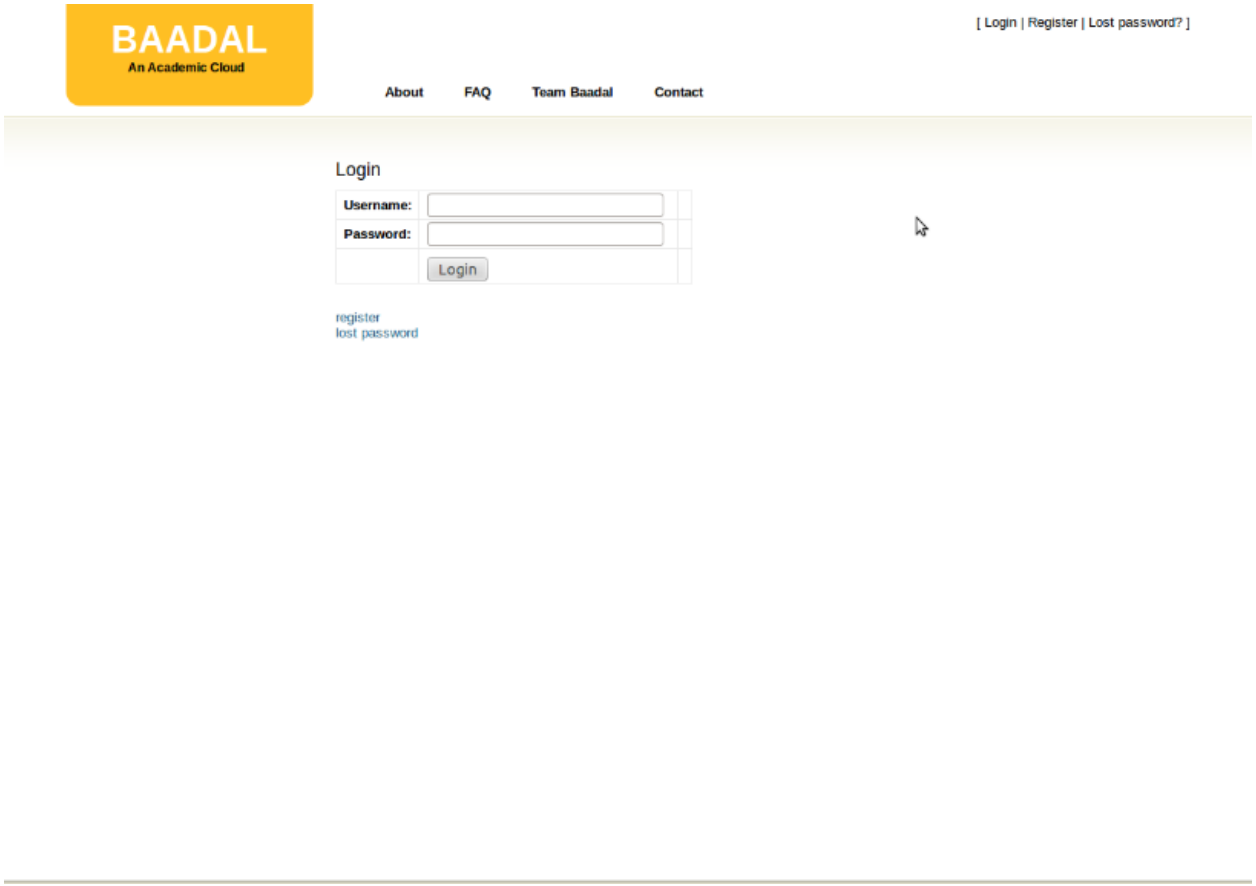
ADMIN FR

1. Baadal HOME PAGE

To open the utility user has to choose any browser (firefox, chrome etc). Then in the address bar, https://<IP_OF_CONTROLLER> and then press enter.
The browser should open the following page:

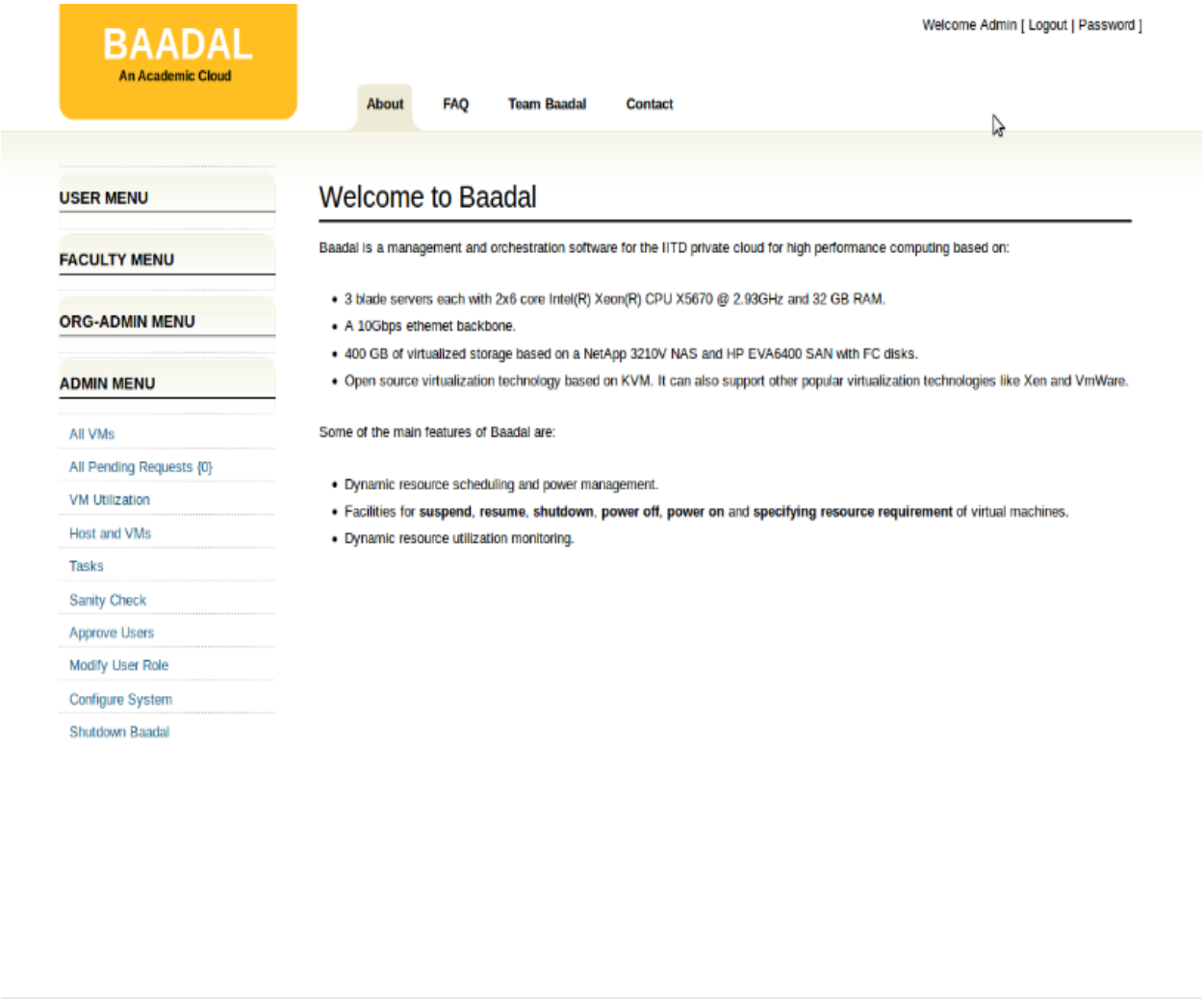


Click on the login at the right upper corner. Following login page should get open in the browser.


BAADAL
An Academic Cloud
[Login | Register | Lost password?]
About
FAQ
Team Baadal
Contact
Login
Username:
Password:
Login
register
lost password

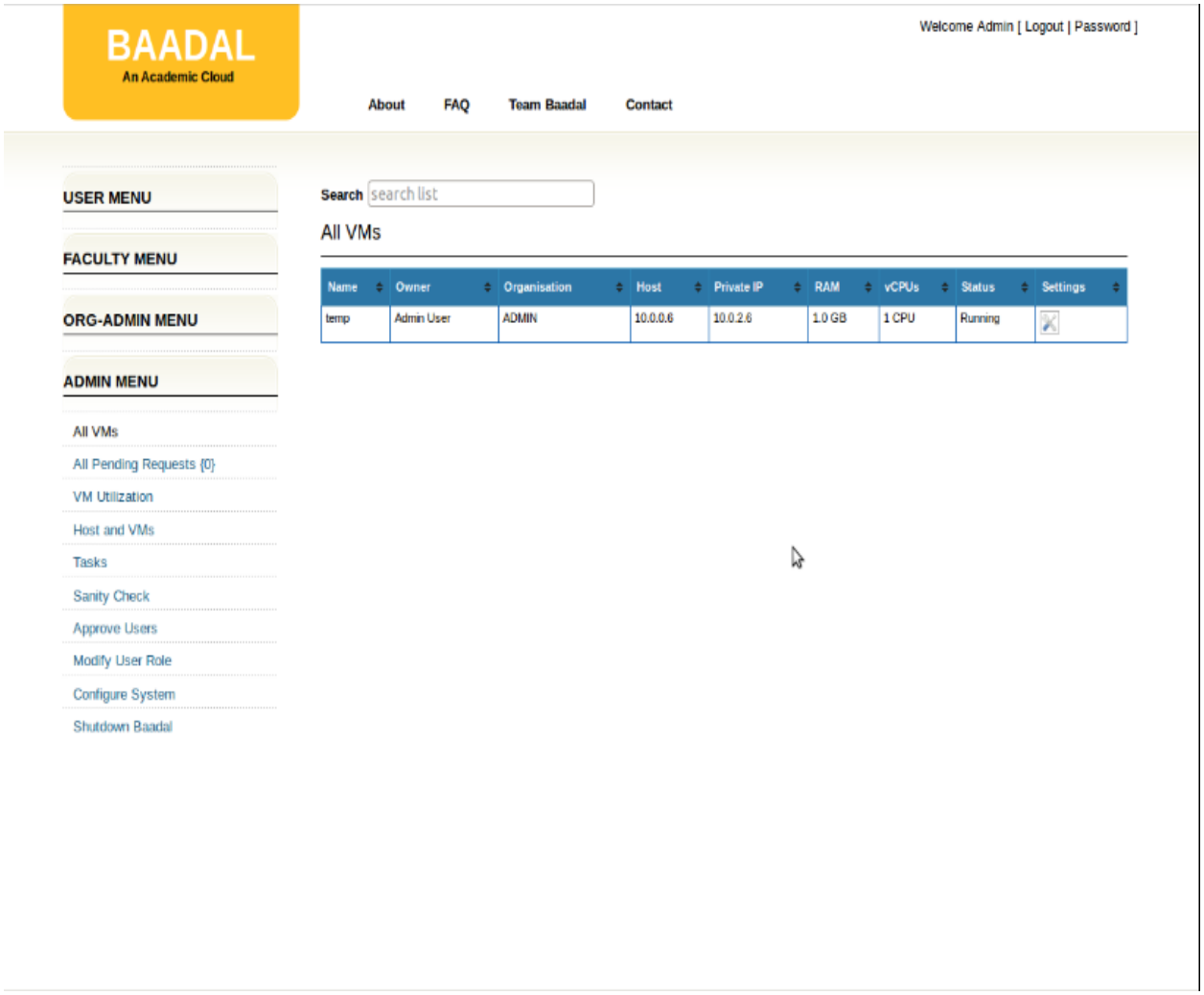
2. Title: Admin Interface

On logging into the admin account admin should get the functionality of all the users [“Student”, ”Faculty”, ”Org-Admin”, ”ADMIN”]. Following should be the output.



3. Title : All VMs list

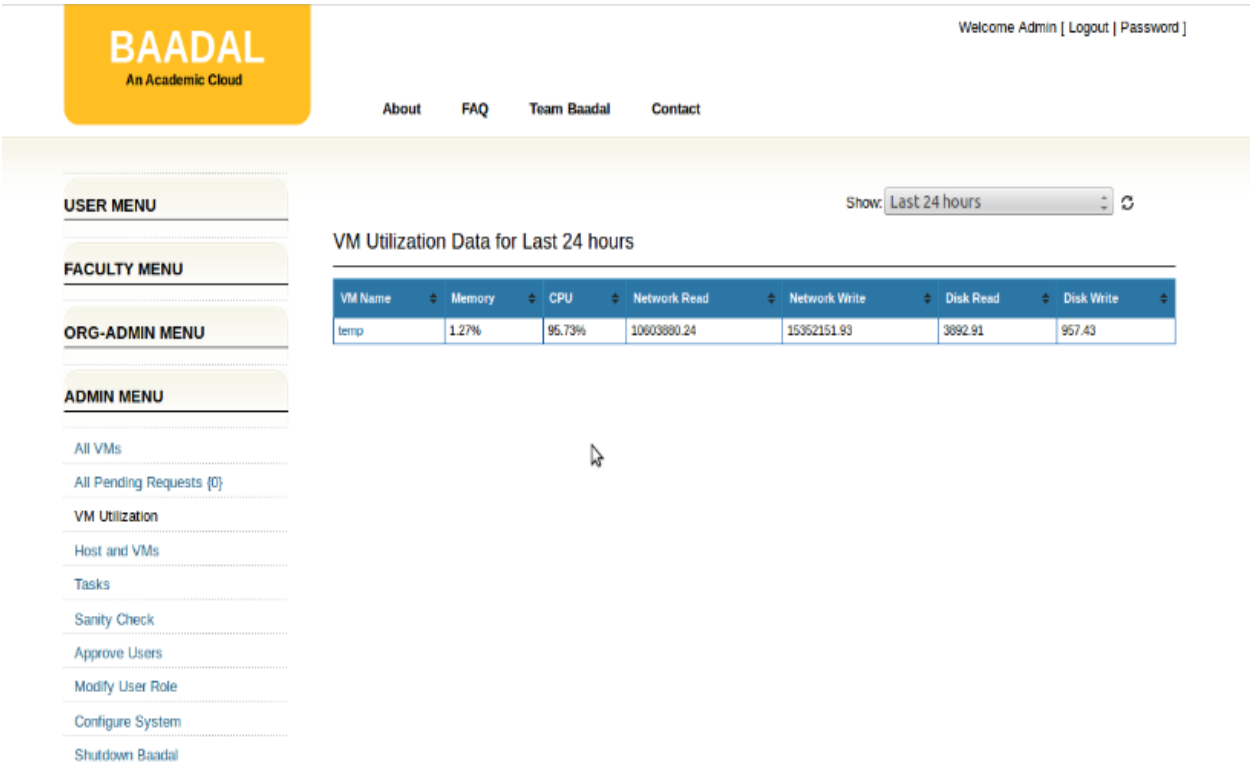
On clicking **ALL VM** functionality, it redirects to the page where admin can see all the VMs. Following should be the output.



Name : This is the name given to the VM
Owner : This is the owner name to whom VM is allocated
Organisation : This stands for the organisation to which user belongs
Host : IP address of the host machine in which VM resides
Private IP : This is the private IP to allocated to the VM
RAM : Memory allocated to the VM
vCPU : Virtual CPUs count allocated to the VM
Status : It shows whether the VM is running, Paused, Shutdown etc
Setting : This redirects the page to setting page. It will be explained in <FR>.

4. **Title : VM Utilization**

This shows the resource utilization of all the VMs of BAADAL. Following should be the output.

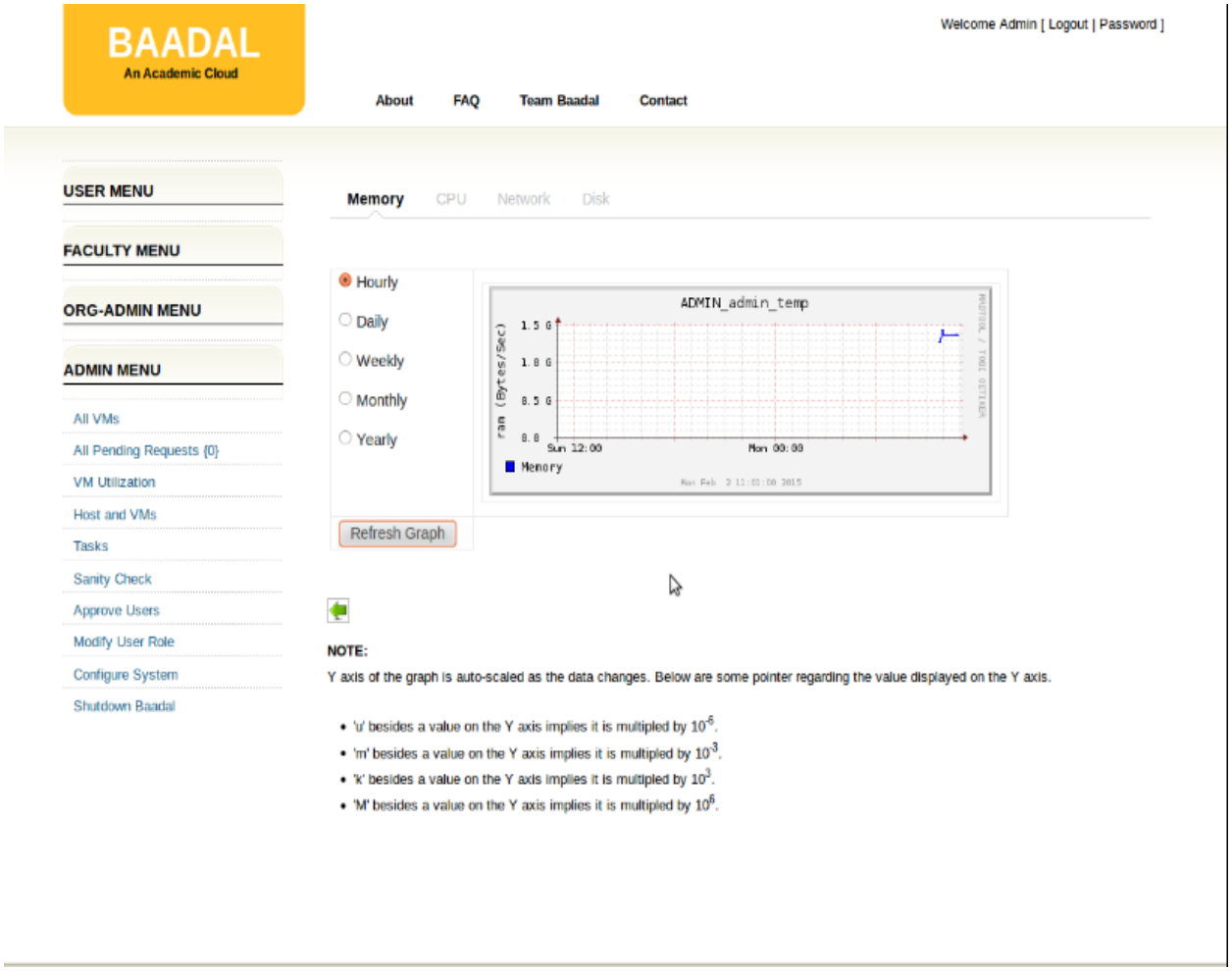


VM name : stands for name of the VM
Memory : Percentage usage of memory by the VM
CPU : Percentage usage of CPU by the VM
Network Read : Shows the bytes read through the network
Network Write : Shows the bytes write through the network
Disk Read : Shows the bytes read from the disk
Disk Write : Shows the bytes write to the disk

5. **Title : VM Performance Graph**

VM utilization functionality shows the resource utilization of the VM. On clicking the VM name in that utilization table it gets redirected to VM performance page. Where it shows the resource utilization in the form of graph.

It has 4 tabs, Memory, CPU, Network, Disk. Each tab shows the performance graph of that resource utilized by VM. This graph can be perceived on Hourly, Daily, Weekly, Monthly, Yearly basis. Following should be the output of VM performance page.



Note : Above page is the output for Memory utilization. Same should be the output for other tabs also

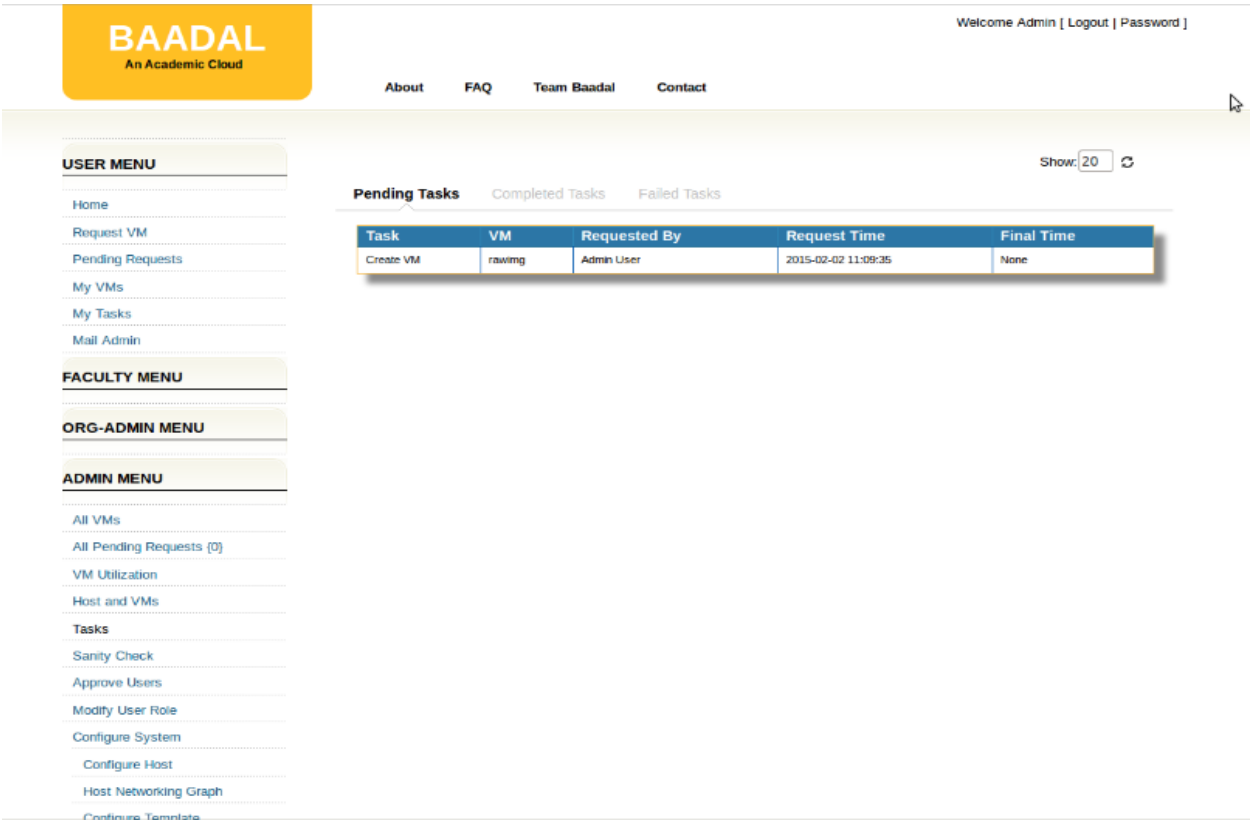
6. **Title : Tasks (Inside Configure System)**

Tasks shows the task assigned to Admin. It has 3 taskbars. They are as follows:

Pending Tasks :

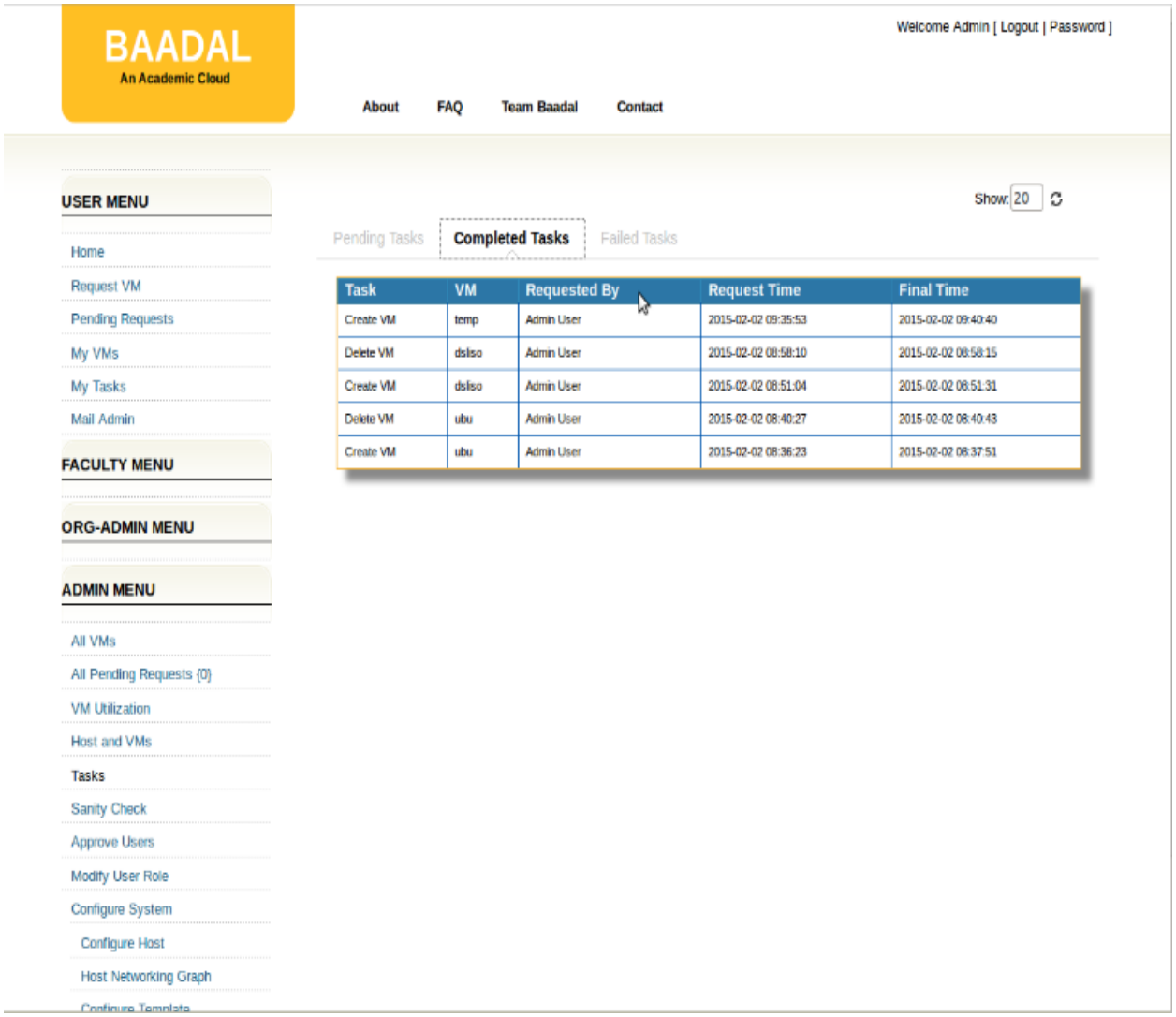
It shows the tasks on go, approved by ADMIN. Output of pending tasks should be as follows. This shows Task [“Create VM”, “Delete VM”, “Snapshot VM”, “Suspend VM”, “Shutdown VM”, “Pause VM”, “Start VM”].

VM (Name), Requested By (User Name), Requested Time shows the time at which the task was requested by the user, and final time shows the time at which task is completed.



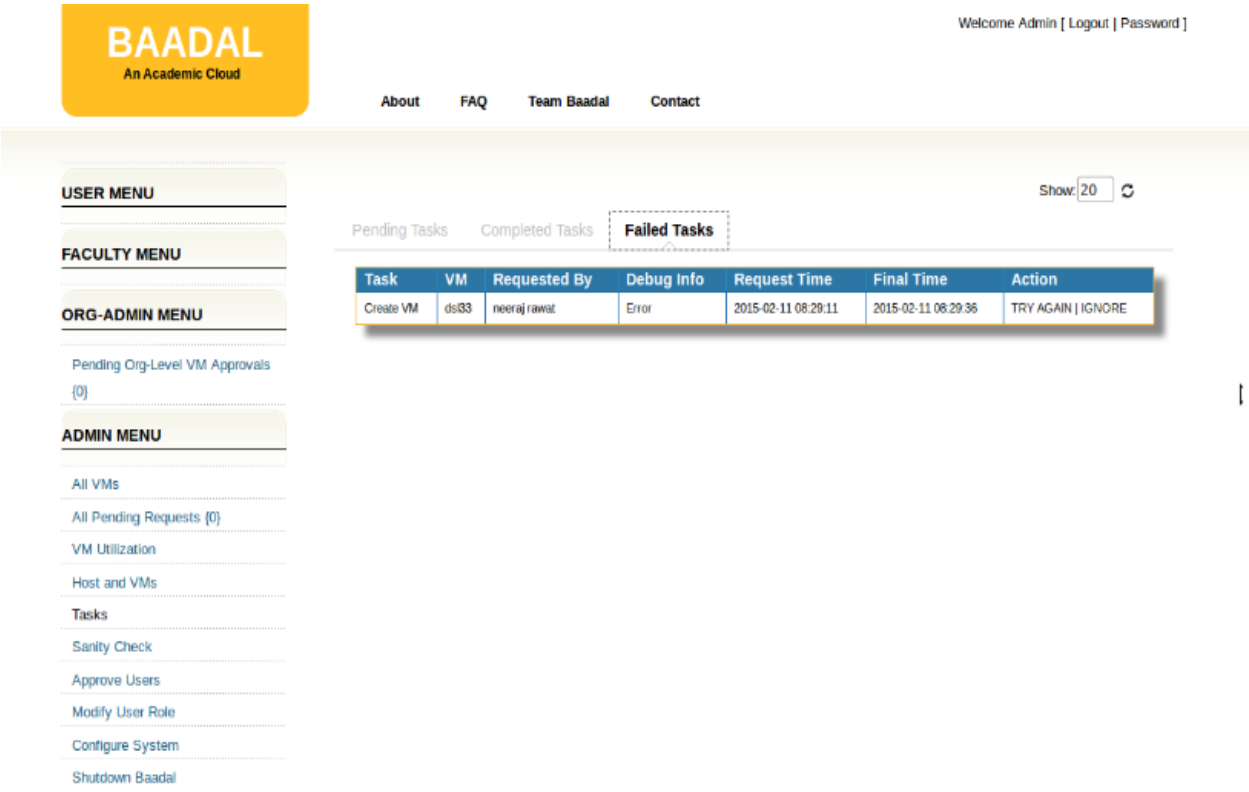
Complete Tasks :

It shows the tasks which were approved by ADMIN and they got completed.



Failed Task :

In case a task got failed then it would be shown in failed task tab as below:



Task : Type of task. In this case it is “Create VM”

VM : This is the name of the VM

Requested By : Name of the user who requested for the VM

Debug Info : “Error” is a hyperlink. It redirects to the page which contains the Error Debug Trace.

Request Time : Time at which VM was requested.

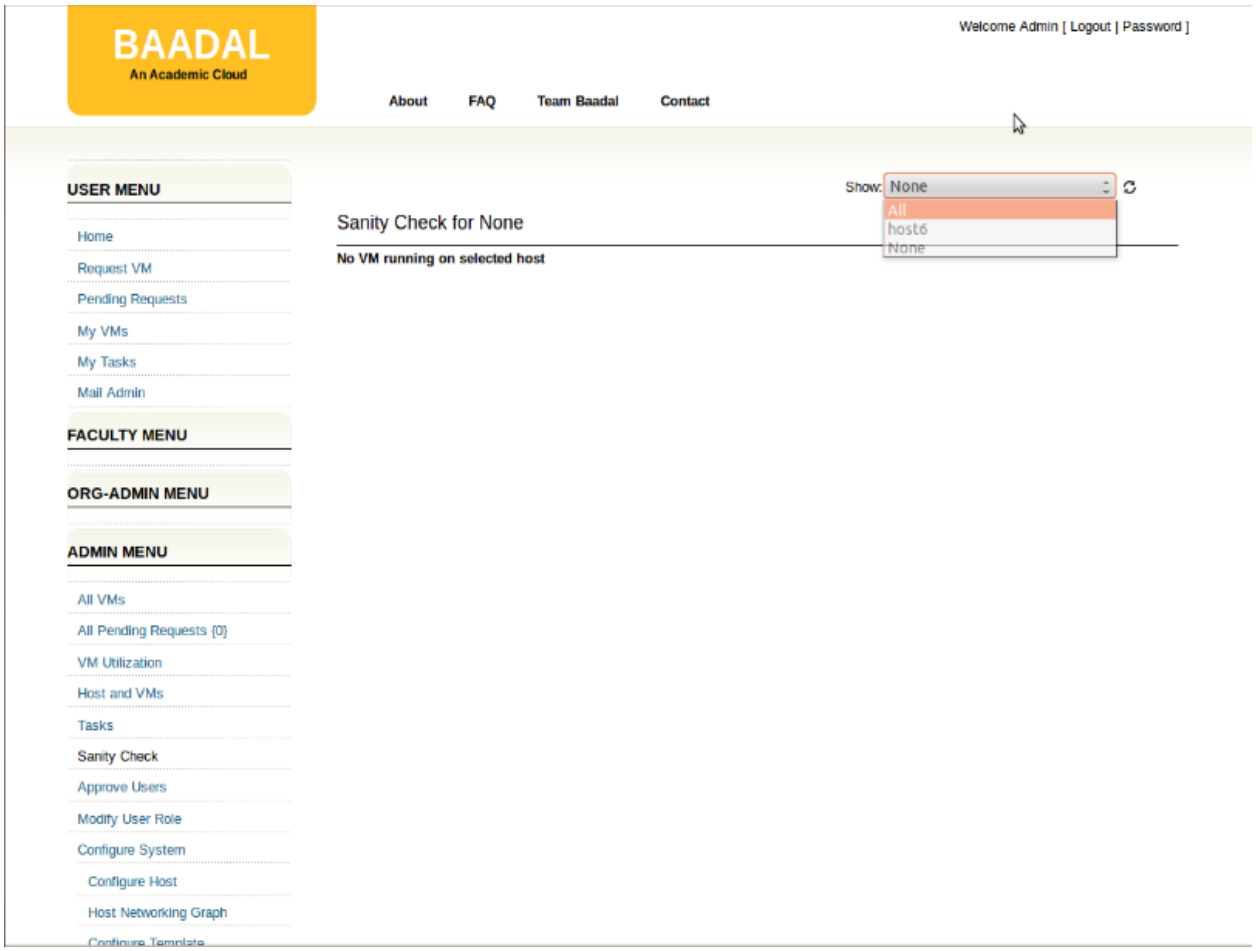
Final Time : Time at which task got failed.

Action : There are two action under this:

- Try Again : Selecting this, will again start the task.
- Ignore : Selecting this, will ignore/reject the task and it will be removed from the task queue

7. **Title : Sanity Check**

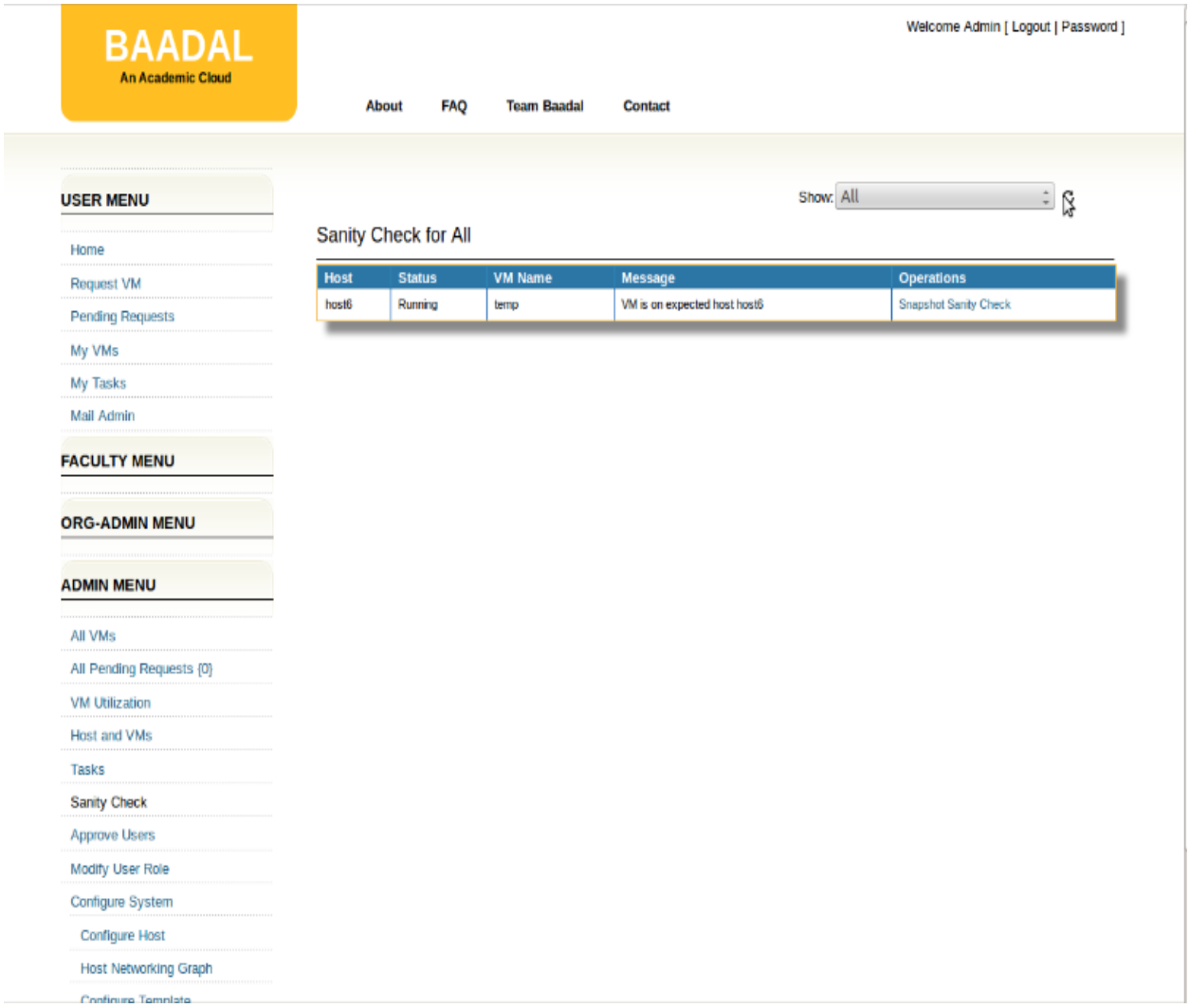
On clicking this functionality, Admin can perform sanity check on host. Following should be the output:



Show dropdown box set to None by default. This should consists of all the hosts name plus “All” in it.

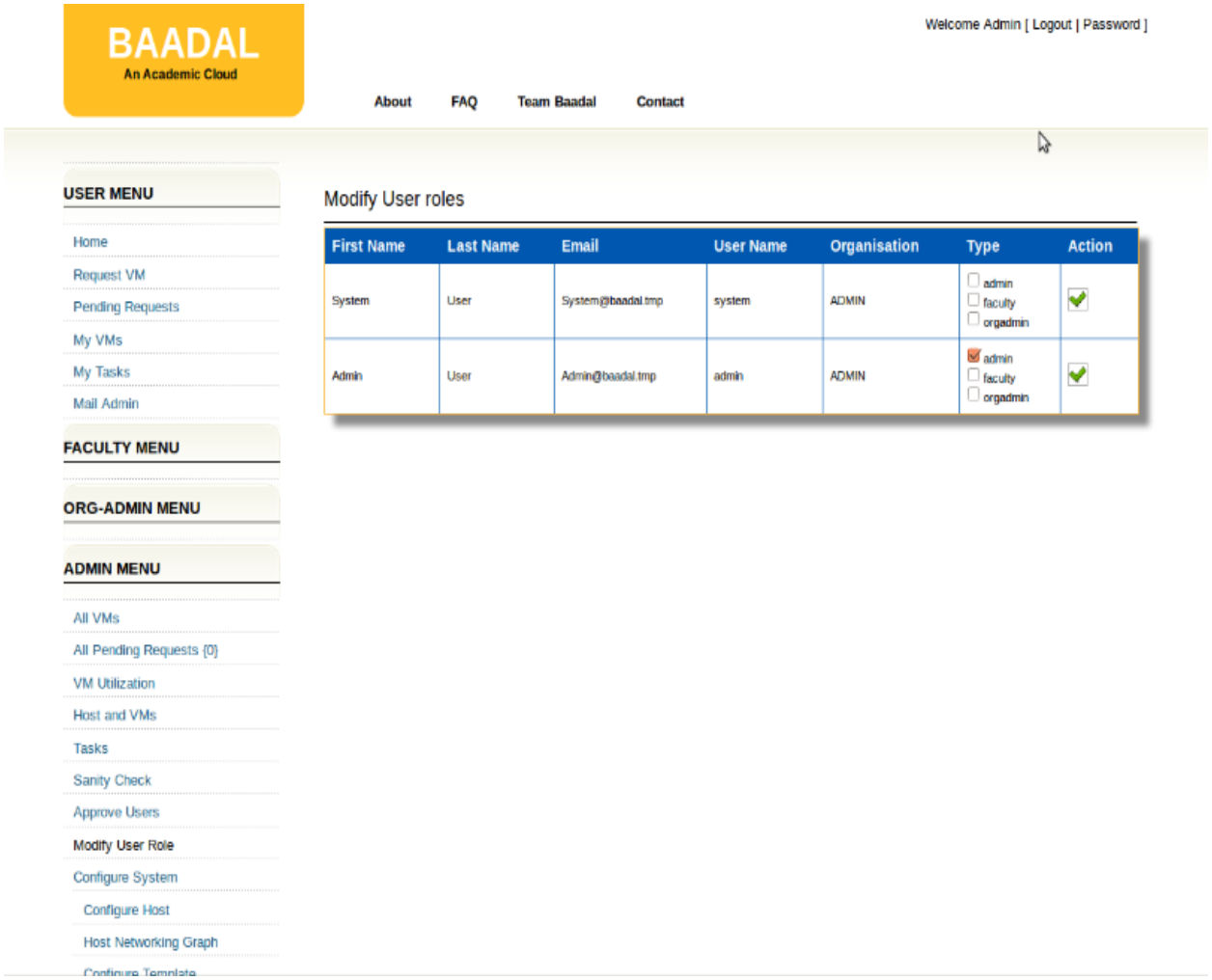
Admin can see select a host or All option to perform the sanity

On selecting a host or All option in Show dropdown following should be the output:


BAADAL
An Academic Cloud
Welcome Admin [Logout | Password]
About
FAQ
Team Baadal
Contact
USER MENU
Home
Request VM
Pending Requests
My VMs
My Tasks
Mail Admin
FACULTY MENU
ORG-ADMIN MENU
ADMIN MENU
All VMs
All Pending Requests (0)
VM Utilization
Host and VMs
Tasks
Sanity Check
Approve Users
Modify User Role
Configure System
Configure Host
Host Networking Graph
Configure Template
Show: All
Sanity Check for All
Host
Status
VM Name
Message
Operations
host6
Running
temp
VM is on expected host host6
Snapshot Sanity Check

8. **Title : Modify User Role**

This utility lists all type of existing users of BAADAL. Using this functionality ADMIN can modify the role of any user, whether its student, faculty, orgadmin or admin itself. On clicking this functionality ADMIN should see the following page:



First Name : Stands for the first name of the User.

Last Name : Stands for the last name of the User

Email : It shows the email id of the user

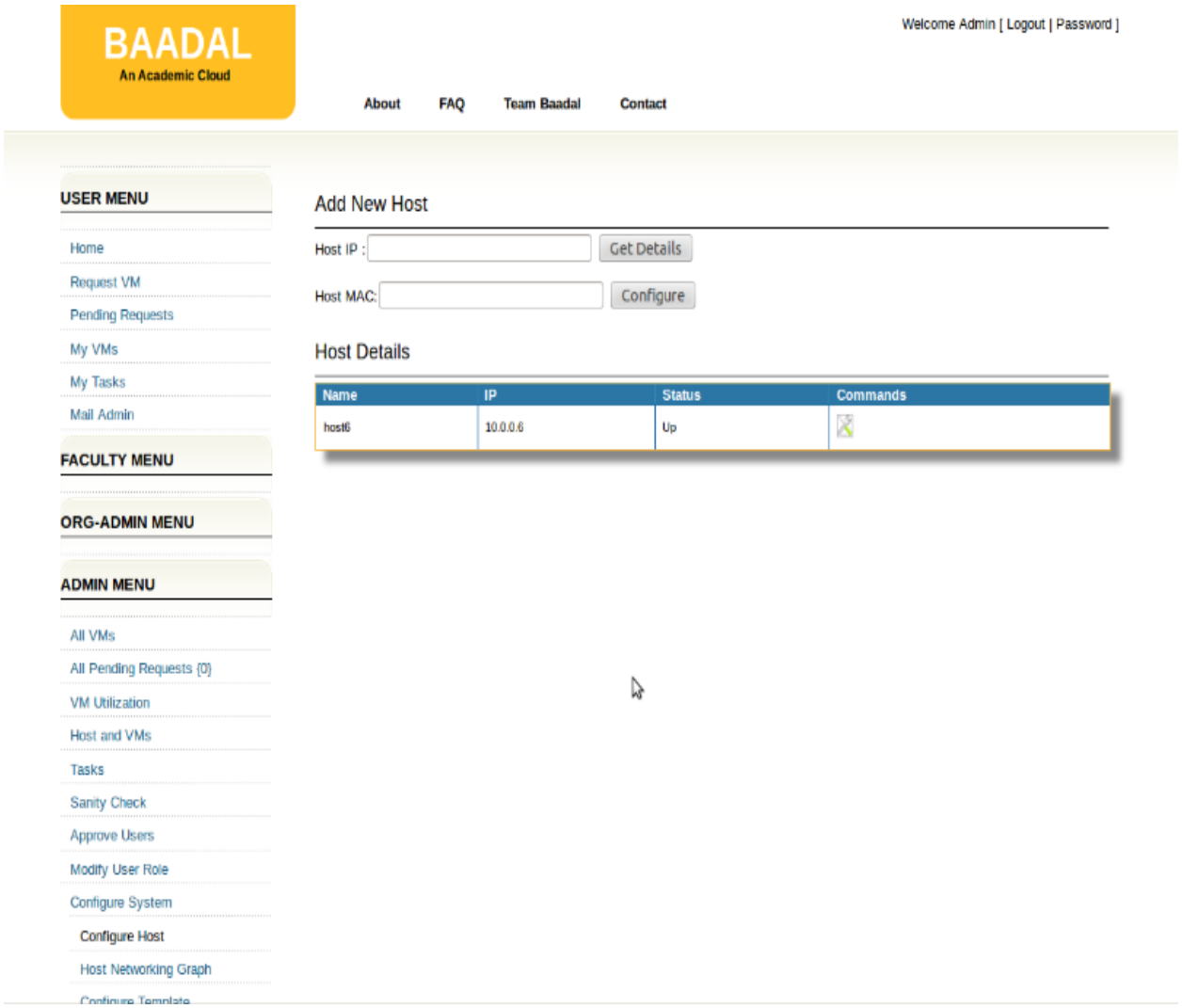
User Name : Stands for username of the USER

Type : This is the main functionality of this utility. Admin can modify the role of the user by ticking the checkboxes of the roles. A user can have multiple roles.

Action : By clicking the RIGHT sign, the modification of the role of the corresponding user will be committed.

9. **Title : Configure Host(Inside Configure System)**

On clicking this functionality it redirects to the page where admin can see the list of all the hosts connected to the controller.



Name : This is the name of the HOST.

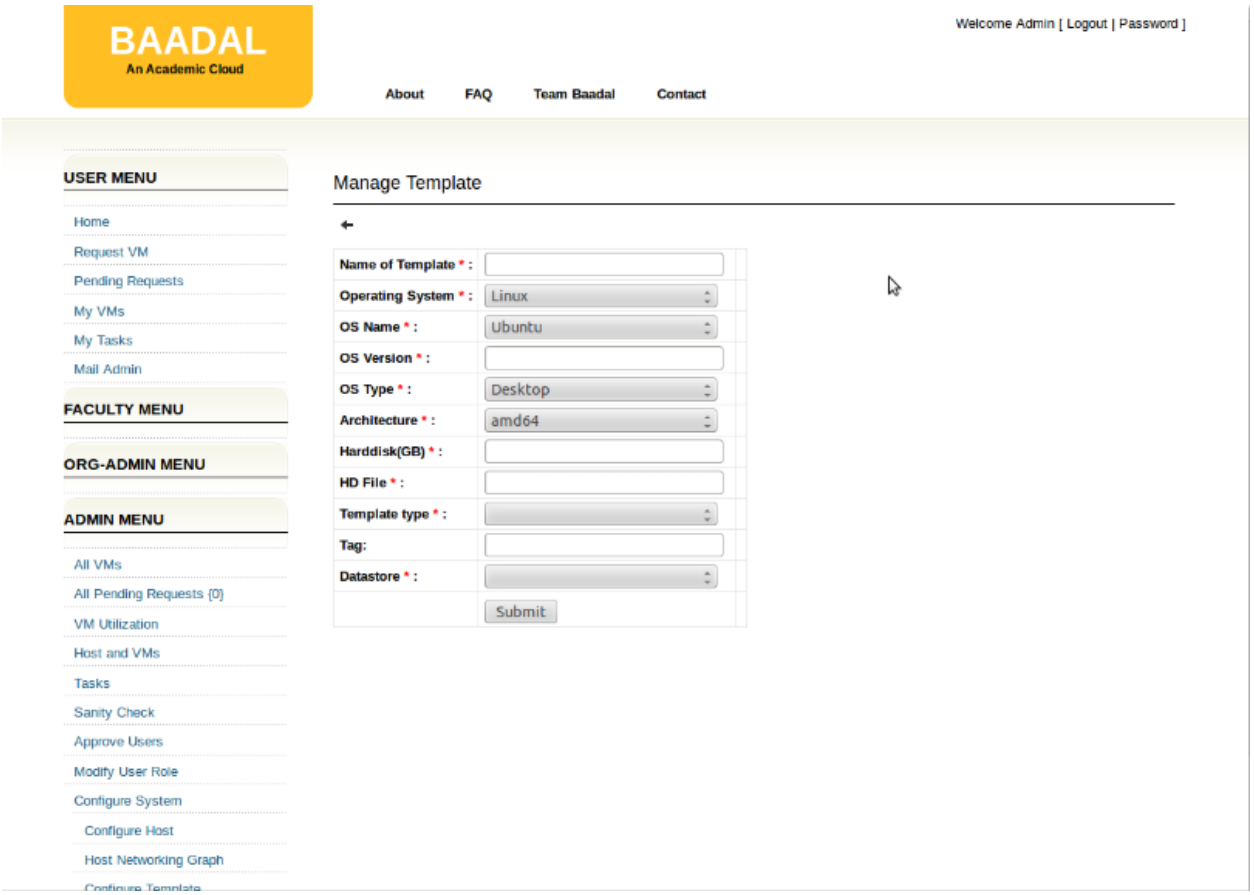
IP : IP address of the host

Status : “Up” means host is pingable

“Down” means host is shutdown

10. **Title : Add template (Inside Configure System)**

Here we create a VM template Id. The VM template is already saved in the FILER. With this page we create an ID to that VM template of filer.



Name of Template : <This is the name given to the template. It can be any name. for example, **test>**

Operating System : <This is dropdown list. Three values are defined. Linux, Windows, MAC**>**

OS Name : <This is a dropdown list. values are Ubuntu**>**

OS Version : < Version of the OS. We can any value to it. Genuine or anything because this has no effect on the operation**>**

OS Type: <Drop Down list. Desktop and Server**>**

Architecture : <Drop down list. for example, amd64 (64 bits), x86_64, i386(32 bits)**>**

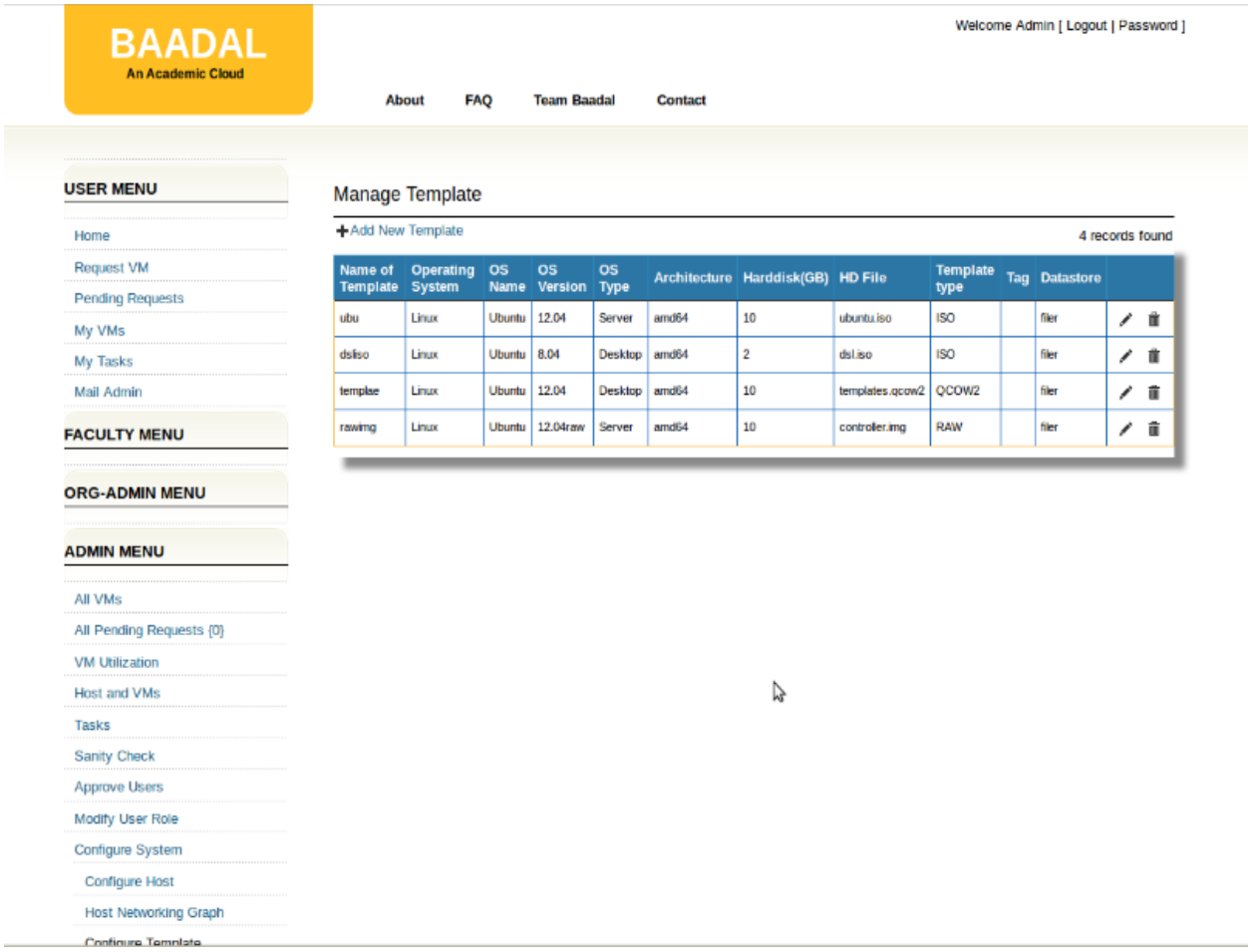
HD File : <Name of the template that is kept in FILER. for example, **template.qcow2. This should be the bootable file>**

Template type : <This is a dropdown list. This is basically the entension of the template. Values consist in it are, QCOW2, RAW, ISO.**>**

Tag : <Optional>

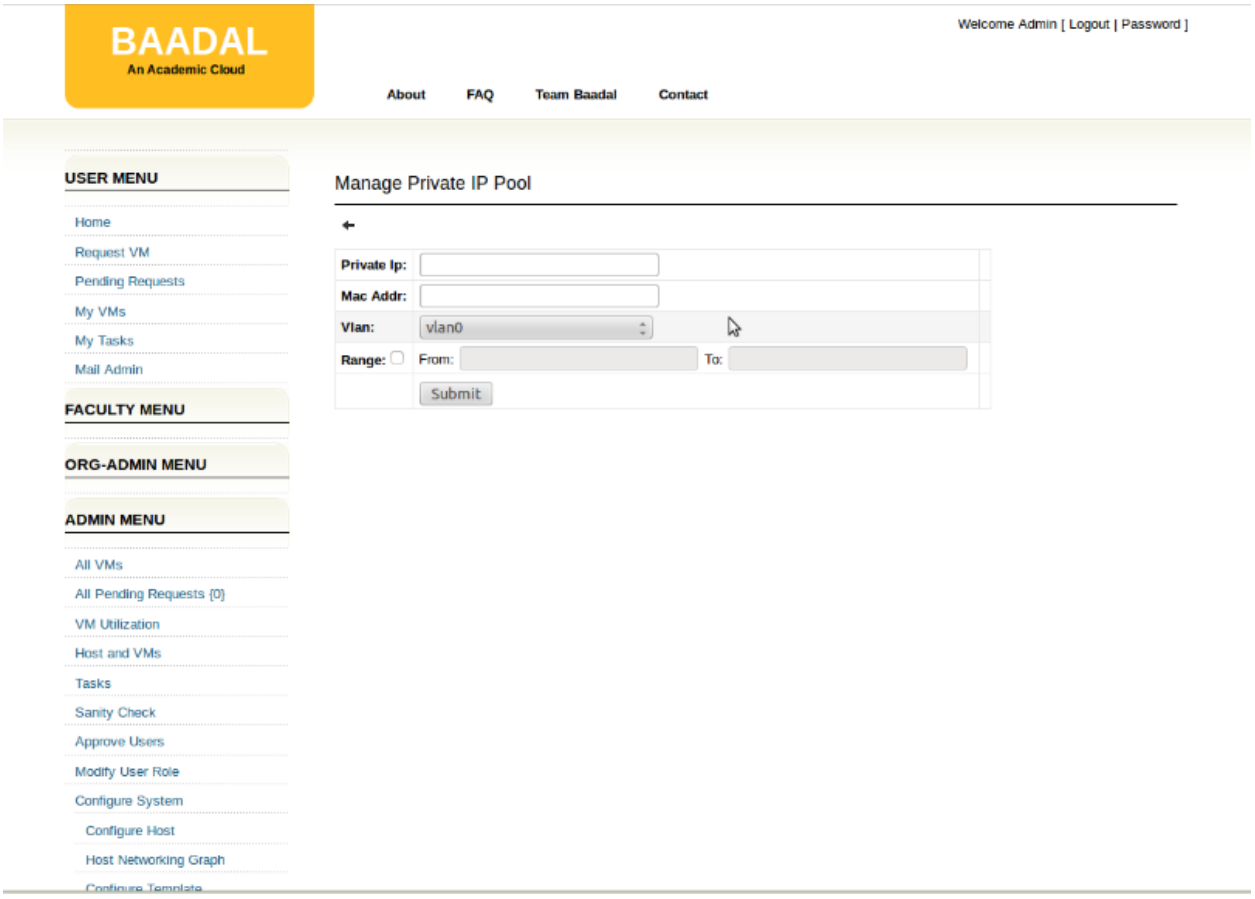
Datastore : <This is the datastore name which is added earlier**>**

Following should be the output of templates



11. **Title : Add Private IP (Inside Configure System)**

Private IPs pool will be defined here. IPs from this pool will be allocated to the VMs which will be created later on. If this pool is empty or fully used up then newly created VM will not get the IP and VM creation will be unsuccessful.



Private Ip : <Here the private Ip will be given. This private must match **VLAN** value. (This will be clarifies later). for example, **10.0.0.6 (It should be correspond to vlan0)>**

Mac Addr : <Here the MAC Address will be given manually. for example, **52:52:00:01:15:06 >**

Vlan : <This is a drop down list, where 5 values of vlan are defined. for any vlan value there will be a netmask of 255.255.0.0. for example,

vlan0 : 10.0.0.0 to 10.0.0.255

vlan1 : 10.0.1.0 to 10.0.1.255

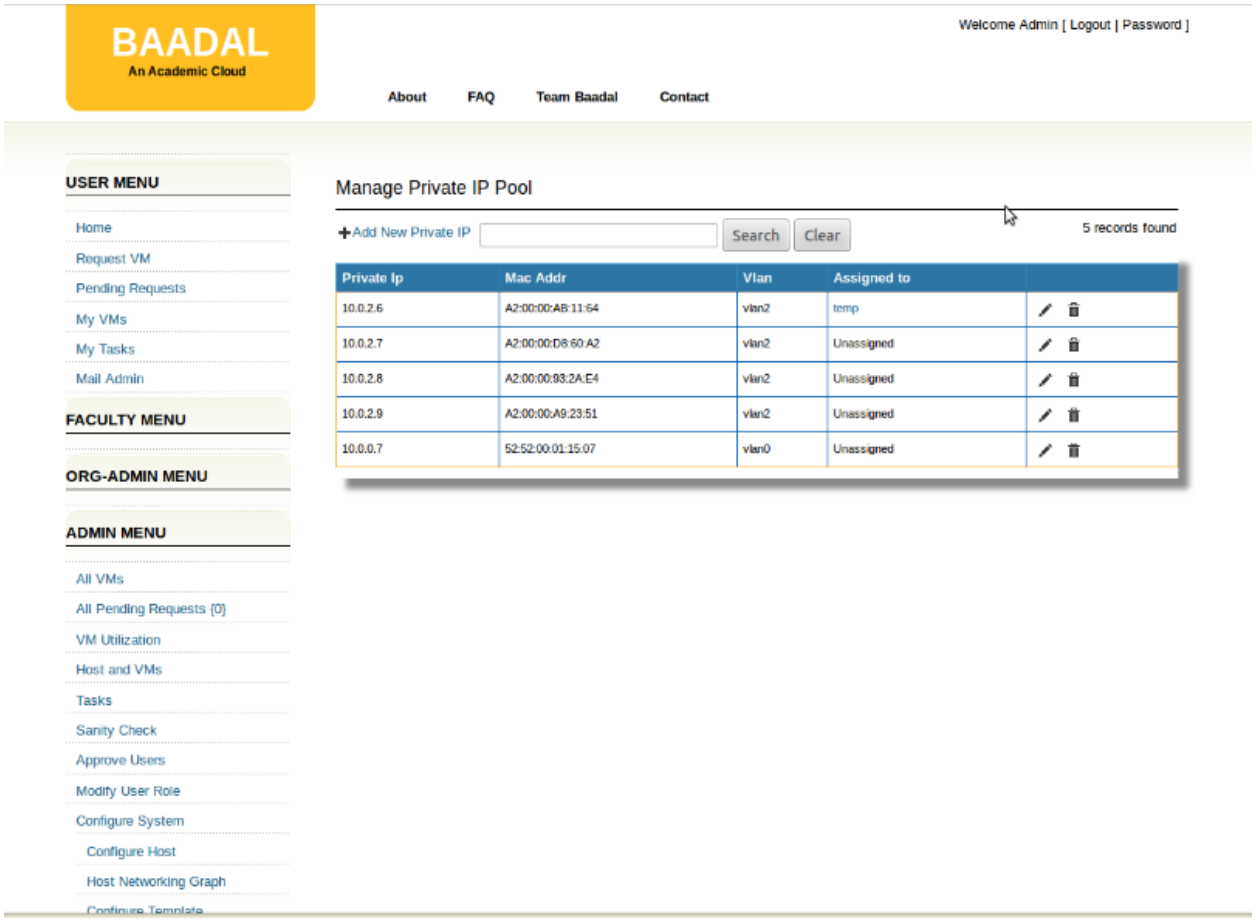
vlan2 : 10.0.2.0 to 10.0.2.255

vlan3 : 10.0.3.0 to 10.0.3.255

vlan4 : 10.0.4.0 to 10.0.4.255**>**

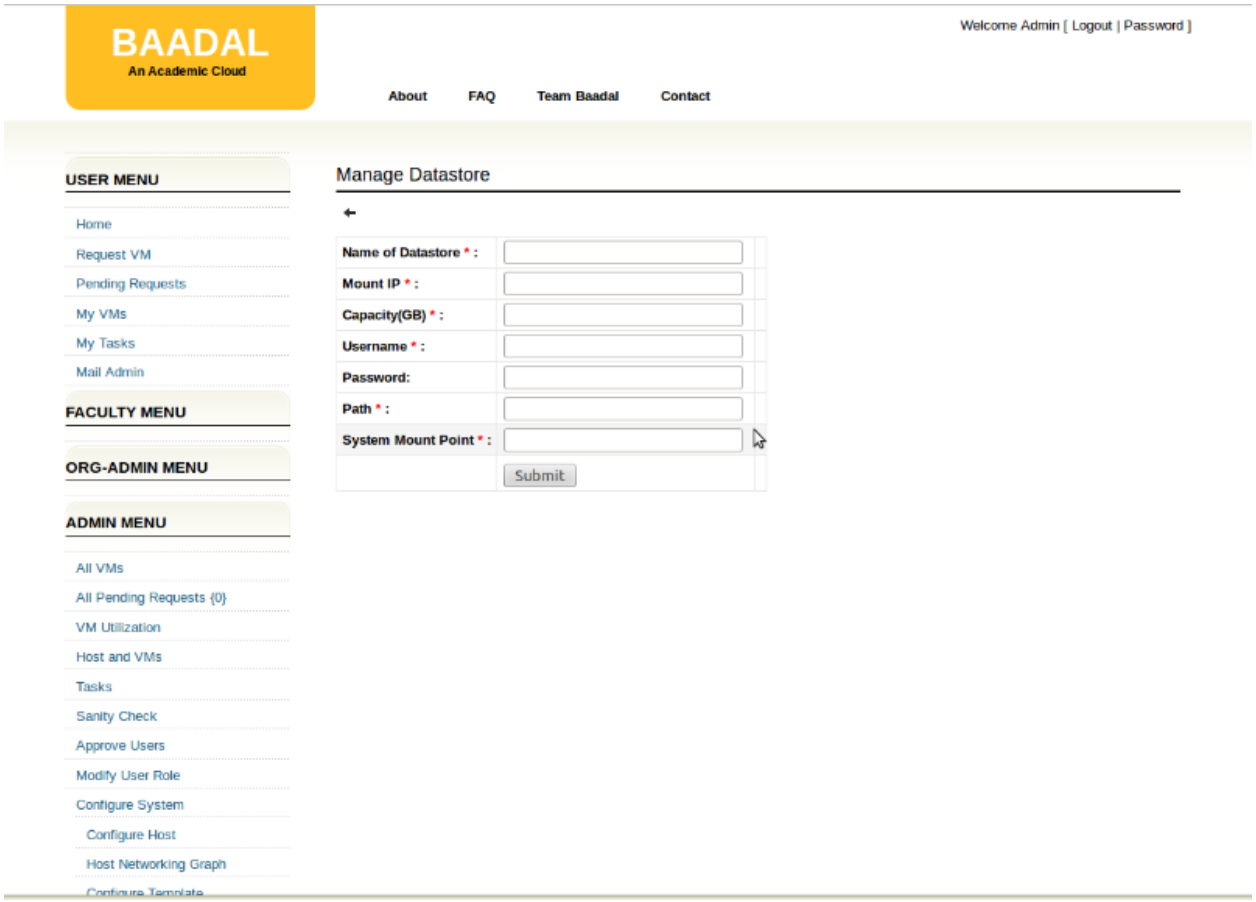
Range : <This is checkbox. If this is selected then above 2 will be disabled. Then we can provide a vlan value and then provide a range of IP address correspond to that vlan. On clicking submit button, a pool of IPs will be created between the range given, with start and end IP values inclusive.for example, **vlan2 , range start-10.0.2.6 and end-10.0.2.9>**

The output of configure private ip pool should be below:



12. **Title : Add datastore (Inside Configure System)**

Datastore is the storage area where all the VM templates, VM images, RRD files, VM history resides. Basically this lies in FILER which is mounted to controller and hosts



Form below the **Manage Datastore.**

Name of Datastore : <Any name can be given to the datastore. This name is the identification of the datastore. for example, **filer>**

Mount IP : <This is the IP Address of the FILER. for example, **10.0.0.1>**

Capacity(GB) : <This is the amount of space we want to allocate to the datastore which is going to be added. for example, **100>**

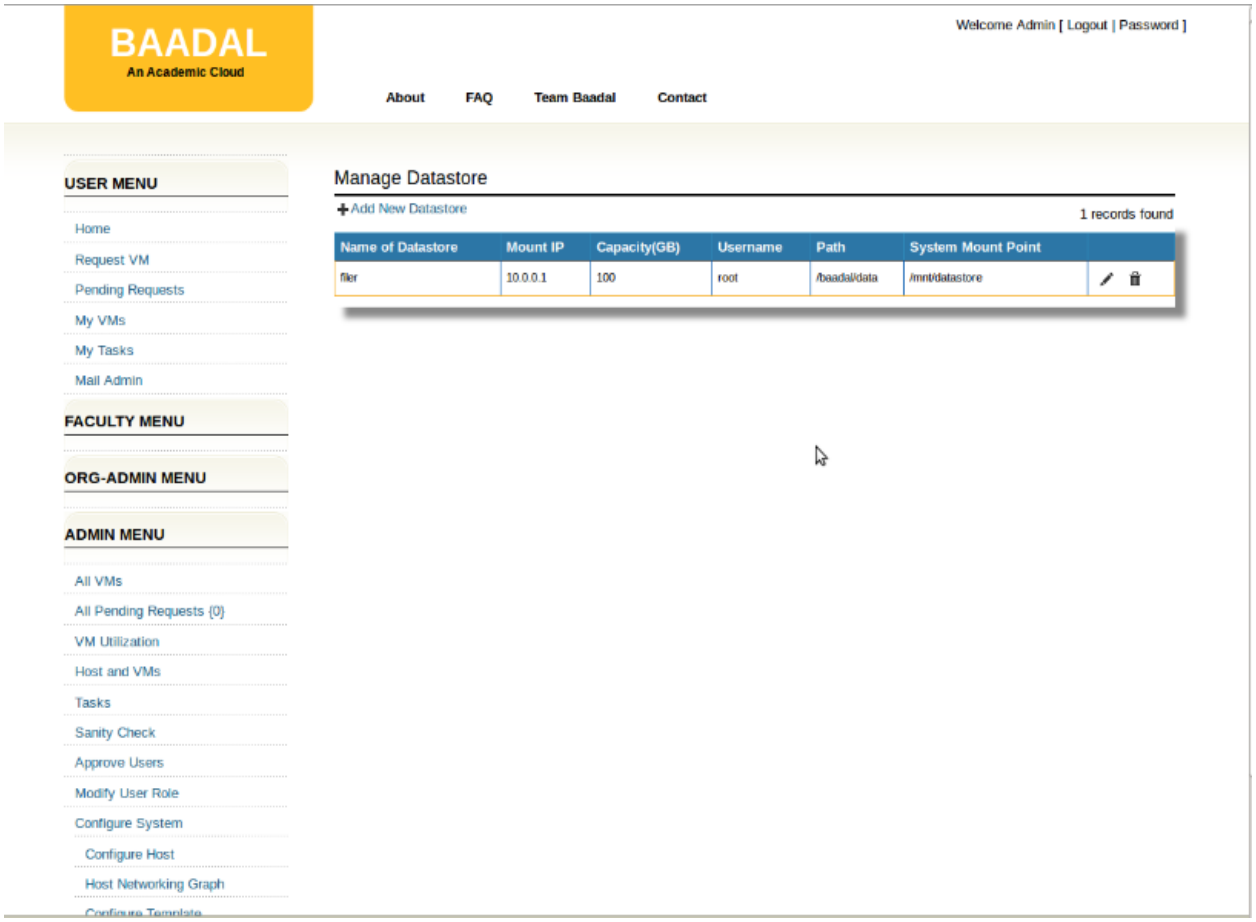
Username : <Username of the FILER system, which will be “root” always**>**

Password : <Password of the root**>**

Path : <This is the path where VM related files has been kept. This path resides in FILER. for example, **“/baadal/data”>**

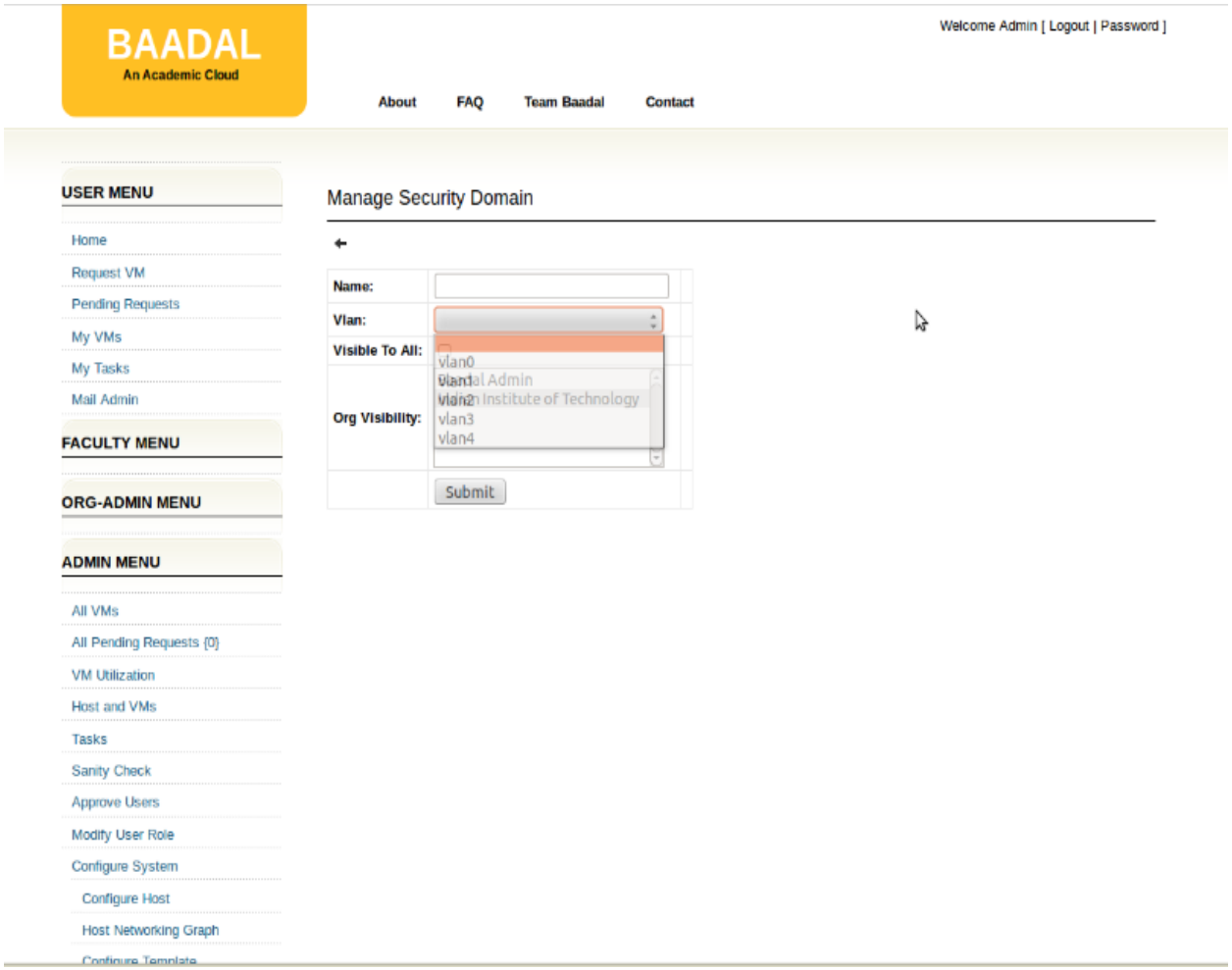
System Mount Point : <This is the path where the “Path” above path will be mounted. System mount point is same in controller and hosts. for example, **“/mnt/datastore”>**

Following should be the output of the datastore page, when any datastore is added to baadal:



13. **Title : Add new security domain**

Using this functionality Admin can create security domains. Following page should be the output on clicking “security domain” functionality.



Name: This stands for the name Admin want to give to the security domain which will be created.

Vlan : This is dropdown list. It has 5 values. [“vlan0 : 10.0.0.0”,”vlan1 : 10.0.1.0”,”vlan2 : 10.0.2.0”,”vlan3 : 10.0.3.0”,”vlan4 : 10.0.4.0”].

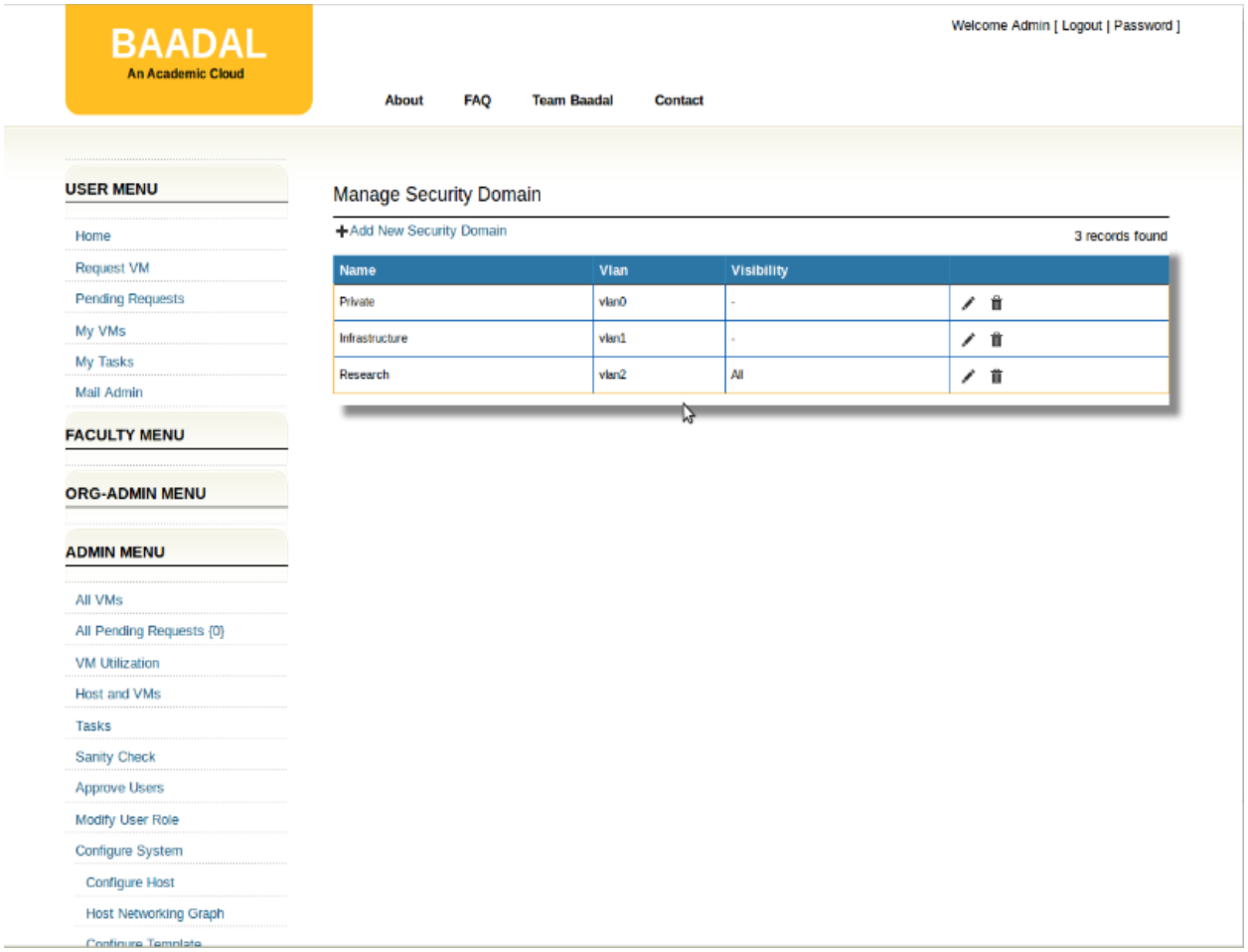
Visibility To All : This is a checkbox component. Setting this will make this security domain visible to all types of users else it will be visible to only ADMIN.

Org Visibility : It has two values. Baadal Admin, Indian Institute of Technology.

Note : "Visibility to All" checkbox means visibility to all users at the time of "Requesting VM" to opt under which security domain user wants his/her VM.

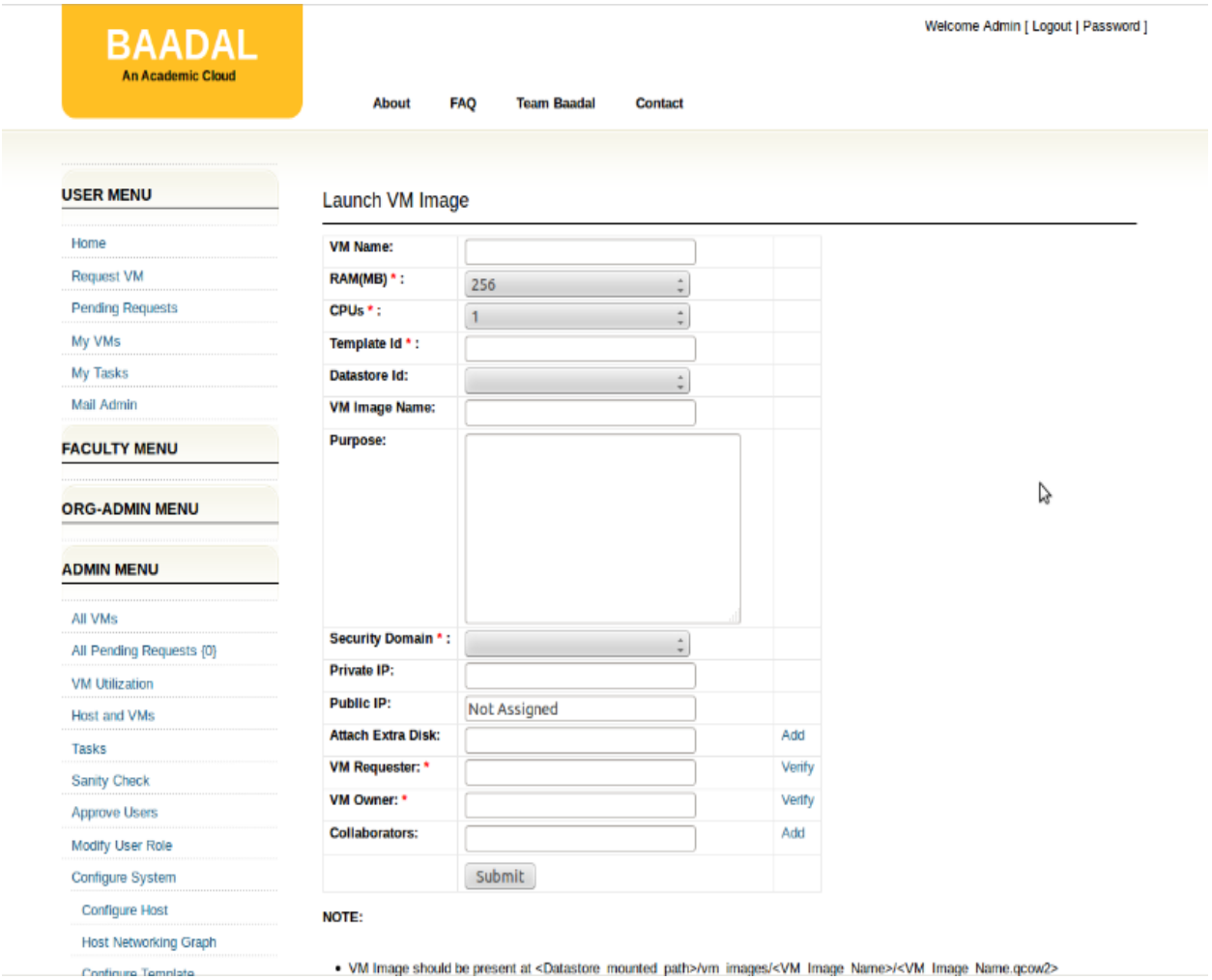
Security domains

Following should be the output:



14. **Title : Launch VM (Under configure system)**

Using this functionality, Admin can launch a VM. Following page should get open on using this functionality



VM Name : Name of the VM

RAM : Dropdown list, value in it are in MBytes.

CPUs : This stands for count of vCPUs for the VM

Template Id : This is the “id” corresponding to the template which is added in the template

Datastore Id : “id” of the datastore where VM Disk will be kept

VM Image Name : Name given to the VM image. Make sure that **“/mnt/datastore/vm_image/<this name>/<this name>.qcow2”** should exists.

Purpose : Optional

Security domain : Select between Research, Infrastructure, Private and any other security domains added to baadal

Private Ip : Manually providing private Ip to the VM. (Keep in mind, this Ip must belong to the chosen security domain). It can be left empty, then random Private Ip from Private Ip pool will be given to this VM.

Public Ip : Provide Public Ip to the VM

Add extra disk : Any extra disk admin want to attach to the VM

VM requester : Name of the VM requester. Then verify whether user with this name exists or not

VM owner : Owner of the VM. Then verify whether user with this name exists or not

Collaborators : Add multiple collaborators to the VM

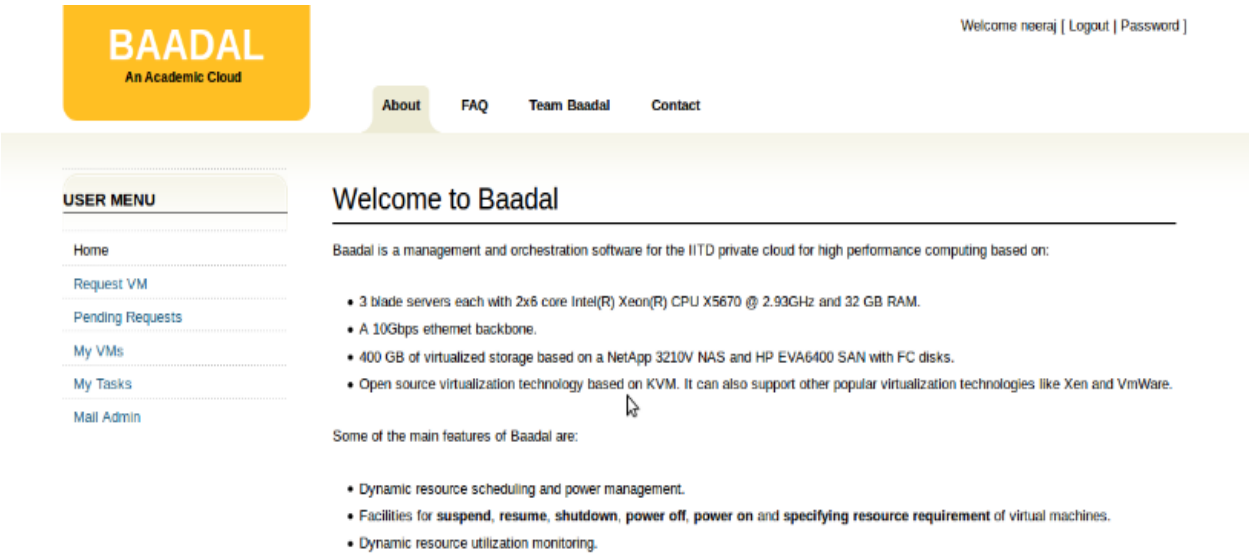
Users FR : Here users include student, faculty and orgadmin

Title : Types of User

In baadal orchestration there are 4 types of users, vis a vis

1. Student
2. Faculty
3. Organization Admin (Ord Admin)
4. Admin

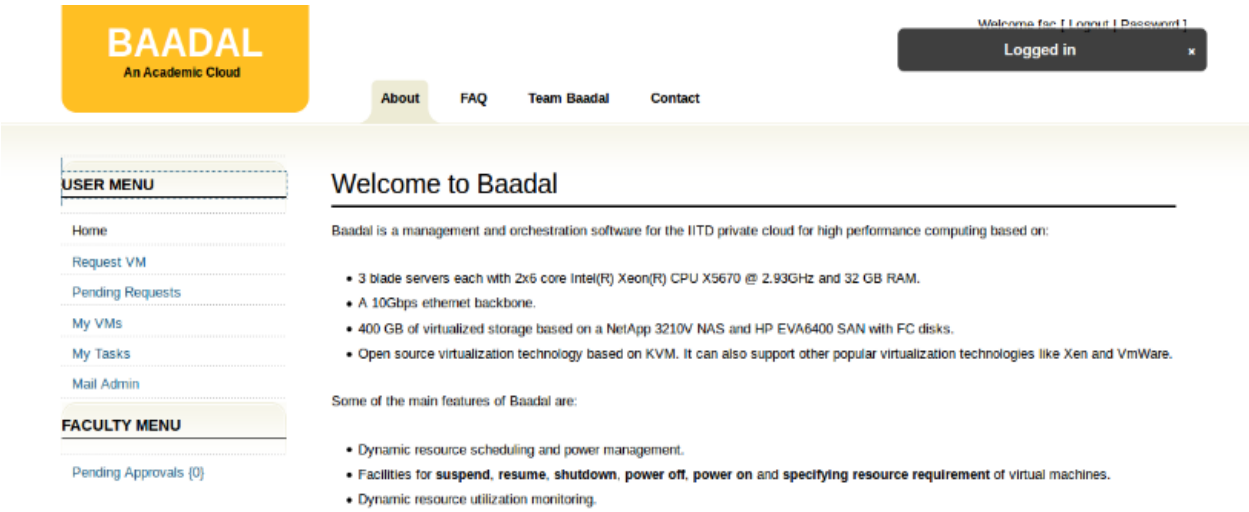
All these users can request for VM. The basic type of user is Student. How it looks like when a student logs in into baadal. It can be seen in the below output:



In the LEFT hand side we can see the functionality given to basic user. We will talk about it in detail.

Faculty as a user:

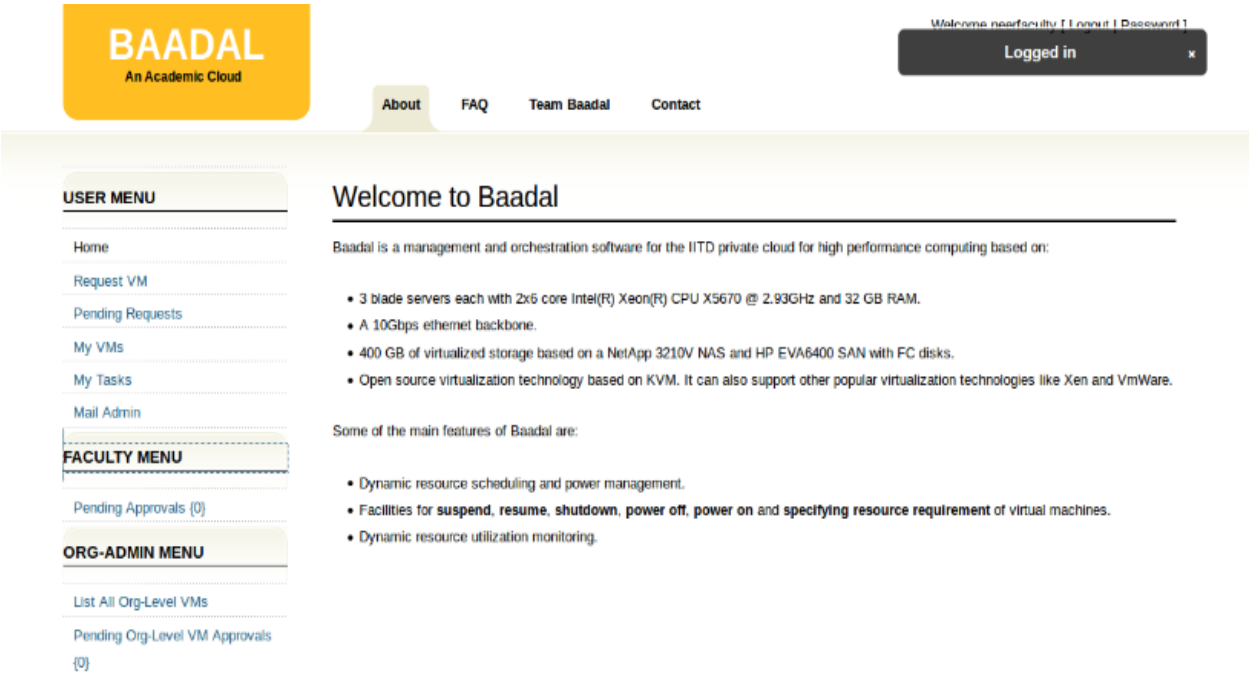
Faculty is also a user with some functionality of being a faculty, over student. We can see in the below output :



As we can see on the LEFT side of the above output. Faculty has all the functionality of a basic user plus he has the functionality of getting the request from basic user under him

Org Admin as a user:

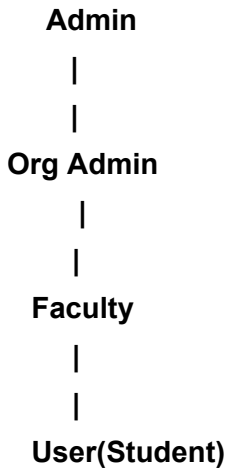
It also has user functionality, faculty. It looks like below:



Org Admin has two functionalities

1. **List all Org Level VMs :** It shows all the VMs of Org Admin level.
2. **Pending Org Level VM Approvals :** It shows the request which comes from faculty.

Hierarchy level of all type of users



Request Hierarchy :

User to faculty

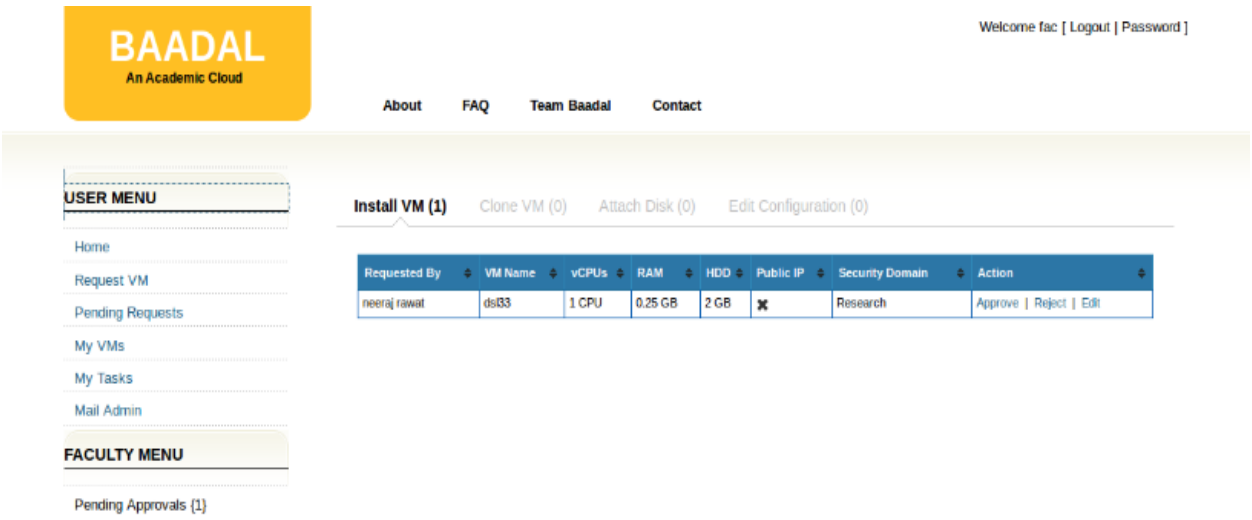
faculty to oeg admin (if faculty approves)

org admin to admin(if oeg admin approves)

Then admin can approve or reject the request

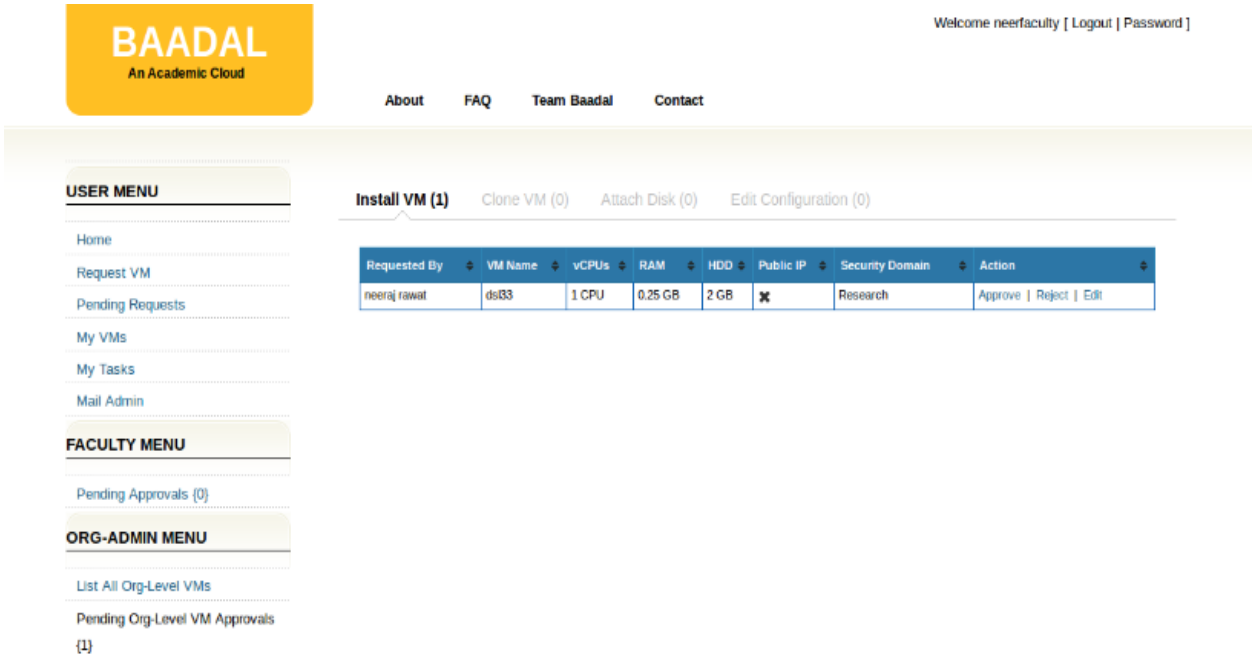
1. Title : Request hierarchy

When a basic user requests for a VM it goes to faculty and it looks like this at faculty level.



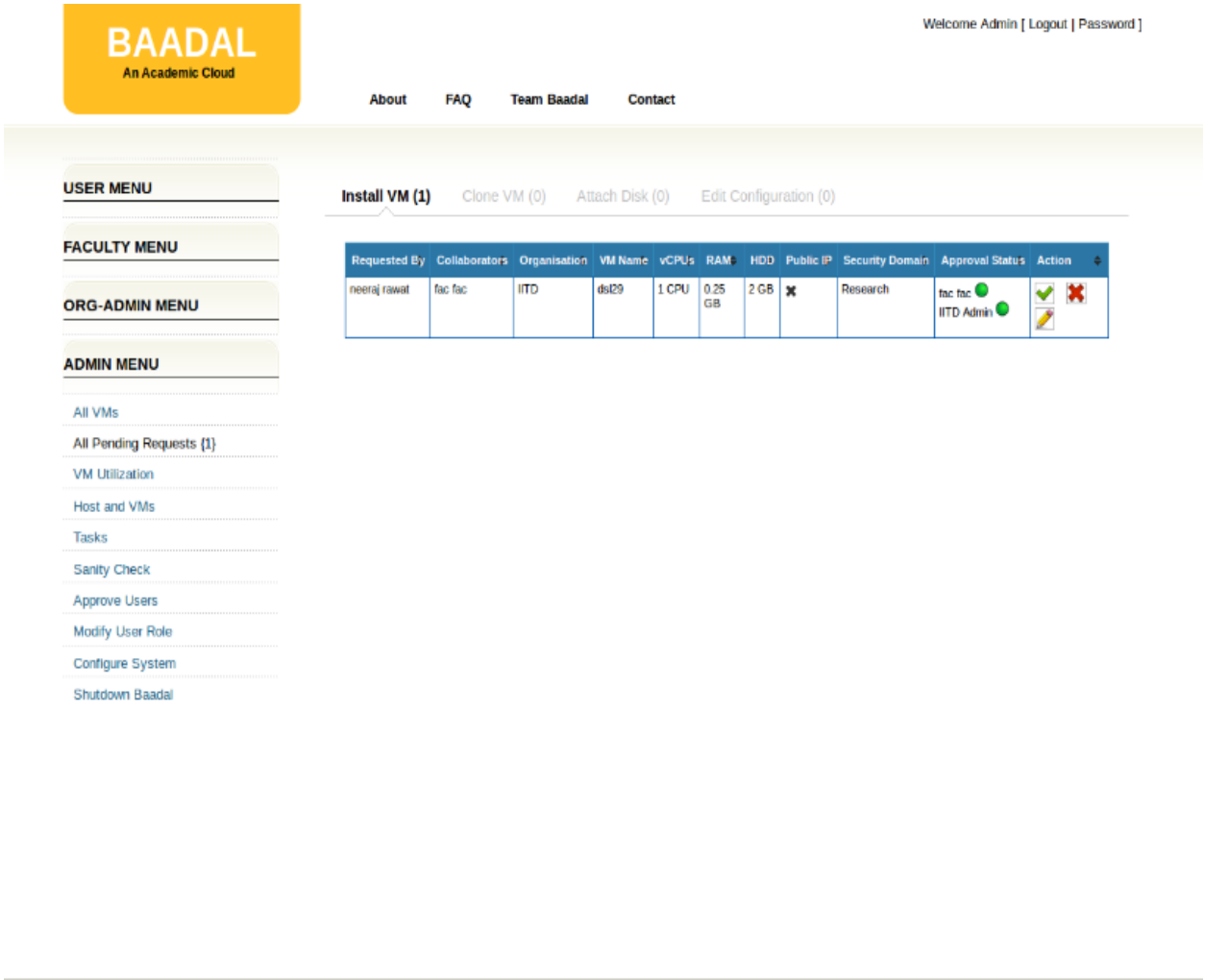
Here the faculty can approve, reject or edit the request.

- If it rejects the request then request will be removed from the request queue.
- Edit request will be discussed below
- It it Approve the request then it goes to Org Admin



Here the org admin can approve, reject or edit the request.

- If it rejects the request then request will be removed from the request queue.
- Edit request will be discussed below
- It it Approve the request then it goes to Admin as below



Here the admin when approves the request, it gets placed into task queue.
Under “Action:

 Approved request

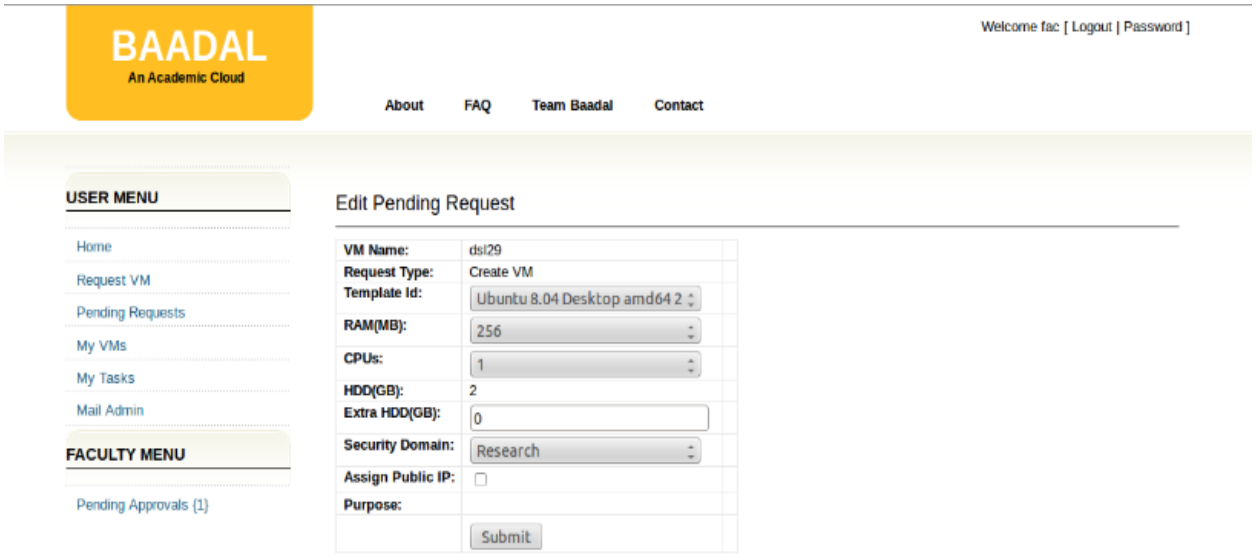
 Reject request

 Edit request

Edit Request:

Faculty , orgadmin and Admin . These all have this functionality to edit the request which comes from the user below in the hierarchy.

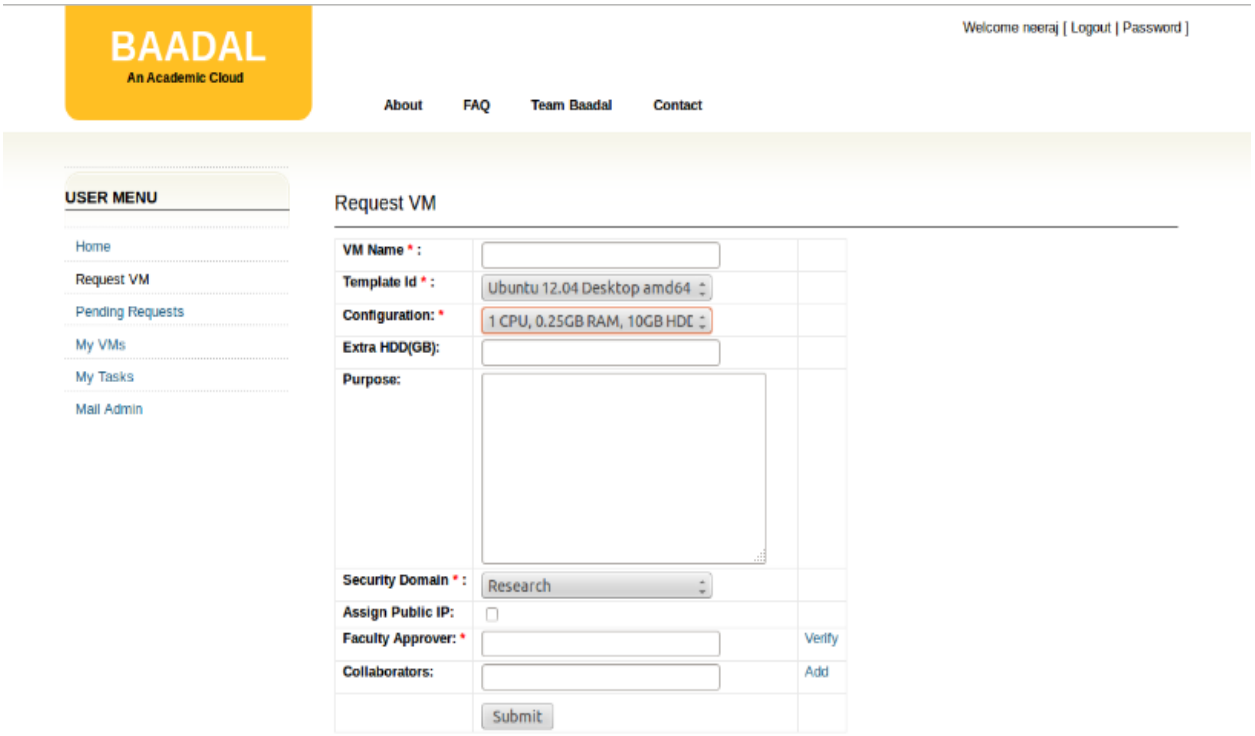
Edit request page looks like below



Here the resources of the VM can be edited by the faculty, orgadmin or Admin.

2. Title : Request VM

A user can request for a VM to the user sitting above him/her. A user can be anyone (student, faculty, org admin or admin himself) . Below is the output for requesting a VM.

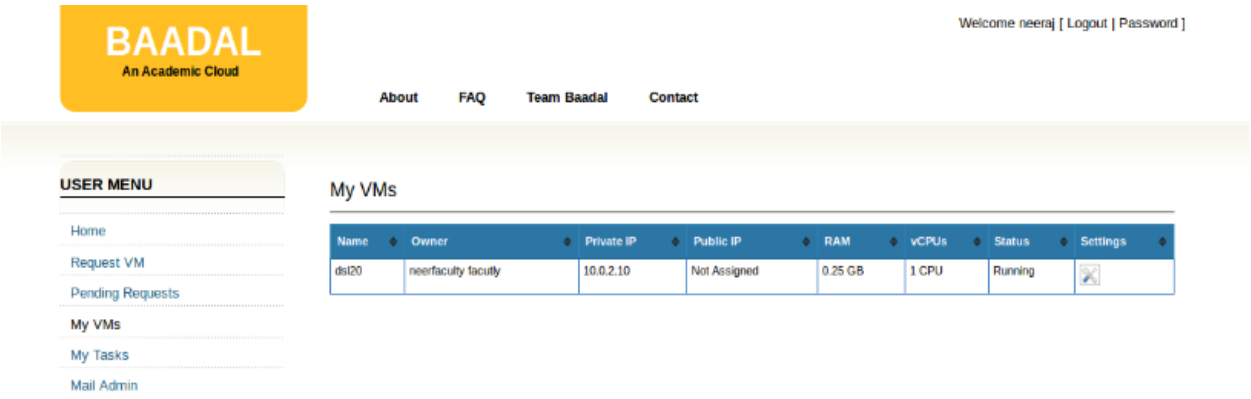


VM Name : The name for the VM user want to give. It should be unique from the VM names which are already requested or running.
Template Id : This belongs to the type of template which the user want to install for its VM.
Configuration : This contains predefined combinations of { CPU(count of vCPU), RAM (in Gb), HDD(in Gb) }. User can select anyone from it for his/her VM
Extra HDD : User can ask for extra disk if he/she found no suitable HDD in any of the combinations in **Configuration**
Purpose : For what purpose VM is required. This is optional.

Security Domain : Under which domain VM should be created. For user only one domain is activated by Admin, that is Research.
Assign Public IP : Whether or not, user want a public ip for the VM.
Faculty Approve : Under whom student is working or to whom student is making his request for the VM. Faculty name should be there in the database and it has to be verified first before going further.
Collaborators : What all users can be collaborators of the VM, it can be added to the VM

3. Title : My VMs

User can see his/her VMs here and also has access to the settings of the VM:



It shows the list of all the VMs under user

Name : Name of the VM

Owner : Name of the user to whom it was requested. It can be faculty or orgadmin or admin.

Private IP : Private IP assigned to the VM

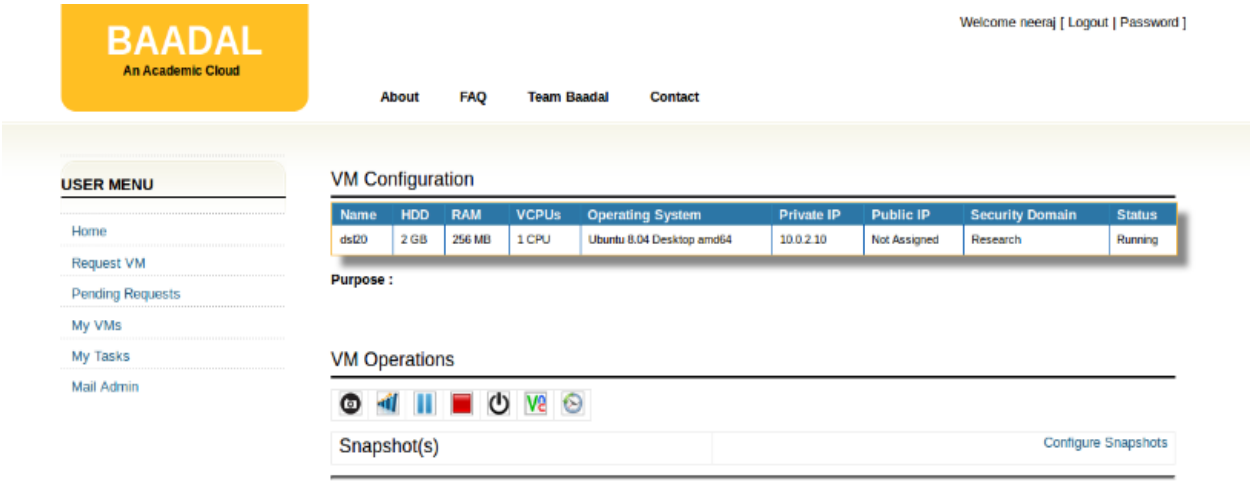
Public IP : If public ip is assigned to the VM then it would be shown here, otherwise it shows “Not Assigned”.

RAM : RAM in gb

CPU : Count of vCPUs

Status : status of VM. Running is this case

Settings : It redirects to the setting page of the VM. It looks like below



Here we can perform operation on VM. Operations are given under “VM Operations”.

This is for taking the snapshot of the VM

 This redirects to the page where the user can see the performance of the VM, resources used by the VM on hourly, daily, weekly, monthly or yearly basis

 This is used to pause the VM

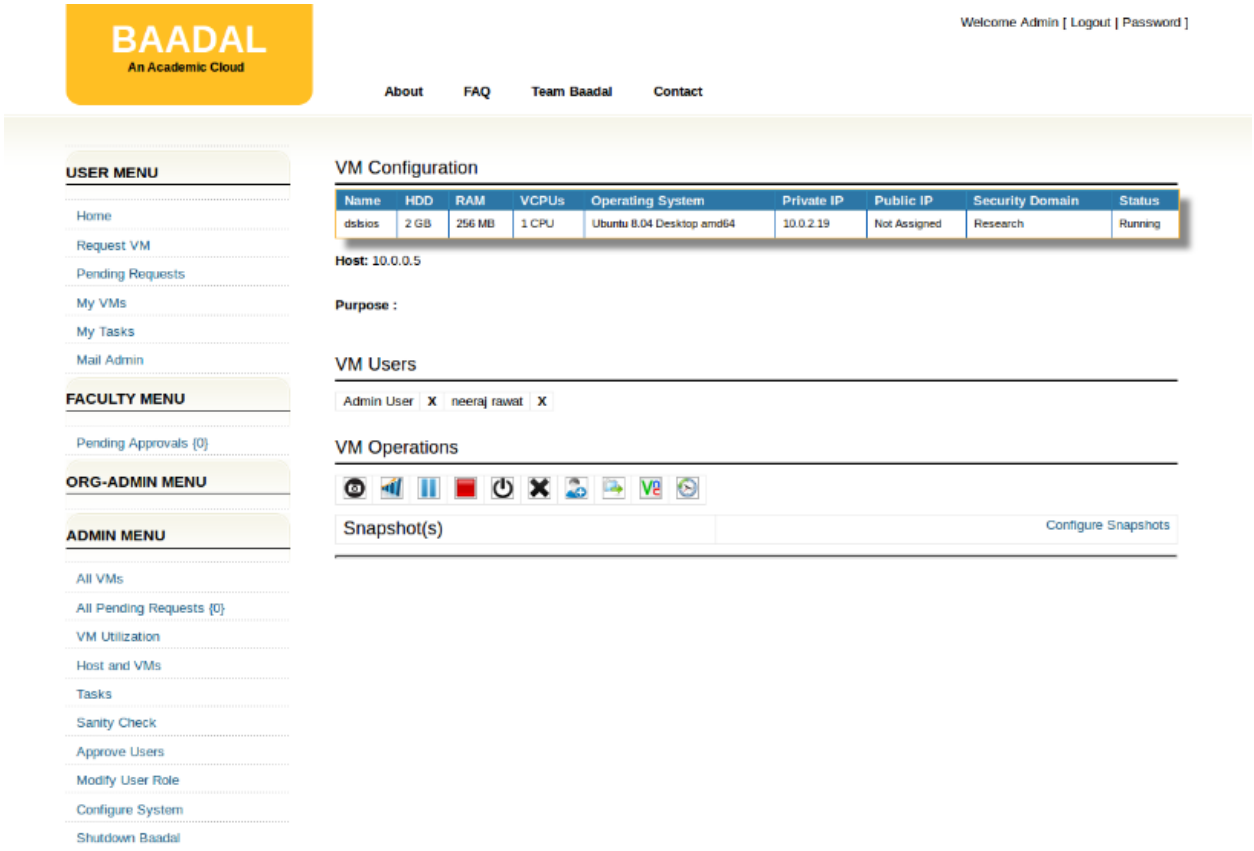
 This is for gracefully shutdown of the VM.

 Forcefully shutdown of the VM

 For the grant of VNC

 To see the VM history. It redirects to the history page

In case user is an Admin. VM setting page is as below



Extra VM operations under Admin are as below:

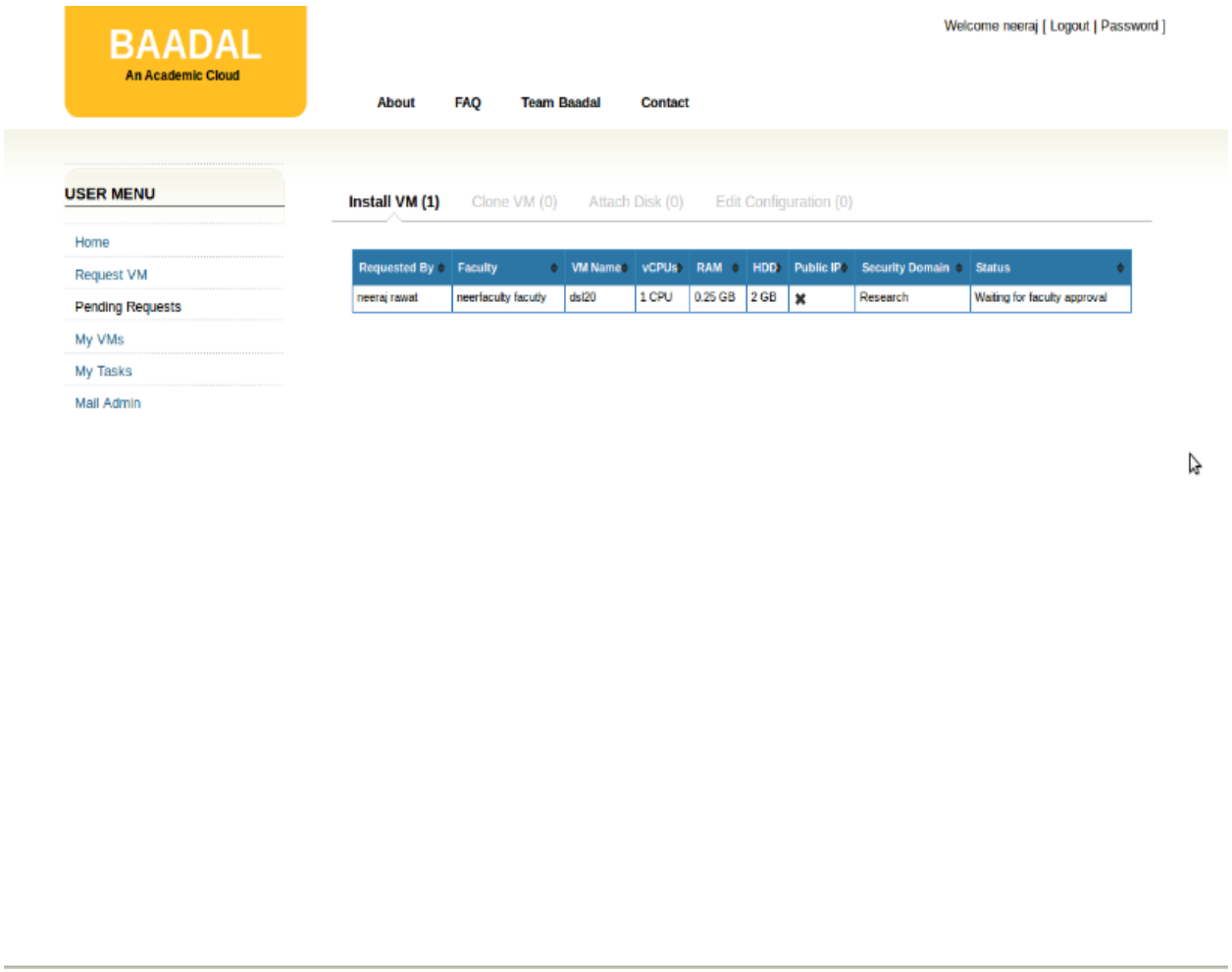
 This is used to add a user to VM. After adding that user will be seen below the “VM users” (above VM Operations)

 It is used to delete the VM. Only Admin can delete any VM

 It is used to migrate the VM.

4. Title : Pending Request

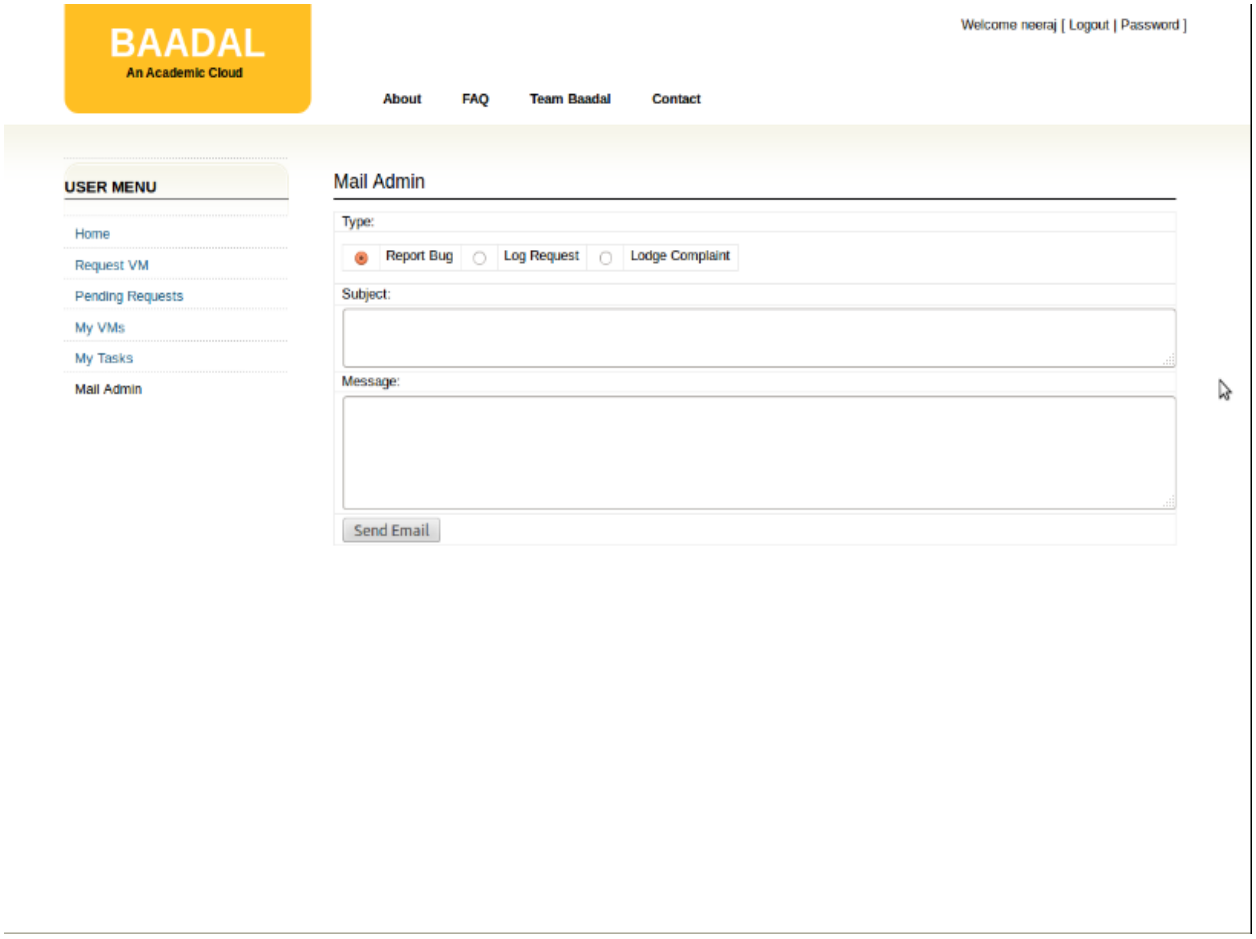
Here a user can see the status of his/her request for VM. Below should be the output:



Requested by : Name of the user who has made the request
Faculty : Name of the faculty to whom the user has made the request.
VM Name : Name of the VM given by the user during making the request
vCPU : Count of vCPU user has requested.
RAM : RAM in gb, user has requested.
HDD : Hard Disk size in gb, user has requested.
Public Ip : Whether user has requested for Public Ip or not. In this case user has not.
Security Domain : Research
Status : waiting for approval or rejection from the faculty.

5. **Title : Mail Admin**

In case a user want to send an email to admin of the baadal. Then below is the output of the mail admin functionality



This functionality is provided all type of users (admin also).