BLEKINGE TEKNISKA HÖGSKOLA

Monitoring the Performance of Virtual Machines

TEAM: '**SHIELD**'

HARSHINI NEKKANTI

HIMA BINDU NUTALAPATI

JYOTHI SPANDANA PENMETSA

NAVYA UPPALAPATI

PRIYASUBHA CHUNDRU

RAYWON TEJA KARI

SAIPHANI KRISHNA PRIYANKA KOLLURI

SASANK SAI SUJAN ADAPA

SRAVANI KANCHARLA

TULASI PRIYANKA SANABOYINA

VEERAVENKATA NAGA SOMESWARA MANITEJA DARISIPUDI

Document Type: **Design Document**

Version 1.3

CONTENTS

1. Preface

This document is intended to provide a description of the design for developing the executable software product. The description of the system is demonstrated using use case models, activity diagrams and sequence models and documented. It is intended for the CEO and the development team. This is the revised version 1.1.

The remainder of the document is organised as follows. Section 2 defines the technical terms and abbreviations used in the document. Sections 3,4,5 describe the detailed design of each module and the respective unit test plans. References are included in Section 6.

Customer: Patrik Arlos

CEO: Dragos Ilie

Revised version v1.3 on 2015-05-29

- Detailed design of the RESTful API included. Refer section 3.1

- Separated the tests for viewing alerts and sending e-mails. Refer section 3.2

- Tests for the data retrieval module added according to the feedback from CEO. Refer section 5.2

Revised version v1.2 on 2015-05-20

- Numbered all the figures in the document. Refer 3.1.1, 3.1.2, 3.1.3, 4.1 and 5.1

- Updated the type of data to be stored into MySQL and RRD databases. Refer 4.1

- Added tests for RESTful API and notifications (traps, SMS or email). Refer section 4.1

- Separate tests written for MySQL database and RRD database. Refer MOD2-TST_2 and MOD2-TST_3 in section 4.2

- Updated module test according to the feedback from the CEO. Refer section 5.2

Revised version v1.1 on 2015-05-15

- Section headings replaced as figure captions in sections 3.1, 4.1 and 5.1.

- Tests added for graphs and alerts. Refer MOD1-TST_4 and MOD1-TST_5 in section 3.2.

- Direction of the arrow corrected. Refer section 4.1.

- Test added to check if the data is stored into the database. Refer MOD2-TST_2 in section 4.2.

- Test added to check if the data is retrieved from a single entity. Refer MOD3-TST_1 in section 5.2.

Initial version v1.0 on 2015-05-05

-Initial release.

2. Glossary and abbreviations

API: Application Program Interface

CN: Compute Node

FNL: Functional

NFL: Non-functional

HTTP: Hyper Text Transfer Protocol

MOD: Module

REQ: Requirement

RRD: Round Robin Database

SYS: System

TST: Test

USR: User

VM: Virtual Machine

3. User interface module

The user interface module provides access for the user to interact with the system. The user can add devices to the monitoring list and remove devices from the monitoring list, view device statistics and utilization graphs, view status of the devices and receive email notifications when devices exceed the thresholds. It allows exporting the data shown to the user, to a third party application using RESTful API. This can be done by using URLs. The requested data will be exported in JSON format. For exporting the resource utilization data, the third party user must request the data using the URL in the format http://localhost/rest.php/?resource=<utlilizationmetric>

This module is placed in the frontend of the high level architecture. It communicates with the data storage unit using ssh. The module satisfies REQ-USR_FNL3, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_FNL7, REQ-USR_FNL8, REQ-USR_NFL1.

3.1. Detailed design

This section explains the design of the frontend module with the help of use case model, sequence model and activity diagrams.

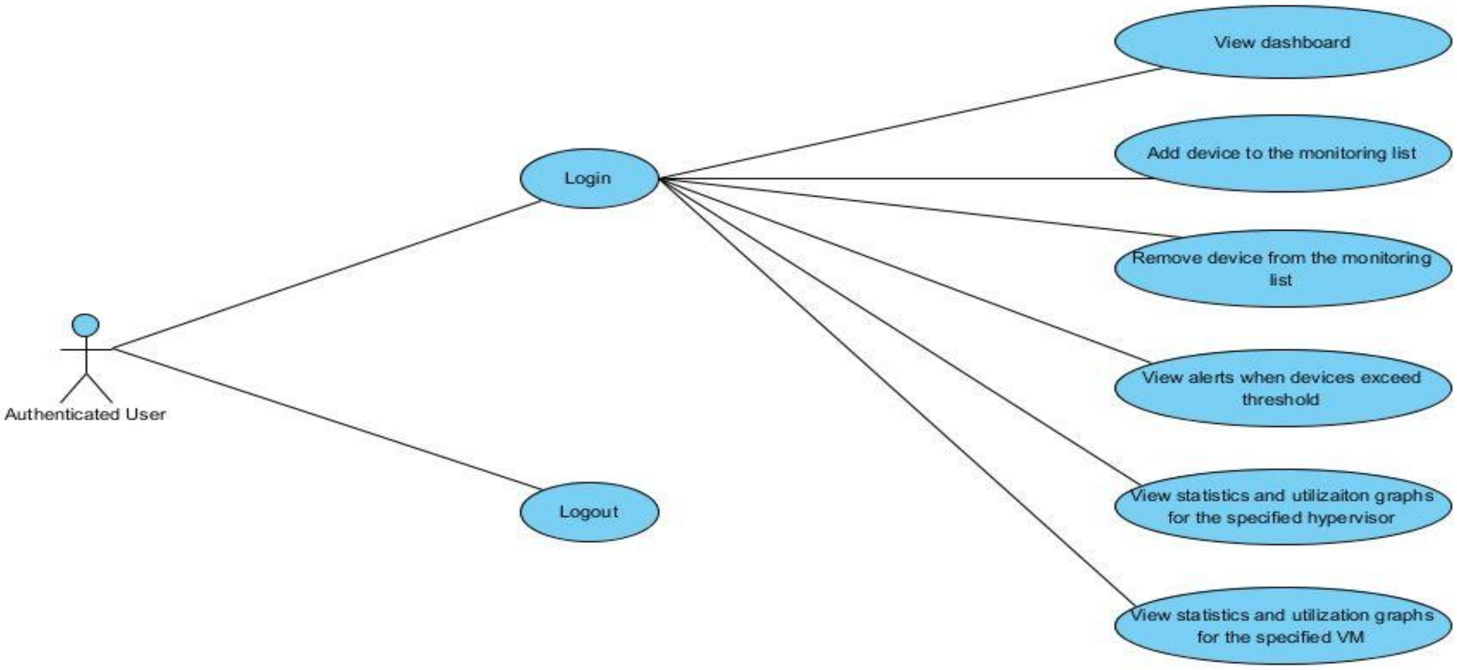
View dashboard
Add device to the monitoring list
Login
Remove device from the monitoring list
Authenticated User
View alerts when devices exceed threshold
Logout
View statistics and utilizaiton graphs for the specified hypervisor
View statistics and utilization graphs for the specified VM

Fig 3.1.1. Use case diagram representing the user interaction with the system

3.1.1. Sequence diagrams

The sequence diagrams show the operation of the processes in the sequential order that they occur.

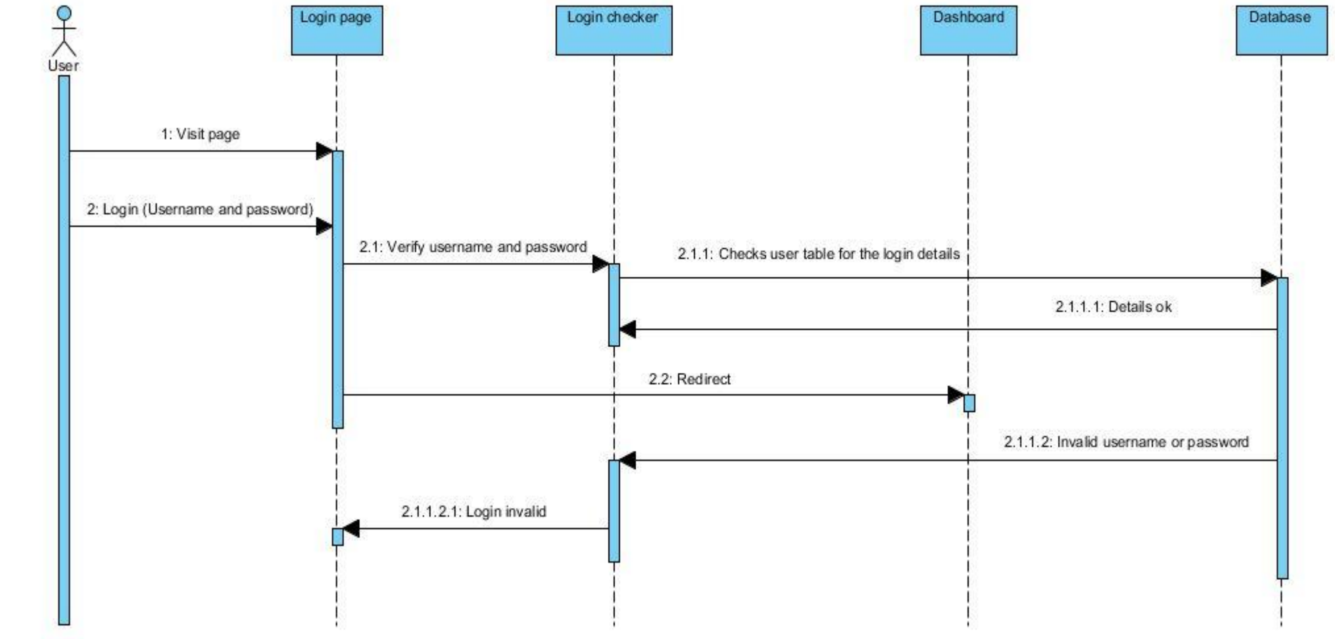
User
Login page
Login checker
Dashboard
Database
1: Visit page
2: Login (Username and password)
2.1: Verify username and password
2.1.1: Checks user table for the login details
2.1.1.1: Details ok
2.2: Redirect
2.1.1.2: Invalid username or password
2.1.1.2.1: Login invalid

Fig 3.1.2.1. Sequence diagram: Login

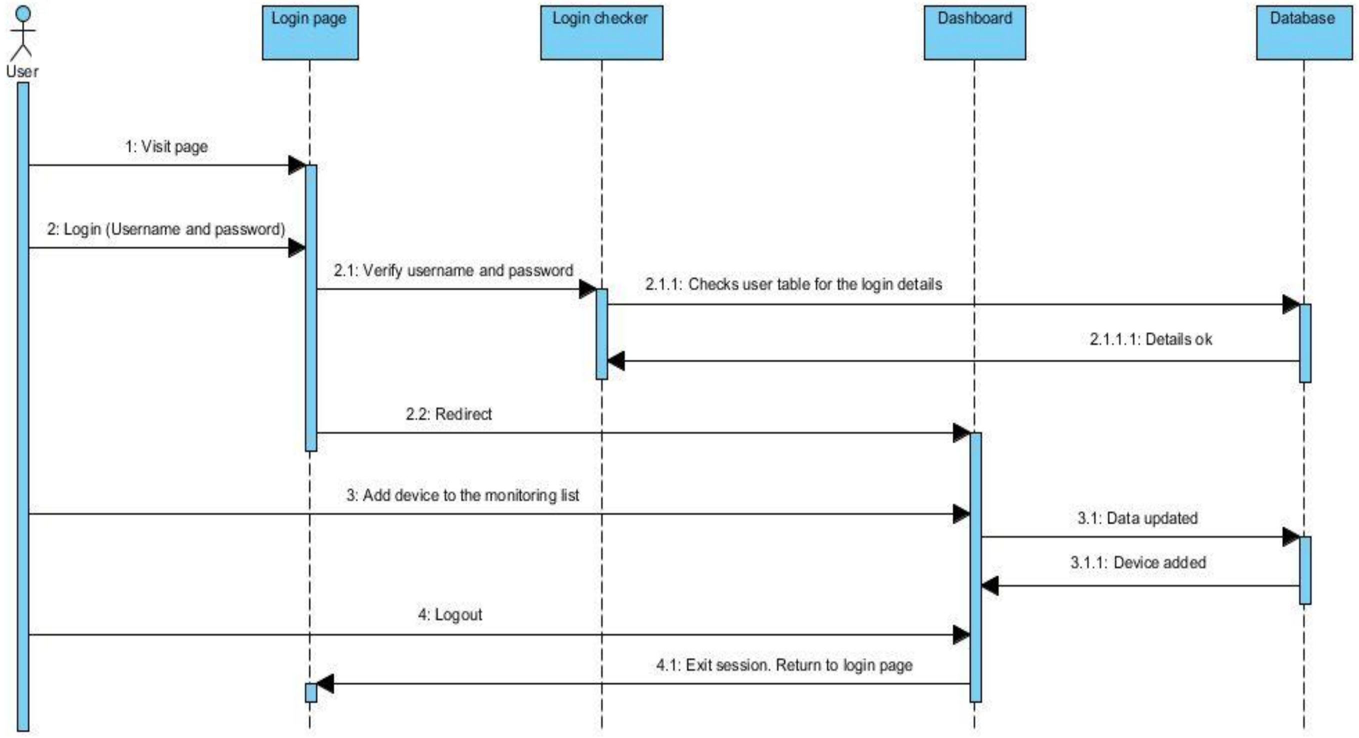
User
Login page
Login checker
Dashboard
Database
1: Visit page
2: Login (Username and password)
2.1: Verify username and password
2.1.1: Checks user table for the login details
2.1.1.1: Details ok
2.2: Redirect
3: Add device to the monitoring list
3.1: Data updated
3.1.1: Device added
4: Logout
4.1: Exit session. Return to login page

Fig 3.1.2.2. Sequence diagram: Add device to the monitoring list

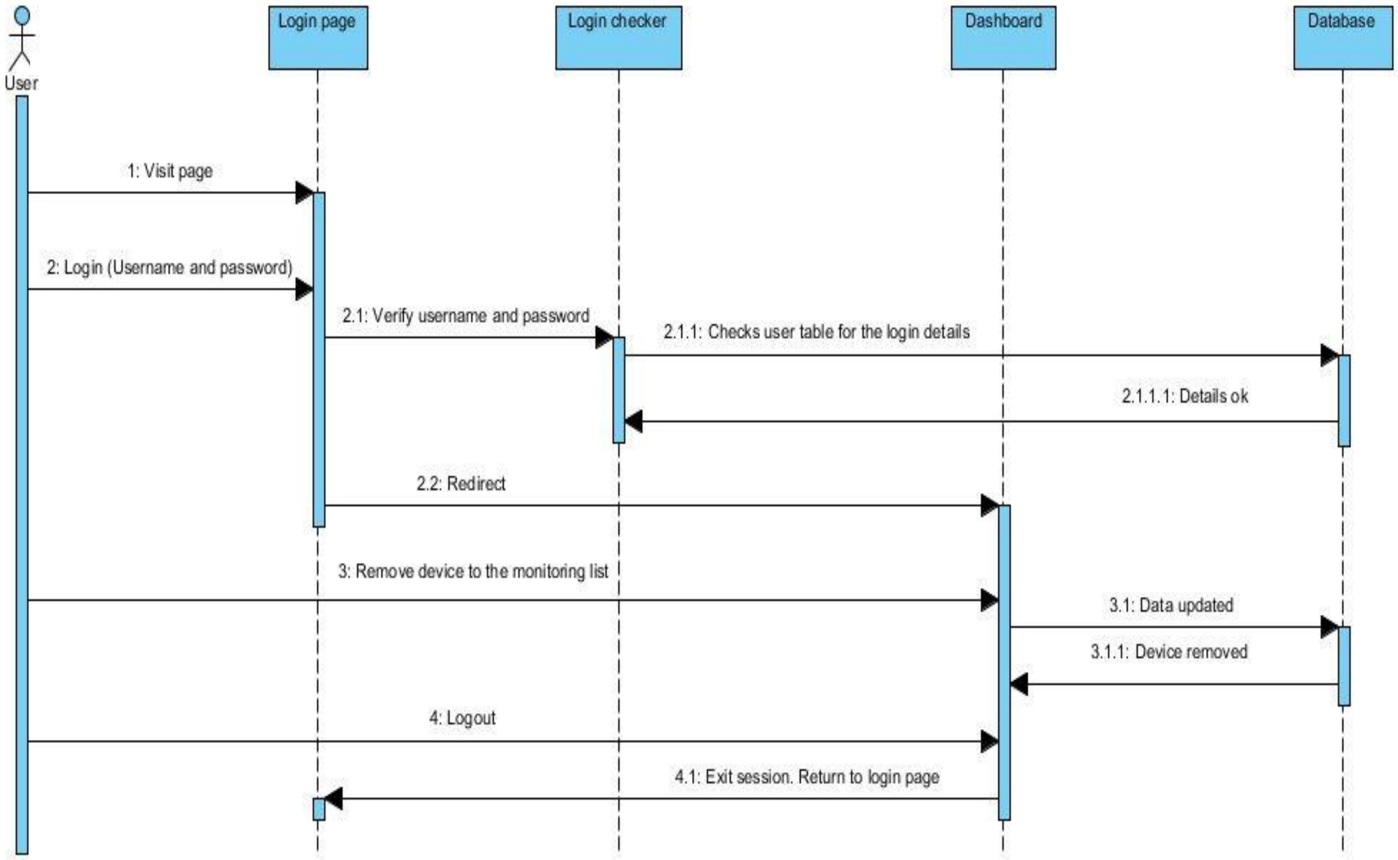
User
Login page
Login checker
Dashboard
Database
1: Visit page
2: Login (Username and password)
2.1: Verify username and password
2.1.1: Checks user table for the login details
2.1.1.1: Details ok
2.2: Redirect
3: Remove device to the monitoring list
3.1: Data updated
3.1.1: Device removed
4: Logout
4.1: Exit session. Return to login page

Fig 3.1.2.3. Sequence diagram: Remove device from the monitoring list

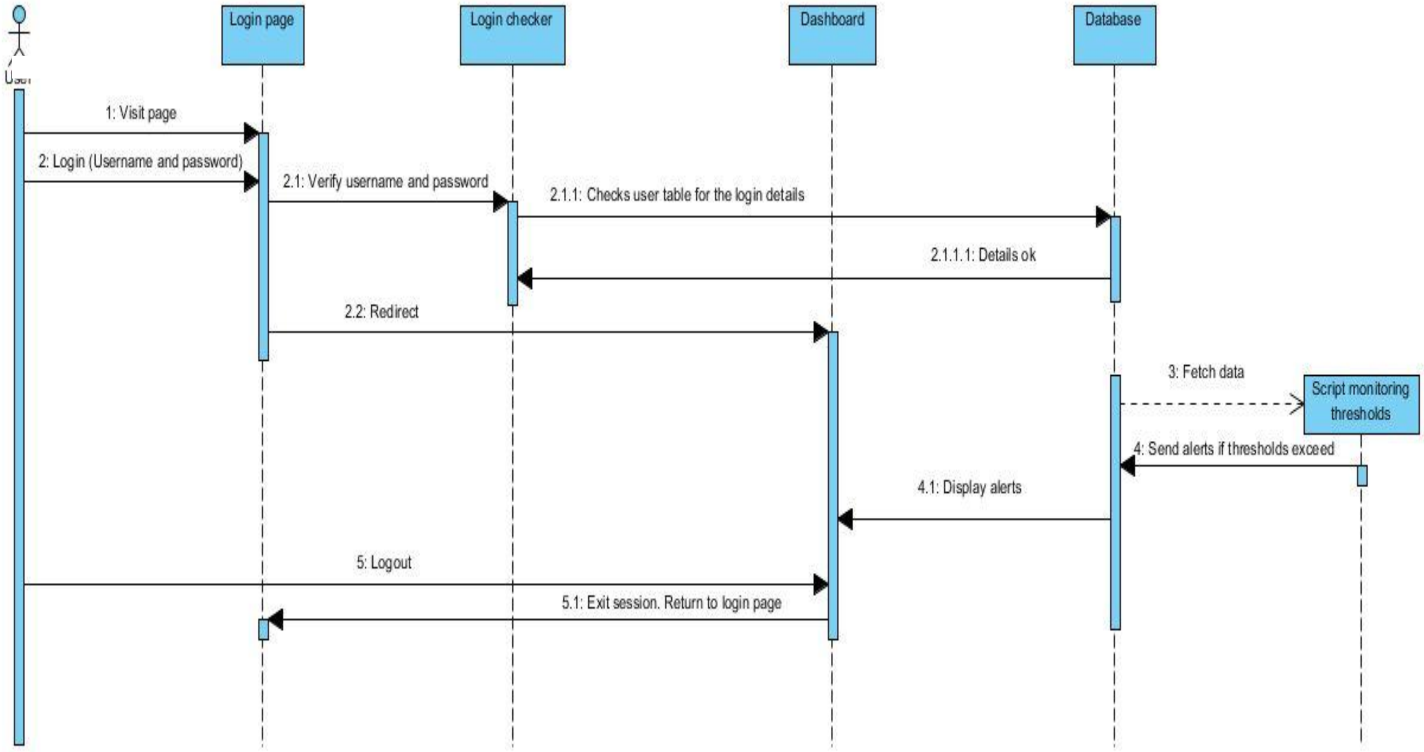
User
Login page
Login checker
Dashboard
Database
1: Visit page
2: Login (Username and password)
2.1: Verify username and password
2.1.1: Checks user table for the login details
2.1.1.1: Details ok
2.2: Redirect
3: Fetch data
Script monitoring thresholds
4: Send alerts if thresholds exceed
4.1: Display alerts
5: Logout
5.1: Exit session. Return to login page

Fig 3.1.2.4. Sequence diagram: View alerts when devices exceed thresholds

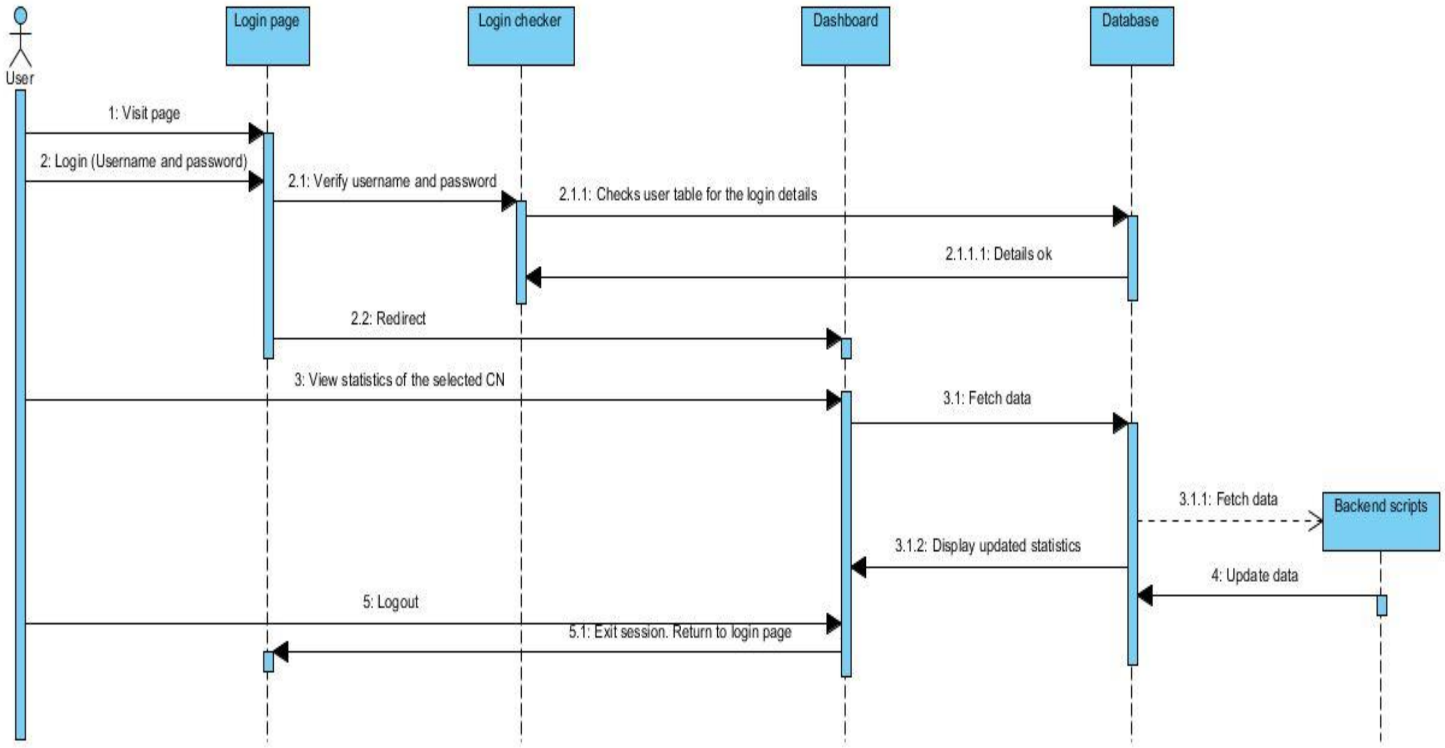
User
Login page
Login checker
Dashboard
Database
Backend scripts
1: Visit page
2: Login (Username and password)
2.1: Verify username and password
2.1.1: Checks user table for the login details
2.1.1.1: Details ok
2.2: Redirect
3: View statistics of the selected CN
3.1: Fetch data
3.1.1: Fetch data
3.1.2: Display updated statistics
4: Update data
5: Logout
5.1: Exit session. Return to login page

Fig 3.1.2.5. Sequence diagram: View statistics of a specific CN

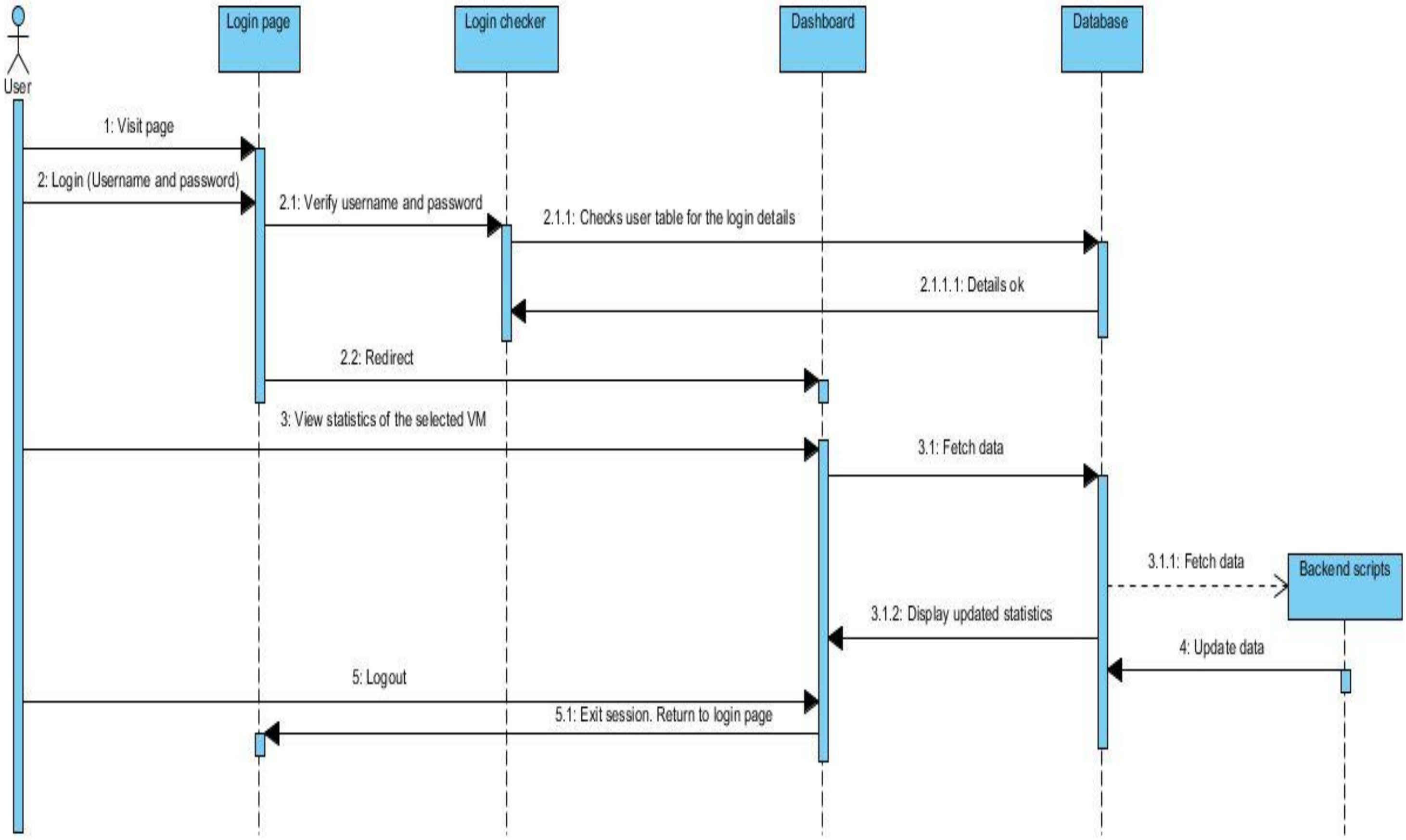
User
Login page
Login checker
Dashboard
Database
Backend scripts
1: Visit page
2: Login (Username and password)
2.1: Verify username and password
2.1.1: Checks user table for the login details
2.1.1.1: Details ok
2.2: Redirect
3: View statistics of the selected VM
3.1: Fetch data
3.1.1: Fetch data
3.1.2: Display updated statistics
4: Update data
5: Logout
5.1: Exit session. Return to login page

Fig 3.1.2.6. Sequence diagram: View statistics of a specific VM

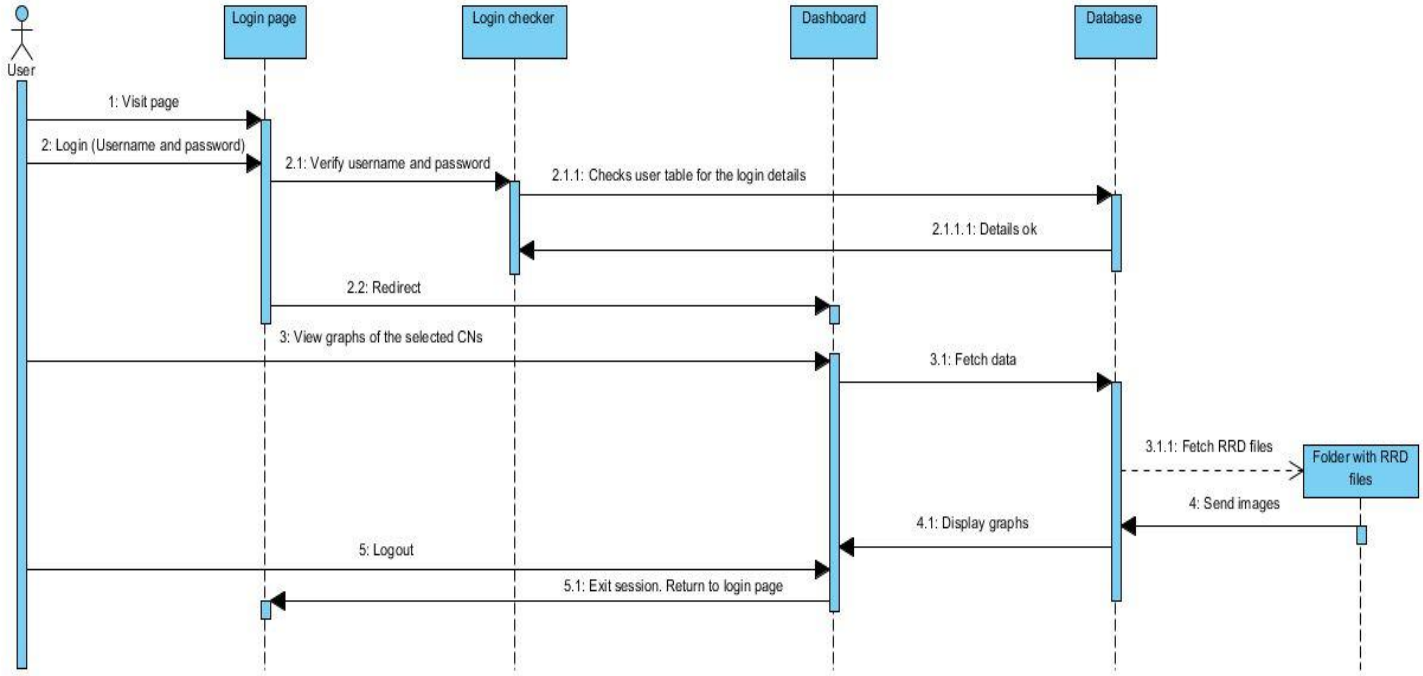

Fig 3.1.2.7. Sequence diagram: View graphs of a specific CN

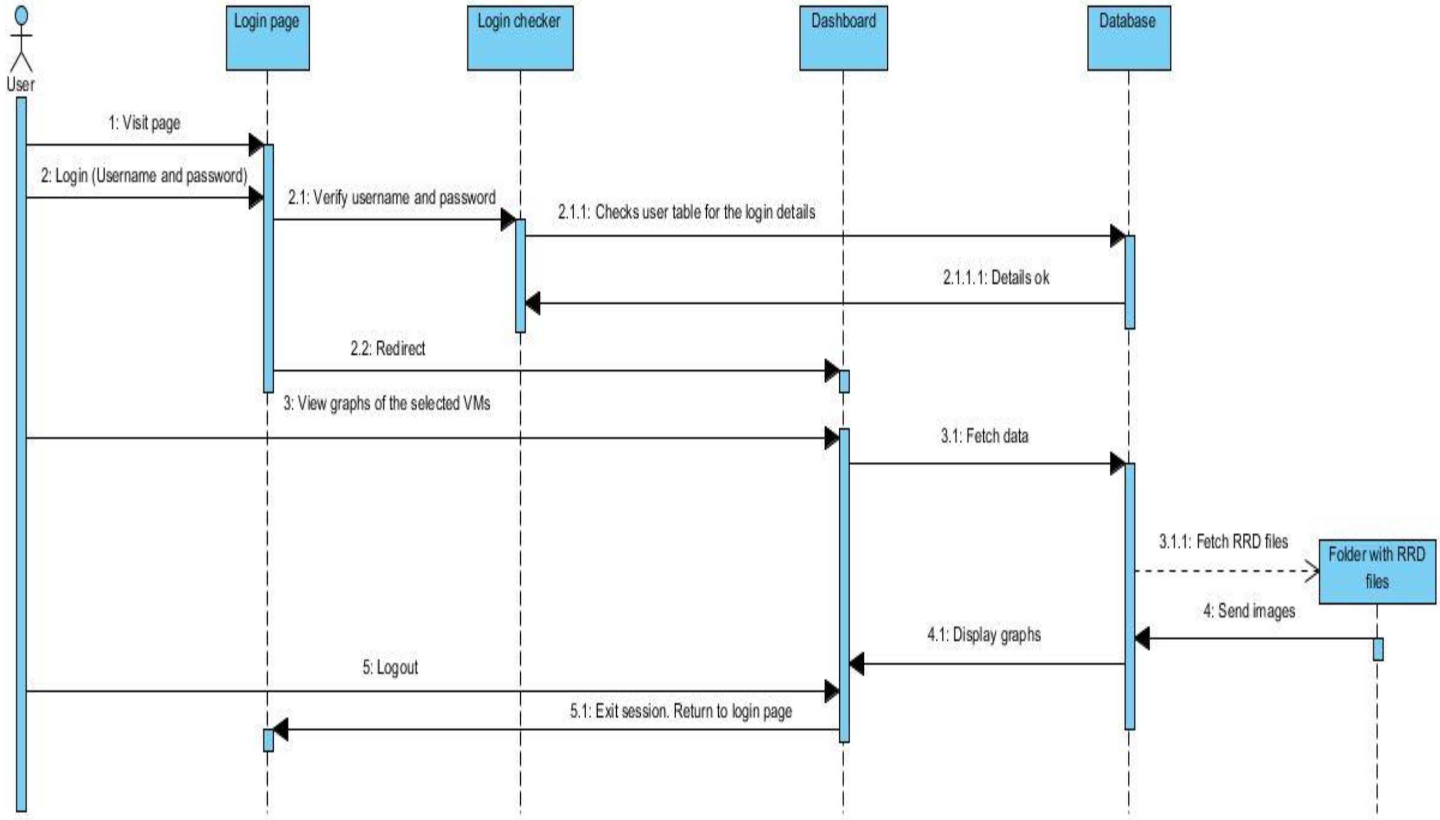

Fig 3.1.2.8. Sequence diagram: View graphs of a specific VM

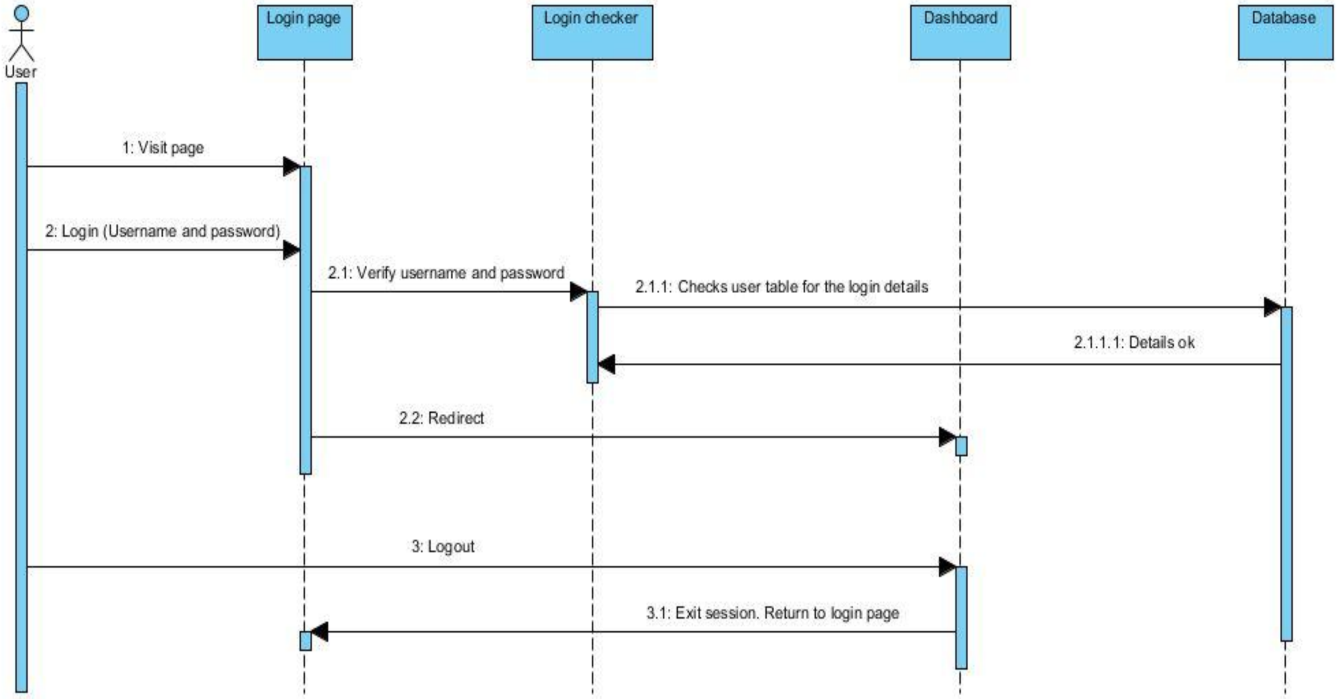
User
Login page
Login checker
Dashboard
Database
1: Visit page
2: Login (Username and password)
2.1: Verify username and password
2.1.1: Checks user table for the login details
2.1.1.1: Details ok
2.2: Redirect
3: Logout
3.1: Exit session. Return to login page

Fig 3.1.2.9. Sequence diagram: Logout

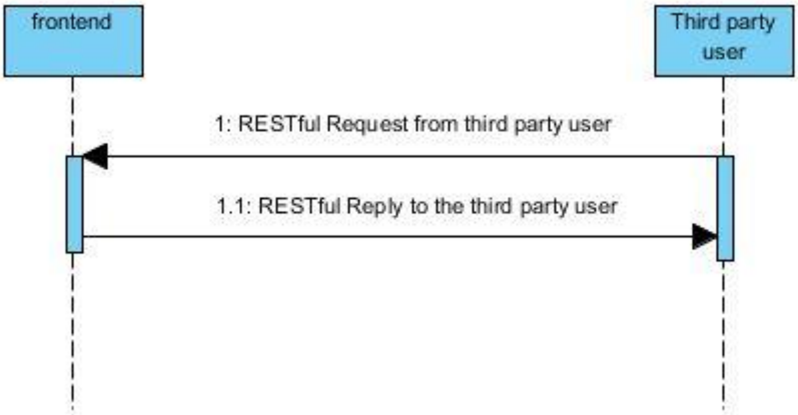
frontend
Third party user
1: RESTful Request from third party user
1.1: RESTful Reply to the third party user

Fig 3.1.2.10. Sequence diagram: Third party user

3.1.2. Activity diagrams

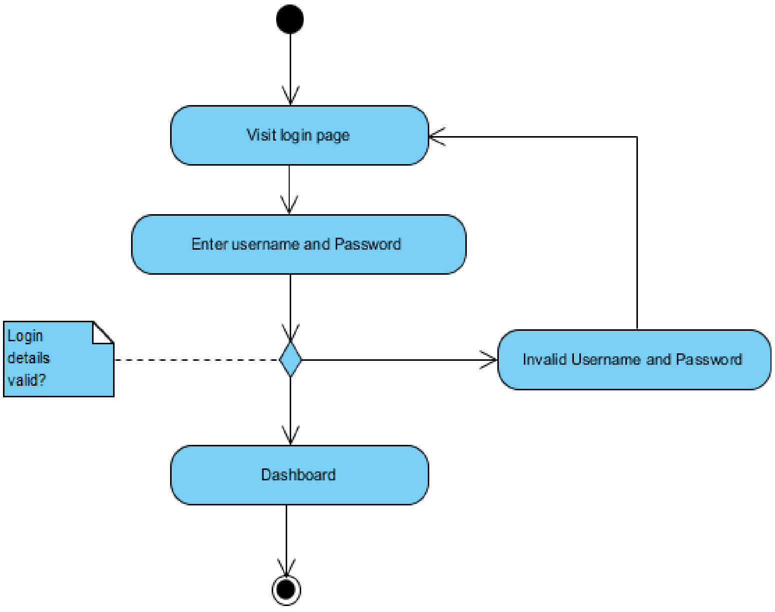

Fig 3.1.3.1. Activity diagram: Login

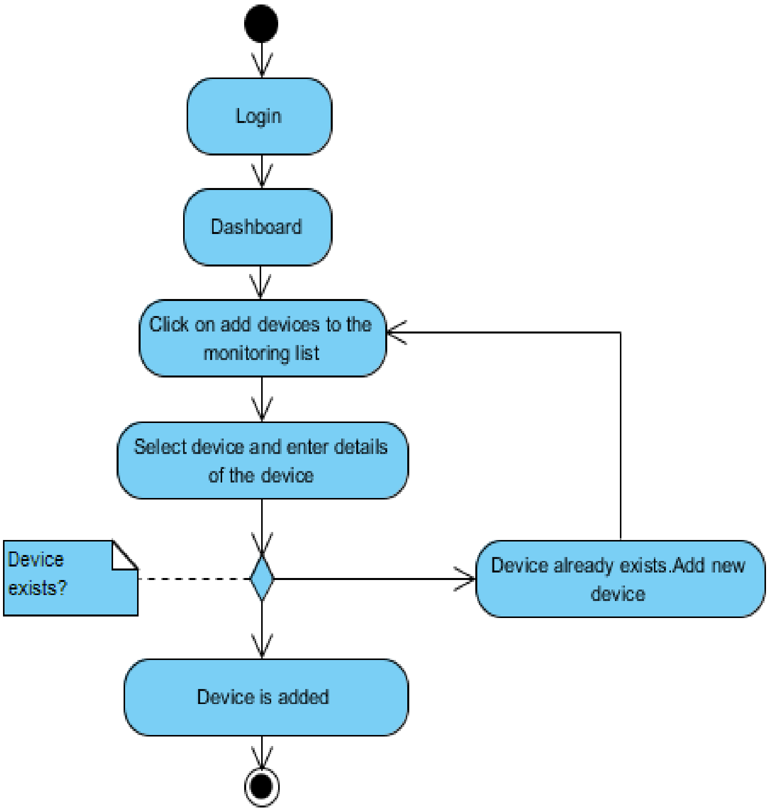
Login
Dashboard
Click on add devices to the monitoring list
Select device and enter details of the device
Device exists?
Device already exists.Add new device
Device is added

Fig 3.1.3.2. Activity diagram: Add device to the monitoring list

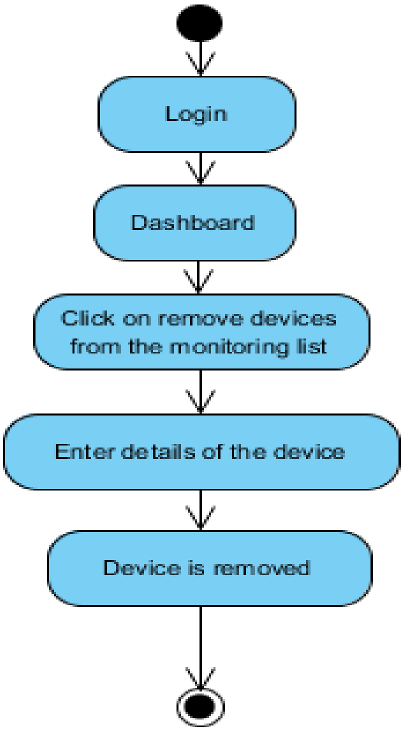
Login
Dashboard
Click on remove devices from the monitoring list
Enter details of the device
Device is removed

Fig 3.1.3.3. Activity diagram: Remove device from the monitoring list

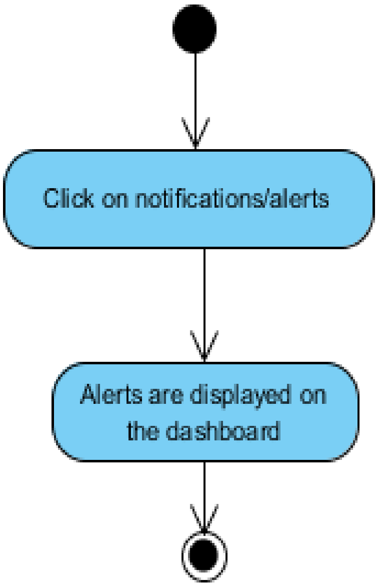
Click on notifications/alerts
Alerts are displayed on the dashboard

Fig 3.1.3.4. Activity diagram: View alerts when devices exceed threshold

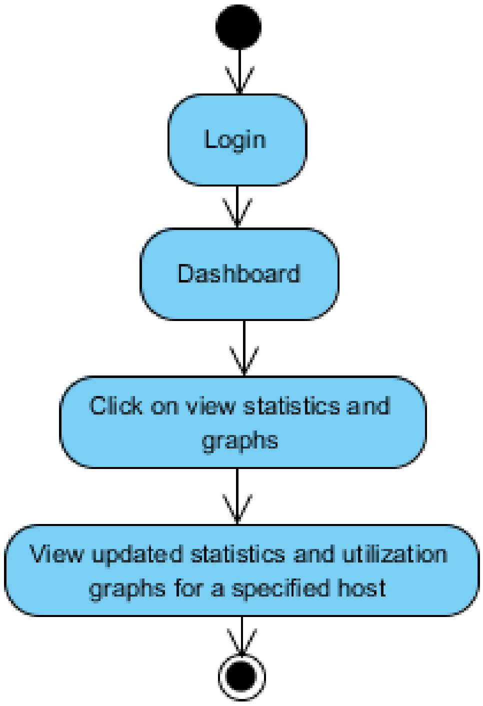
Login
Dashboard
Click on view statistics and graphs
View updated statistics and utilization graphs for a specified host

Fig 3.1.3.5. Activity diagram: View statistics and utilization graphs of a specific CN

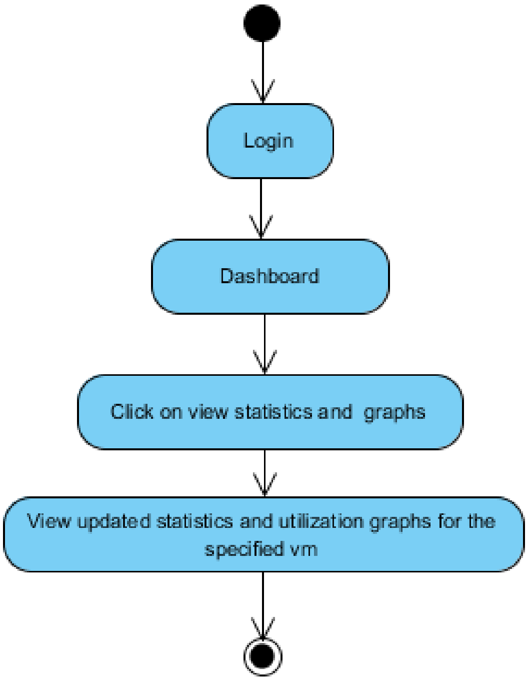
Login
Dashboard
Click on view statistics and graphs
View updated statistics and utilization graphs for the specified vm

Fig 3.1.3.6. Activity diagram: View statistics and utilization graphs of a specific VM

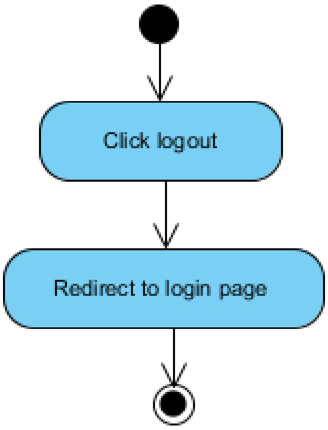

Fig 3.1.3.7. Activity diagram: Logout

3.2. Unit test plan

The unit test plan describes the procedure to test the module and verify its functions according to the requirements.

Module 1: User Interface System

Test: MOD1-TST_1

Purpose: To test if the login details are valid.

Requirements: REQ-USR_FNL2, REQ-USR_FNL3, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_NFL1

Environment: Web browser.

Operation:

- Open the web browser
- Enter URL to access the login page
- Provide username ‘admin’ and password ‘shield’

Expected Result: The user will be redirected to the dashboard

Test: MOD1-TST_2

Purpose: Test for adding device to the monitoring list.

Requirements: REQ-USR_FNL2, REQ-USR_FNL3, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_NFL1

Environment: Dashboard.

Operation:

- Enter login details to access the dashboard.
- Click on devices and select 'Add device to the monitoring list'
- Enter the IP address of the device and click 'Add' to add the device.

Expected Result: Device will be added to monitoring list on dashboard.

Test: MOD1-TST_3

Purpose: Test for removing device from the monitoring list

Requirements: REQ-USR_FNL2, REQ-USR_FNL3, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_NFL1

Environment: Dashboard.

Operation:

- Enter login details to access the dashboard.
- Click on the device which has to be removed.
- Click 'Remove device' to remove the device from the monitoring list.

Expected Result: Device will be removed from the dashboard.

Test: MOD1-TST_4

Purpose: Test for displaying graphs on the dashboard

Requirements: REQ-USR_FNL1, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_NFL1

Environment: Dashboard.

Operation:

- Enter login details to access the dashboard.
- Click on the device for which the graphs are required.
- Select the resource (CPU load and utilization, memory usage, network I/O usage or disk usage) for which the graphs are to be displayed.

Expected Result: The graphs for the resources of the required device will be displayed.

Test: MOD1-TST_5

Purpose: Test for displaying alerts.

Requirements: REQ-USR_FNL1, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_NFL1

Environment: Dashboard.

Operation:

- Enter login details to access the dashboard.
- Whenever the devices exceed thresholds, the colour of the device should change to red.

Expected Result: When any device's resource consumption exceed threshold, device colour changes.

Test: MOD1-TST_6

Purpose: Test for notifying the user via email when devices exceed thresholds.

Requirements: REQ-USR_FNL1, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_NFL1

Environment: Dashboard.

Operation:

- Enter login details to access the dashboard.
- Whenever any device exceeds threshold, an email should be sent to the user indicating the device status.

Expected Result: Notification as an email is sent to the user.

Test: MOD1-TST_7

Purpose: To check the scalability of the software

Requirements: REQ-USR_FNL1, REQ-USR_FNL5, REQ-USR_NFL1.

Environment: Dashboard

Operation:

- Enter login details to access the dashboard.
- Click on ‘Add device’ and add a number of devices.
- Select any resource and view the utilization graphs and statistics for all the devices.
- If graphs are produced for the newly added devices, scalability test is successful.

Expected Result: Utilization graphs and statistics should be obtained for newly added devices.

Test: MOD1-TST_8

Purpose: RESTful API to export data shown to the user

Requirements: REQ-USR_FNL8.

Environment: PHP, Web browser

Operation:

- Enter the URL provided for the third party access with the required parameters
- The requested information is displayed in JSON format

Expected Result: Data exported successfully.

4. Data Storage Module

The data storage module is a database used to store the information retrieved by the data retrieval module. The data stored in the database can be accessed by the user interface module using MySQL API and RRD API. MySQL database is used to store the details of the device and the login credentials while RRD is used to store the utilization and historical aggregate data required in generating the utilization graphs.

4.1. Detailed design

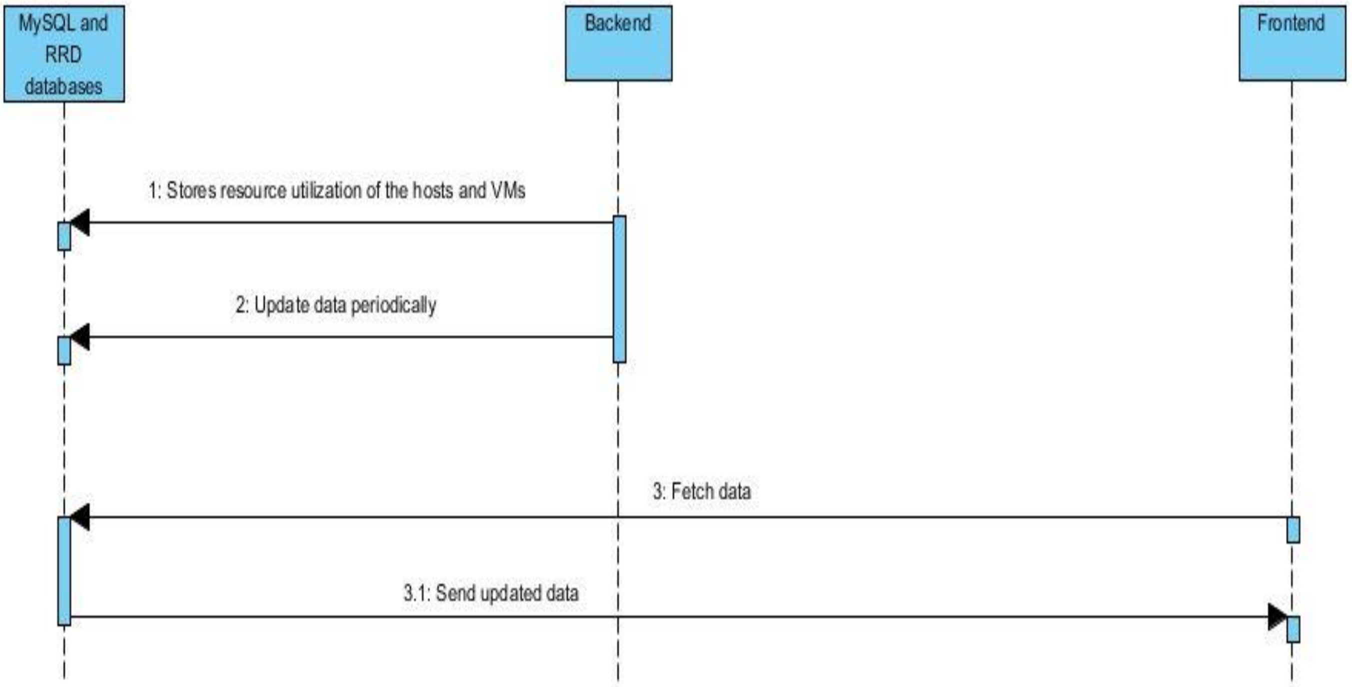
MySQL and RRD databases
Backend
Frontend
1: Stores resource utilization of the hosts and VMs
2: Update data periodically
3: Fetch data
3.1: Send updated data

Fig 4.1. Sequence diagram for data storage module

4.2. Unit test plan

The unit test plan describes the procedure to test the module and verify its functions according to the requirements.

Module 2: Data Storage Unit

Test: MOD2-TST_1

Purpose: To verify credibility of the software.

Requirements: REQ-USR_FNL1, REQ-USR_FNL3, REQ-USR_NFL1

Environment: Login to the dashboard and select a device.

Operation:

- Enter login details to access the dashboard.
- Select a device and click ‘View statistics and graphs’.
- Obtain the resource utilization metrics in the terminal using the respective commands.

Expected Result: The values obtained on the dashboard and on the terminal should be similar.

Test: MOD2-TST_2

Purpose: Test to check whether the resource utilization metrics are stored into RRD database.

Requirements: REQ-USR_FNL1, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_NFL1

Environment: Dashboard.

Operation:

- Retrieve the data using the data retrieval module.
- Store the retrieved data into the database.
- View the database for the number of rows and columns affected.

Expected Result: Resource utilization metrics will be stored into the database.

Test: MOD2-TST_3:

Purpose: Test to check whether the device data and login credential data are stored into MySQL database

Requirements:

Environment: MySQL database

Operation:

- Create table with username and password as fields.
- Add data into the table using Insert statements.
- Create table and add the data of the device with device details as input field.

Expected result: Data should be added into the database.

5. Data retrieval module

The data retrieval module is used to retrieve the CPU load and utilization, I/O usage, network usage, memory usage and disk usage of the devices to be monitored. SNMP is used to poll the devices for the information and the retrieved information is stored in a database. When the product is installed, the retrieval module retrieves the information and stores into the database periodically.

5.1. Detailed design

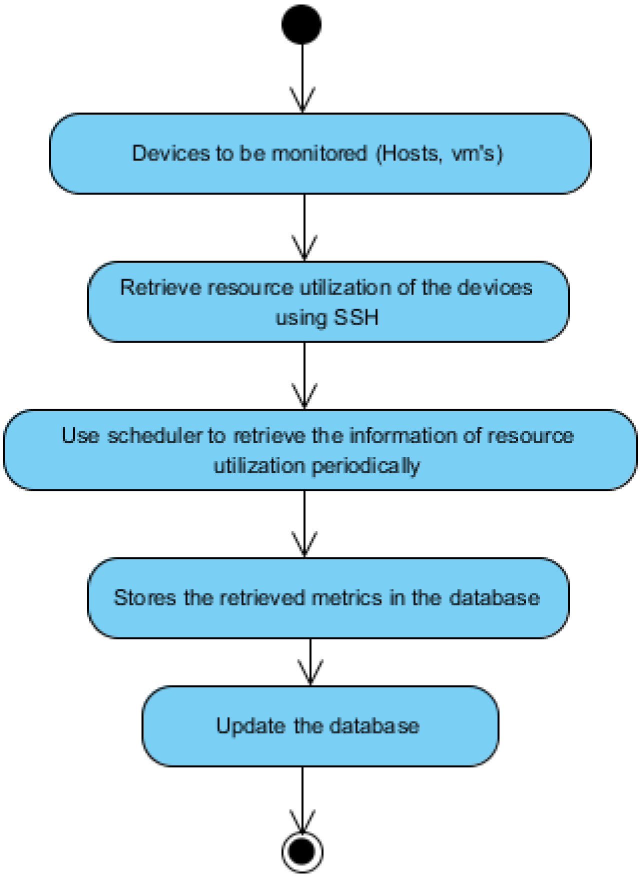
Devices to be monitored (Hosts, vm's)
Retrieve resource utilization of the devices using SSH
Use scheduler to retrieve the information of resource utilization periodically
Stores the retrieved metrics in the database
Update the database

Fig. 5.1. Activity diagram for data retrieval module

5.2. Unit test plan

The unit test plan describes the procedure to test the module and verify its functions according to the requirements.

Module 3: Data Retrieval Module

Test: MOD3-TST_1

Purpose: Test to check if the required hypervisors are installed

Requirements: REQ-USR_FNL1, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_NFL1

Environment: Ubuntu terminal.

Operation:

- Open the terminal in Ubuntu and enter the command “sudo virt-what”
- The hypervisor installed on the system will be displayed on the terminal.

Expected Result: The command will be executed and the installed hypervisors are shown.

Test: MOD3-TST_2

Purpose: Test for communication with the hypervisors (Xen/KVM)

Requirements: REQ-USR_FNL1, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_NFL1

Environment: Xen and KVM

Prerequisite: SSH should be run once at least with the command "ssh <username>@<IP address of the system>" and enter the password of that system.

Operation:

- Open the terminal in Ubuntu and enter any command (for example: " sudo ifconfig ") to retrieve information about the required system

Expected Result: The requested information will be displayed on the terminal (IP address if " sudo ifconfig " is used).

Test: MOD3-TST_3

Purpose: Test to check if the data is retrieved for a single host.

Requirements: REQ-USR_FNL1, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_NFL1

Environment: Ubuntu terminal.

Operation:

- Open terminal in Ubuntu and run the file "utils.pl"
- The resource utilization (CPU, memory, network I/O and disk usage) values of the specific host are displayed.

Expected Result: The script is executed and respective values will be displayed on the terminal.

Test: MOD3-TST_4

Purpose: Test to check if the data is retrieved for single VM

Requirements: REQ-USR_FNL1, REQ-USR_FNL4, REQ-USR_FNL5, REQ-USR_FNL6, REQ-USR_NFL1

Environment: Ubuntu terminal.

Operation:

- Open the terminal in Ubuntu and run the file “vmutils.pl”
- The resource utilization (CPU, memory, network I/O and disk usage) values of the VM are displayed.

Expected Result: The script will be executed and respective values will be displayed on the terminal.

6. References

- Sommerville, Ian. *Software Engineering*, 9th ed. Addison-Wesley, 2011