UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Software Requirements Specification

Team Red

22 February 2018

Contents

1 Introduction

1.1 Purpose

The software requirements specication (SRS) will appropriately outline the functional requirements of the Benchmarking system to ensure that an external party, such as designers, developers and clients, could develop the functionality to a required degree without further instruction. Hence, the functional requirements should be precise and extensive to eliminate deviation from the goals of the system.

1.2 Scope

The Benchmarking system is a web based application designed to enable performance profiling by providing hardware and hardware monitoring capabilities, then storing and reporting the findings for each service request. The system will monitor and report CPU usage, memory usage, power consumption, heat generation, elapsed time and network usage. The system will be able to provide hardware to run benchmarking requests and will be able to accept hardware that the user provides themselves. The system will minimize side effects that are not a concern of the benchmarking requests. The system will accept user code that may come in different programming languages, confirm its runnable and execute it to generate profiling information. The system will not optimize the code nor correct it. Through using the Benchmarking system, users will be able to see the efficiency of their provided code and how it will run on specific infrastructure. The system will also provide administrative capabilities for specific users, allowing them to manage other users and view their activities.

1.3 Definitions, Acronyms and Abbreviations

SRS	Software Requirements Specification
WIFI	Wireless Fidelity
Administrator	User with absolute privileges and system control
Hardware Seeking User	User that uses the Benchmarking system and requests hardware from it
Hardware Providing User	User that uses the Benchmarking system and provides their own hardware
System	Composition of infrastructure, resources, networks and subsystems necessary to make up the Benchmarking system
CPU	Central Processing Unit

1.4 References

- D.C. Kung, Object Oriented Software Engineering. McGraw Hill, 2014

1.5 Overview

The SRS will help give a detailed representation of the functional requirements and how the sub-elements of the system interact with one another to achieve the Benchmarking system's purpose.

2 Overall Description

2.1 Product Perspective

The main system will be a server on the University of Pretoria (UP) campus and will be connected to a database that stores profiling information of previously executed benchmarking requests. Users will log on to the web-based application through a browser and provide code to the Benchmarking system. They will then be able to choose between providing their own hardware for monitoring or use existing hardware provided by the system. They then have to choose which attributes are to be monitored. They will then execute the benchmarking request and view it real-time and/or view the profiling report generated at the end of the benchmarking request. Hence users will be able to view their code's performance and how it compares to other versions or other code. Administrative users will be able to log on to the Benchmarking system and view the activities of non-managerial users, as well as manage their accounts.

2.2 Product Function

- The service will enable code and/or hardware based performance profiling for given code samples or executables
- The system will provide profiling information real-time and post benchmark request execution
- The system will provide access to profiling information of previously run benchmarking requests
- The system will allow administrative users to manage existing normal users and the system itself
- The system will allow users to provide their own hardware for benchmarking requests
- The system will allow users to export and/or share profiling information
- The system will allow users to choose hardware specifications for each benchmarking request

2.3 User Characteristics

- Hardware seeking users: These are your average users. They will have a basic knowledge of computers and computer hardware. These will be the users that provide code and use existing hardware provided by the Benchmarking system. They will have normal user privileges, thus have control only over their own account.
- Hardware providing users: These users have a higher knowledge of computers and computer hardware than the usual hardware seeking user. These will be the users that provide their own hardware for benchmarking requests. They will have normal user privileges, thus have control only over their own account.
- Managerial users: These are users with elevated account privileges. They have managerial skills and a basic knowledge of computers and computer hardware. Managerial users will be able to monitor normal user activities as well as manage existing normal user accounts.
- Administrator: These are back-end users. They have access to the Benchmarking system's existing hardware as well managerial capabilities over existing normal and managerial users. They have expert knowledge of computers and computer hardware. They have full control over system resources.

2.4 Constraints

This section elaborates on the limitations of the options that are available when developing the Benchmarking system

- The system is a web-based application and therefore must run on a device that has browser support.
- The system runs on a remote server therefore an internet connection is required in order to communicate with the Benchmarking system.
- The system may not report real-time accurately, depending on the network connection strength.
- The system's profiling report information is limited by the existing hardware in the system as it can not provide information for hardware it can not monitor.

2.5 Assumptions and Dependencies

All assumptions relate to the user and the device they are using. It is assumed that the user seeking to use the Benchmarking system has basic knowledge of how to use an IT device with browser support. It is assumed that the device the user is using has browser support and can connect to the internet.

3 Specific Requirements

This section expands on the functional requirements of the system. It gives a detailed description of the system and all of its use cases.

3.1 External Interface Requirements

3.1.1 User Interface

- Initially, the user interface should show a login screen on a browser for first-time users and returning users.
- Once logged in, users will be directed to a page that allows them to create a benchmarking request. There will be functionality allowing users to upload code, choose hardware, or provide hardware, then permit them to execute the benchmarking request.
- Once the benchmarking request has started running, the user will have a page available to view live benchmarking requests running in real-time.
- There will be a page allowing users to view previous benchmarking requests and the profiling information generated by them.
- There will be a page allowing users to edit their account information.
- For users with administrative rights, there will be a page where they may view other existing users and their current or historical activities with the Benchmarking system. They may also manage the users from this page.

3.1.2 Hardware Interface

This interface applies only for users providing hardware to the Benchmarking system.

- There will be an interface that pops up, asking the user for permission to run on their machine.
- If the user allows access, an interface will check the user's system to see what hardware is available and let the user know what can be utilized by the Benchmarking system.
- After the user specifies the what hardware they would like to use, the interface closes and returns control to the browser interface.

3.1.3 Software Interface

- Browser application and database communication to get benchmarking request and user information.
- Browser application and the users hardware application communication to get information about the users available hardware.

- Browser application and hardware monitoring subsystem communication to get real-time and post benchmarking request profiling information.

3.1.4 Communication Interface

- Non implemented by the benchmarking system. Communication is left to the browser and underlying operating system.

3.2 Functional Requirements

This section will elaborate on all functional requirements in the system. It includes all Use Case diagrams, Actor-System interaction diagrams and Traceability matrices.

3.2.1 High-level Requirements

- FR-1: The system should provide a web interface for users to request and specify the benchmarking services they need.
- FR-2: The system should be able to execute benchmarking requests on isolated machines.
- FR-3: The system should be able to analyze a variety of performance attributes.
- FR-4: The system should allow managers to manage users and profiling nodes.
- FR-5: The system should allow users to register, login and edit user accounts

3.2.2 Use cases

1. User Account Management Subsystem
   (a) Register
      i. **Description:** The register functionality allows users the capability to self-register on the system. These users can be only be a basic type of user with basic user roles, other types of users with advanced roles can only be registered on the User Account Management Subsystem.
      ii. **Precondition:**No other user exists on the system with same email address. The email address is valid.
      iii. **Postcondition:**User receives registration confirmation

   (b) Login

i. **Description:** The login functionality allows registered users to log in to the system, only logged in users can use the system.

ii. **Precondition:**Valid credentials of a registered user are submitted.

iii. **Postcondition:** User successfully logged into the system. User logged in event successfully logged on the system.

(c) Edit Account

i. **Description:**The edit account functionality is offers registered users the capability to modify their attributes and/or properties on the system.

ii. **Precondition:** User is logged in. New user information does not contravene any attribute/property uniqueness constraints on the system. If the update is/includes a display picture the file type of the new display picture is supported

iii. **Postcondition:** User successfully updated on the system and changes are immediately reflected. User update event successfully logged on the system.

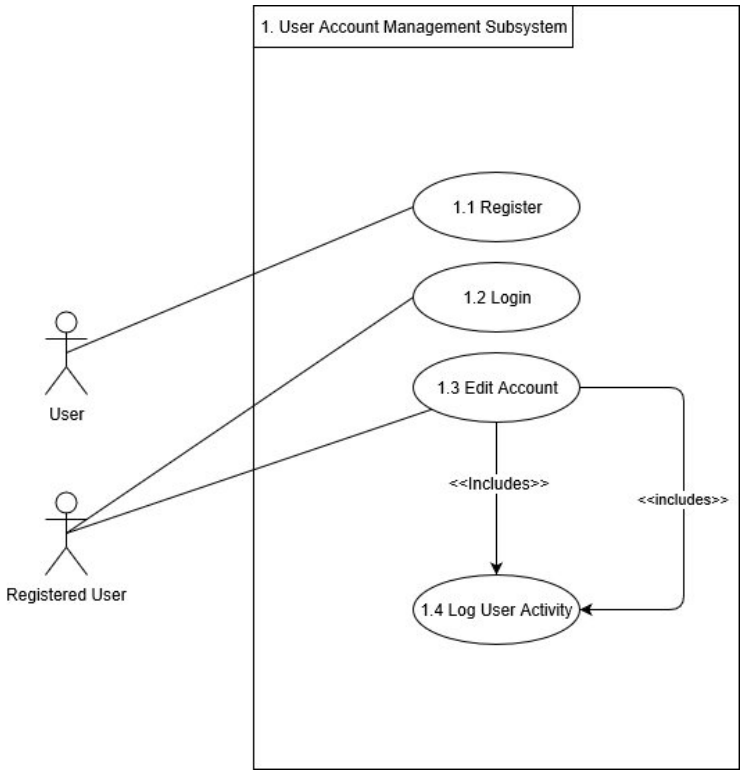

Figure 1: User Account Management Subsystem

2. Code Management Subsystem

   (a) Read in Code

      i. **Description:** The Code Management subsystem should permit the user to either upload a code-based executable or type in code for further processing.

      ii. **Precondition:** The user must have internet access and an active account. He/she should be logged into the system. There should be a user interface that will allow the user to either upload code or type it in.

      iii. **Postcondition:** The provided code should be uploaded and loaded into the system for further processing.

   (b) Check Code

      i. **Description:** The Code Management subsystem should check if the code provided is supported.

ii. **Precondition:** The user must have internet access and an active account. He/she should be logged into the system. Code should be uploaded to be checked.

iii. **Postcondition:** The provided code should either be stored for compilation if supported or an error should be clearly shown if it isn't.

(c) Compile Code

i. **Description:** The Code Management subsystem should be able to compile code for execution.

ii. **Precondition:** The user must have internet access and an active account. He/she should be logged into the system. Code should be uploaded to be checked and validated.

iii. **Postcondition:** The provided code should either be compiled to machine code or interpreted to intermediate code. This depends on the language selected for benchmarking. Failure and syntax errors should also be reported and the program should handle those failures accordingly.

(d) Run Code

i. **Description:** The Code Management subsystem should be able to execute code and use the output in a meaningful way.

ii. **Precondition:** The user must have internet access and an active account. He/she should be logged into the system. Code should be uploaded to be checked, validated and compiled accordingly.

iii. **Postcondition:** Code should be executed and benchmarked after successful compilation according to the user-selected criteria and available options. If the execution failed, the user should be notified and the user should be given the option to either retry or use other available options on the system.

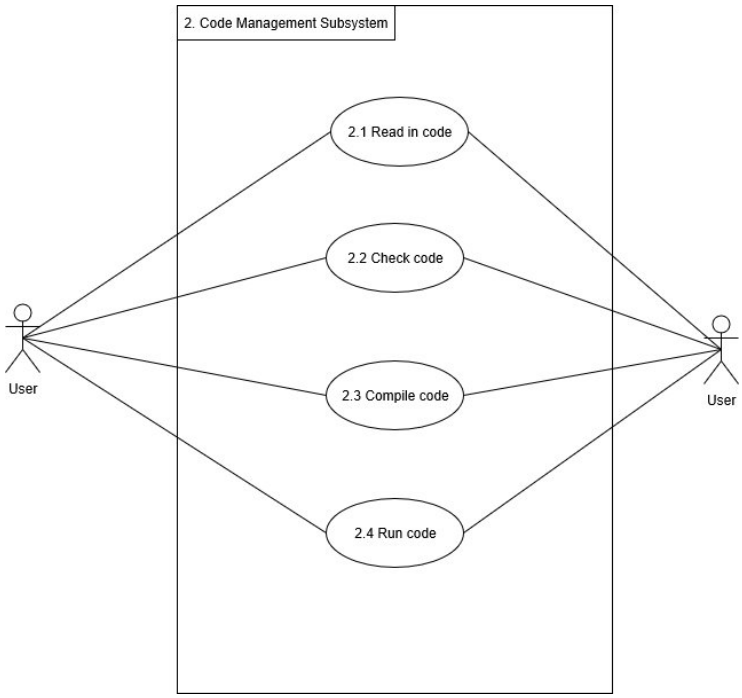

Figure 2: Code Management Subsystem

3. Hardware Management Subsystem

   (a) Add Computer

      i. **Description:** The system should allow a user subscribed to the benchmarking service to add a computer that will be usable by the individual to perform a benchmark.

      ii. **Precondition:** The user must be an active subscriber to the benchmarking service as well as have access to the internet.

      iii. **Postcondition:** The system accepts the newly added computer and makes it available to the individual for benchmarking.

   (b) Add Computer Component

      i. **Description:** The system should allow the user to add new computer components to the already existing computer on the benchmarking service, this can be characterized as adding more ram to a computer or adding a Bluetooth adapter to a computer.

      ii. **Precondition:** The user must be an active subscriber to the benchmarking service as well as have access to the internet.

      iii. **Postcondition:** The system accepts the newly added compute component and makes it available to the individual for bench-

marking.

(c) Remove Computer

i. **Description:** The system should allow a user to remove a computer they added to the benchmarking service.

ii. **Precondition:** The user must be an active subscriber to the benchmarking service as well as have access to the internet.

iii. **Postcondition:** The service removes the relevant computer from the benchmarking service.

(d) Remove Computer Component

i. **Description:** The system should allow a user to remove a computer component they added from the benchmark service.

ii. **Precondition:** The user must be an active subscriber to the benchmarking service as well as have access to the internet.

iii. **Postcondition:** The system removes the relevant computer component from the benchmarking service.

(e) Request Hardware

i. **Description:** The system should allow the user to make a request to use hardware available on the service, or for additional resources, these additional resources can be computers or computer components, a request can also be for a hardware configuration change.

ii. **Precondition:** The user must be an active subscriber to the benchmarking service as well as have access to the internet.

iii. **Postcondition:** The system accepts the request and the alerts the relevant parties of the hardware request. The user is then put on a queue if the resources is currently unavailable and is alerted once available.

(f) Test/Verify Hardware

i. **Description:** The system should allow the users to test and verify that the hardware is functioning as intended and that the hardware is available.

ii. **Precondition:** The user must be an active subscriber to the benchmarking service as well as have access to the internet.

iii. **Postcondition:** The system runs checks for hardware failures as well as the availability of the hardware; the user is then alerted of the findings.

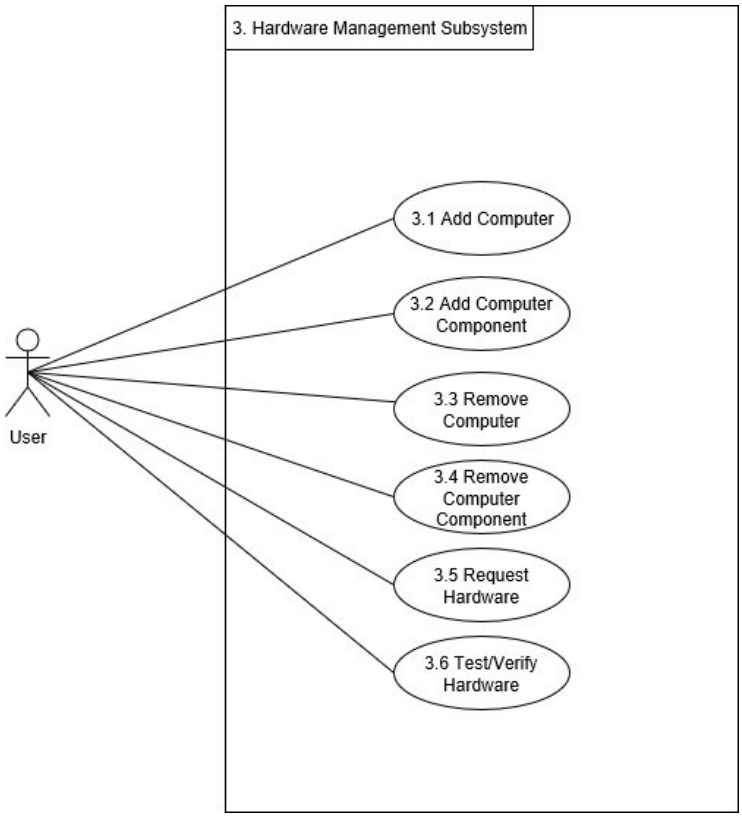

Figure 3: Hardware Management Subsystem

4. Hardware Monitoring Subsystem

   (a) CPU Monitoring

      i. **Description:** The Hardware Monitoring subsystem should be able to analyze CPU usage for each benchmarking request. It should be able to report that usage in real-time and as profiling information. It should also be able to report hardware failure.

      ii. **Precondition:** The user must have an active user account. The user must have specified a CPU for their benchmarking request through the Hardware Management subsystem. The user must also be on the correct page of the interface to view the real-time CPU usage and a benchmark request must be running.

      iii. **Postcondition:** CPU usage is reported to the user interface in real time and stored in a report for the user to view at a later time. If no CPU was specified or another error occurred, then the appropriate error message is displayed.

(b) Memory Usage Monitoring

i. **Description:** The Hardware Monitoring subsystem should be able to analyze both memory and storage usage for each benchmarking request. It should be able to report the usage in real-time and as profiling information. It should also be able to report hardware failure.

ii. **Precondition:** The user must have an active user account. The user must have specified memory and storage for their benchmarking request. The user must also be on the correct page of the user interface to view real-time memory usage and a benchmark request must be running.

iii. **Postcondition:** Memory usage is reported in real-time and stored in a report for the user to view at a later time. If no memory or storage is specified or another error occurred, then the appropriate error message is displayed.

(c) Power Consumption Monitoring

i. **Description:** The Hardware Monitoring subsystem should be able to analyze power consumption for each benchmarking request. It should be able to report the consumption in real-time and as profiling information. It should also be able to report hardware failure.

ii. **Precondition:** The user must have an active user account. The user must specify that they would like power consumption to be recorded. The user must also be on the correct page of the interface to view real-time power consumption and a benchmark request must be running. The machine running the benchmark must have the appropriate monitoring tools.

iii. **Postcondition:** Power consumption is reported in real-time and stored in a report for the user to view later. If the user does not specify that they would like power consumption to be reported, then no information about power consumption is displayed or stored.

(d) Heat Generation Monitoring

i. **Description:** The Hardware Monitoring subsystem should be able to analyze heat generation from the machine that a user's benchmarking request is being run on. It should be able to report that heat generation both in real-time and as profiling information. It should also be able to report hardware failure.

ii. **Precondition:** The user must have an active user account. The user must specify that they would like heat generation to be recorded. The machine running the benchmark must have the appropriate sensors. The user must also be on the correct page of

the interface to view real-time heat generation and a benchmark request must be running.

iii. **Postcondition:** Heat generation is reported in real-time and stored in a report for the user to view later. If the user does not specify that they would like heat generation to be reported, then no information about heat generation is displayed or stored.

(e) Elapsed Time Monitoring

i. **Description:** The Hardware Monitoring subsystem should be able to report the elapsed time of a benchmarking request both in real-time and as profiling information. It should also be able to report hardware failure.

ii. **Precondition:** The user must have an active user account. The user must be on the correct page of the interface to view real-time elapsed time and a benchmark request must be running.

iii. **Postcondition:** Elapsed time is always reported in real-time and stored in a report for the user to view later. If the user does not specify that they would like elapsed time to be reported, then no information about elapsed time is displayed or stored.

(f) Network Monitoring

i. **Description:** The Hardware Monitoring subsystem should be able to report network usage of benchmarking requests both in real-time and as profiling information. It should also be able to report hardware failure.

ii. **Precondition:** The user must have an active user account. The user must provide code that requires a network connection. The machine running the benchmark must have the appropriate network connection. The user must be on the correct page of the interface to view the real-time usage and a benchmark request must be running.

iii. **Postcondition:** Network usage is reported in real-time and stored in a report for the user to view later. If the user does not specify that they would like network usage to be reported, then no information about network usage is displayed or stored.

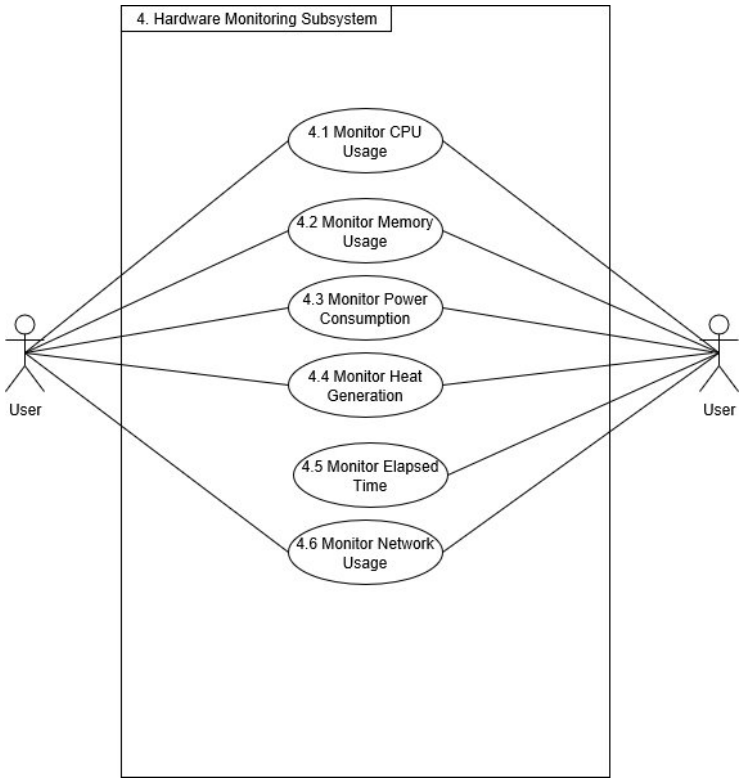

Figure 4: Hardware Monitoring Subsystem

5. Manager Subsystem

   (a) Create User

      i. **Description:** The purpose of the create user functionality is to offer manager users the capability to create new users on the system. These users can be any type of user, with the exception that a user needs to have the right permissions to create said user as per the roles and permissions matrix.

      ii. **Precondition:** User has not been created. No other user exists on the system with same unique attributes (e.g. email, cellphone number, etc). Manager has permissions to create a new user with said attributes and properties.

      iii. **Postcondition:** User successfully created on the system and can commence using the system. User creation event successfully logged on the system.

   (b) Update User

i. **Description:** The purpose of the update user functionality is to offer manager users the capability to modify attributes and/or properties of existing users on the system.

ii. **Precondition:** User already exists on the system. New user information does not contravene any attribute/property uniqueness constraints on the system.Manager has permissions to modify said user.

iii. **Postcondition:** User successfully updated on the system and changes are immediately reflected. User update event successfully logged on the system.

(c) View User

i. **Description:** The purpose of the view user functionality is to offer manager users the capability to view the attributes and/or properties of users currently existing on the system.

ii. **Precondition:** User exists on the system. Manager has permissions to view said user.

iii. **Postcondition:** Manager can successfully view attributes and/or properties of said user. User view event successfully logged on the system.

(d) Delete User

i. **Description:** The purpose of the view user functionality is to offer manager users the capability to view the attributes and/or properties of users currently existing on the system.

ii. **Precondition:** User already exists on the system. Manager has permissions to delete said user.

iii. **Postcondition:** User successfully delete from the system and and changes are immediately reflected, i.e. deleted user should not be able to use the system.User deletion event successfully logged on the system.

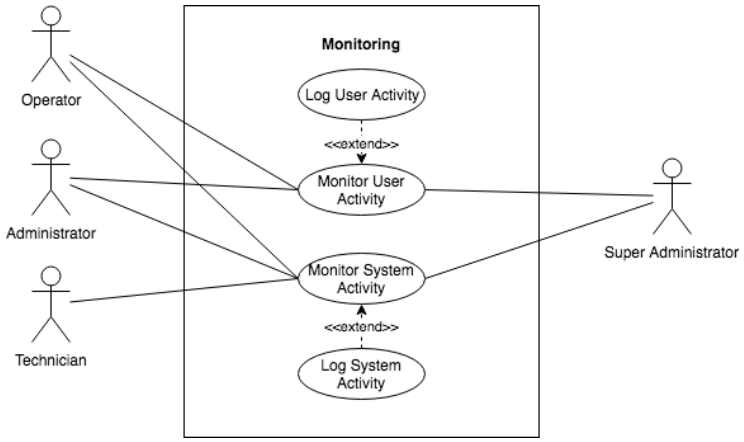

Figure 5: Manager Subsystem part 1

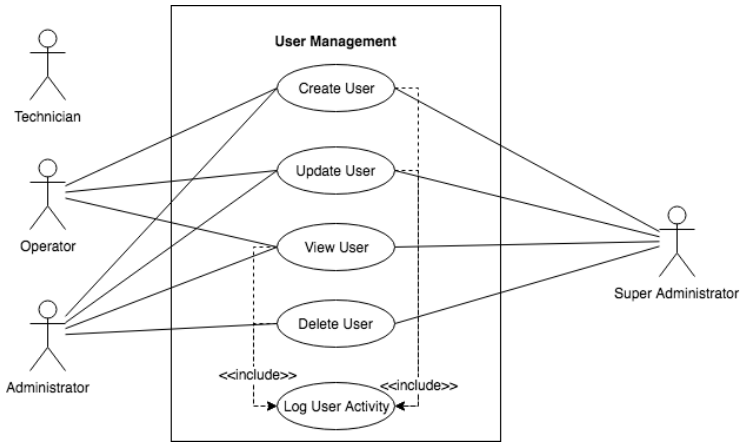

Figure 6: Manager Subsystem part 2

3.3 Actor-System Interaction Modeling

1. User Account Management Subsystem

   (a) Login

Preconditions:	Valid credentials of a registered user are submitted
Actor: User	**System:** Benchmarking Service
	0. The system displays the home page
1. The user selects "Sign in" option	2. The system displays the sign in page
3. User enters their email address and password	4. The system displays the user's homepage
Postconditions:	The user receives registration confirmation

   (b) Register

Preconditions:	No other user exists on the system with same email address and the email address is valid
Actor: Registered User	**System:** Benchmarking Service
	0. The system displays the home page
1. The user selects "Sign up" option	2. The system displays the sign up page
3. User enters their email address and chooses a password and confirms it and selects sign up	4. The system displays registration confirmation page
	5. The system sends user confirmation email
Postconditions:	User successfully logged into the system and a user logged in event is logged on the system.

   (c) Edit Account

Preconditions:	User is logged in, the new user information does not contravene any attribute/property uniqueness constraints on the system and if the update is/includes a display picture the file type of the new display picture is supported
Actor: Registered User	**System:** Benchmarking Service
	0. The system displays the user's home page
1. The user selects "Edit account" option	2. The system displays the edit account page
3. User edits attributes and selects "submit"	4. The system updates user's attributes and changes reflect
Postconditions:	User successfully updated on the system and changes are immediately reflected and a user logged in event is logged on the system.

2. Code Management Subsystem

   (a) Read in code

Preconditions:	The user must have internet access and an active account. He/she should be logged into the system. There should be a user interface that will allow the user to either upload code or type it in.
Actor: System	**System:** Benchmarking System.
0. User uploads/types in code.	1. Code uploaded or typed in and read.
Postconditions:	The user must be registed and logged in within the benchmark service in order for him to upload or type the source code.

   (b) Check Code

Preconditions:	The user must have internet access and an active account. He/she should be logged into the system. Code should be uploaded to be checked.
Actor: System	**System:** Benchmarking System.
0. User uploads/types in code.	1. Code is checked to find out whether it is supported.
Postconditions:	The provided code should either be stored for compilation if supported or an error should be clearly shown if it isn't.

(c) Compile Code

Preconditions:	The user must have internet access and an active account. He/she should be logged into the system. Code should be uploaded to be checked and validated.
Actor: System	**System:** Benchmarking System.
0. User uploads code.	1. Code is checked to find out if it is supported.
2. Code is compiled if support check is successful.	
Postconditions:	The provided code should either be compiled to machine code or interpreted to intermediate code. This depends on the language selected for benchmarking. Failure and syntax errors should also be reported and the program should handle those failures accordingly.

(d) Run Code

Preconditions:	The user must have internet access and an active account. He/she should be logged into the system. Code should be uploaded to be checked, validated and compiled accordingly.
Actor: System	**System:** Benchmarking System.
0. User uploads code.	1. Code is checked to find out if it is supported.
2. Code is compiled if support check is successful.	3. Code is executed.
Postconditions:	Code should be executed and benchmarked after successful compilation according to the user-selected criteria and available options. If the execution failed, the user should be notified and the user should be given the option to either retry or use other available options on the system.

3. Hardware Management Subsystem

(a) Add Computer

Preconditions	The user must be an active subscriber to the benchmarking service as well as have access to the internet.
Actor: User	**System:** Benchmarking System
0. User will login, then choose to the add computer option	1. The system accepts the new computer and alerts the user of the successful addition of hardware.
2. The user acknowledges the alert.	
Postconditions	The system accepts the newly added computer and makes it available to the individual for benchmarking.

(b) Add Computer Component

Preconditions	The user must be an active subscriber to the benchmarking service as well as have access to the internet.
Actor: User	**System:** Benchmarking System
0. User makes the hardware component change.	
1. User will login, then choose to the add computer component option	2. The system accepts the new computer component and alerts the user of the successful addition of hardware.
3. The user acknowledges the alert.	
Postconditions	The system accepts the newly added compute component and makes it available to the individual for benchmarking.

(c) Remove Computer

Preconditions	The user must be an active subscriber to the benchmarking service as well as have access to the internet.
Actor: User	**System:** Benchmarking System
0. User will login, then choose to the remove computer option	1. The system accepts the new change and alerts the user of the successful removal of the computer.
2. The user acknowledges the alert.	
Postconditions	The system removes the relevant computer from the benchmarking service.

(d) Remove Computer Component

Preconditions	The user must be an active subscriber to the benchmarking service as well as have access to the internet.
Actor: User	**System:** Benchmarking System
0. User makes the hardware component change.	
1. User will login, then choose to the remove computer component option	2. The system accepts the removal of the computer component and alerts the user of the successful removal of hardware.
3. The user acknowledges the alert.	
Postconditions	The service removes the relevant computer component from the benchmarking service.

(e) Request Hardware

Preconditions	The user must be an active subscriber to the benchmarking service as well as have access to the internet.
Actor: User	**System:** Benchmarking System
0. The user will login and make a request for hardware or configuration change.	1. The system accepts the request, and alerts the user of the successful hardware change.
2. The user acknowledges the alert.	
Postconditions	The system accepts the request and the alerts the relevant parties of the hardware request. The user is then put on a queue if the resources is currently unavailable and is alerted once available.

(f) Test/Verify Hardware

Preconditions	The user must be an active subscriber to the benchmarking service as well as have access to the internet.
Actor: User	**System:** Benchmarking System
0. The user will login and make requests.	1. The sytem accepts the requests.
2. The user interacts with the hardware.	
Postconditions	The service runs checks for hardware failures as well as the availability of the hardware that was requested.

4. Hardware Monitoring Subsystem

   (a) CPU Usage Monitoring

Preconditions:	The user must have an active user account. The user must have specified a CPU for their benchmarking request through the Hardware Management subsystem
Actor: User	**System:** Benchmarking System
0. User runs benchmarking request	1. System monitors and records CPU usage real-time
2. User views CPU usage real-time	
	3. System creates and stores the profiling information for the CPU usage of the benchmarking request
4. User views profiling information	
Postconditions:	CPU usage is reported to the user interface in real time and stored in a report for the user to view later. If no CPU was specified or another error occurred, then the appropriate error message is displayed.

   (b) Memory Usage Monitoring

Preconditions:	The user must have an active user account. The user must have specified memory and storage for their benchmarking request
Actor: User	**System:** Benchmarking System
0. User runs benchmarking request	1. System monitors and records memory usage real-time
2. User views memory usage real-time	
	3. System creates and stores the profiling information for the memory usage of the benchmarking request
4. User views profiling information	
Postconditions:	Memory usage is reported in real-time and stored in a report for the user to view later. If no memory or storage is specified or another error occurred, then the appropriate error message is displayed

(c) Power Consumption Monitoring

Preconditions:	The user must have an active user account. The user must specify that they would like power consumption to be recorded and The machine running the benchmark must have the appropriate monitoring tools
Actor: User	**System:** Benchmarking System
0. User runs benchmarking request	1. System monitors and records power consumption real-time
2. User views power consumption real-time	
	3. System creates and stores the profiling information for the power consumption of the benchmarking request
4. User views profiling information	
Postconditions:	Power consumption is reported in real-time and stored in a report for the user to view later. If the user does not specify that they would like power consumption to be reported, then no information about power consumption is displayed or stored.

(d) Heat Generation Monitoring

Preconditions:	The user must have an active user account. The user must specify that they would like heat generation to be recorded. The machine running the benchmark must have the appropriate sensors.
Actor: User	**System:** Benchmarking System
0. User runs benchmarking request	1. System monitors and records heat generation real-time
2. User views heat generation real-time	
	3. System creates and stores the profiling information for the heat generation of the benchmarking request
4. User views profiling information	
Postconditions:	Heat generation is reported in real-time and stored in a report for the user to view later. If the user does not specify that they would like heat generation to be reported, then no information about heat generation is displayed or stored. If something went wrong, the appropriate error message is sent out.

(e) Elapsed Time Monitoring

Preconditions:	The user must have an active user account
Actor: User	**System:** Benchmarking System
0. User runs benchmarking request	1. System monitors and records elapsed time real-time
2. User views elapsed time real-time	
	3. System creates and stores the profiling information for the elapsed time of the benchmarking request
4. User views profiling information	
Postconditions:	Elapsed time is always reported in real-time and stored in a report for the user to view later. If the user does not specify that they would like elapsed time to be reported, then no information about elapsed time is displayed or stored. If something went wrong, the appropriate error message is sent out.

(f) Network Usage Monitoring

Preconditions:	The user must have an active user account. The user must provide code that requires a network connection. The machine running the benchmark must have the appropriate network connection.
Actor: User	**System:** Benchmarking System
0. User runs benchmarking request	1. System monitors and records network usage real-time
2. User views network usage real-time	
	3. System creates and stores the profiling information for the network usage of the benchmarking request
4. User views profiling information	
Postconditions:	Network usage is reported in real-time and stored in a report for the user to view later. If the user does not specify that they would like network usage to be reported, then no information about network usage is displayed or stored. If something went wrong the appropriate error message is sent out.

5. Manager Subsystem

   (a) Create User

Preconditions:	User has not been created. No other user exists on the system with same unique attributes (e.g. email, cellphone number, etc) Manager has permissions to create a new user with said attributes and properties.
Actor: User	**System:** Benchmarking System
	0. The system displays the create user page.
1. The user inputs the information required for the user.	2. The system validates the users input.
2. The user clicks submit.	3. The system creates the user on the system.
Post Conditions:	User successfully created on the system and can commence using the system. User creation event successfully logged on the system.

   (b) Update User

Preconditions:	User already exists on the system. New user information does not contravene any attribute/property uniqueness constraints on the system. Manager has permissions to modify said user
Actor: User	**System:** Benchmarking System
	0. The system displays the edit user page.
1. The user modifies the information for the user.	2. The system validates the users input.
2. The user clicks submit.	3. The system updates the user on the system.
Post Conditions:	User successfully updated on the system and changes are immediately reflected. User update event successfully logged on the system.

(c) View User

Preconditions:	User exists on the system. Manager has permissions to view said user.
Actor: User	**System:** Benchmarking System
	0. The system displays the users page.
1. The user selects the user they wish to view.	2. The system returns the information of the user.
Post Conditions:	Manager can successfully view attributes and/or properties of said user. User view event successfully logged on the system.

(d) Delete User

Preconditions:	User already exists on the system. Manager has permissions to delete said user.
Actor: User	**System:** Benchmarking System
	0. The system displays the users page.
1. The user selects the user they wish to delete.	2. The system prompts the user for delete confirmation.
3. The user clicks confirm action.	4. The system deletes the user from the database.
Post Conditions:	User successfully delete from the system and and changes are immediately reflected, i.e. deleted user should not be able to use the system. User deletion event successfully logged on the system.

(e) User Activity Monitoring

Preconditions:	System events have been successfully logged on the system.
Actor: User	**System:** Benchmarking System
	0. The system displays the historical system activity log.
1. The user views and filters historical system log results on the screen.	
Post Conditions:	Manager user can successfully view and filter historical system events.

(f) Log User Activity

Preconditions:	User is about to perform an action. User action has not yet been logged.
Actor: User	**System:** Benchmarking System
	0. The system listens for user activity.
1. The user performs an actions.	2. The system records the action in the activity logs.
Post Conditions:	User action is successfully logged.

(g) Log System Activity

Preconditions:	System action has not yet been logged.
	System: Benchmarking System
	0. The system listens for user activity.
	2. The system performs an action.
	2. The system records the action in the activity logs.
Post Conditions:	System action is successfully logged