

University of Ottawa

Assignment 2:
OCSF and SimpleChat

Feyi Adesanya and Ethan Plant

SEG 2105: Introduction to Software Engineering

Professor Miguel A. Garzón

October 28, 2020

Part 1 Question 3

The jstack command prints Java stack traces of Java thread for a specific Java process and the thread dump is a snapshot of the state of the threads of the specified Java process. For answering this question we will only focus on the non-daemon threads, which are the threads being used by the program

*Used the commands for netstat and lsof commands for mac according to:
<https://www.lifewire.com/using-netstat-command-on-mac-4176069>

Feyi Adesanya's JPS and JSTAT output



```
11395 Jps
11350
11372 EchoServer
11373 ClientConsole
11375 ClientConsole
```

Output of using the jps -v command

```

jstackOutput2.txt

"Thread-0" #14 prio=5 os_prio=31 cpu=204.00ms elapsed=171.12s tid=0x00007fc952850000 nid=0x6203 runnable [0x000070000567c000]
    java.lang.Thread.State: RUNNABLE
        at sun.nio.ch.Net.poll(java.base@13.0.1/Native Method)
        at sun.nio.ch.NioSocketImpl.park(java.base@13.0.1/NioSocketImpl.java:182)
        at sun.nio.ch.NioSocketImpl.timedAccept(java.base@13.0.1/NioSocketImpl.java:710)
        at sun.nio.ch.NioSocketImpl.accept(java.base@13.0.1/NioSocketImpl.java:752)
        at java.net.ServerSocket.implAccept(java.base@13.0.1/ServerSocket.java:662)
        at java.net.ServerSocket.platformImplAccept(java.base@13.0.1/ServerSocket.java:628)
        at java.net.ServerSocket.implAccept(java.base@13.0.1/ServerSocket.java:604)
        at java.net.ServerSocket.implAccept(java.base@13.0.1/ServerSocket.java:561)
        at java.net.ServerSocket.accept(java.base@13.0.1/ServerSocket.java:518)
        at ocsf.server.AbstractServer.run(AbstractServer.java:320)
        at java.lang.Thread.run(java.base@13.0.1/Thread.java:830)

"DestroyJavaVM" #15 prio=5 os_prio=31 cpu=456.74ms elapsed=171.12s tid=0x00007fc95210a000 nid=0x2303 waiting on condition
[0x0000000000000000]
    java.lang.Thread.State: RUNNABLE

"Thread-1" #16 prio=5 os_prio=31 cpu=55.55ms elapsed=165.94s tid=0x00007fc95304b800 nid=0xe07 runnable [0x000070000577f000]
    java.lang.Thread.State: RUNNABLE
        at sun.nio.ch.SocketDispatcher.read0(java.base@13.0.1/Native Method)
        at sun.nio.ch.SocketDispatcher.read(java.base@13.0.1/SocketDispatcher.java:47)
        at sun.nio.ch.NioSocketImpl.tryRead(java.base@13.0.1/NioSocketImpl.java:262)
        at sun.nio.ch.NioSocketImpl.implRead(java.base@13.0.1/NioSocketImpl.java:313)
        at sun.nio.ch.NioSocketImpl.read(java.base@13.0.1/NioSocketImpl.java:351)
        at sun.nio.ch.NioSocketImpl$1.read(java.base@13.0.1/NioSocketImpl.java:802)
        at java.net.Socket$SocketInputStream.read(java.base@13.0.1/Socket.java:937)
        at java.net.Socket$SocketInputStream.read(java.base@13.0.1/Socket.java:932)
        at java.io.ObjectInputStream$PeekInputStream.peek(java.base@13.0.1/ObjectInputStream.java:2723)
        at java.io.ObjectInputStream$BlockDataInputStream.peek(java.base@13.0.1/ObjectInputStream.java:3050)
        at java.io.ObjectInputStream$BlockDataInputStream.peekByte(java.base@13.0.1/ObjectInputStream.java:3060)
        at java.io.ObjectInputStream.readObject0(java.base@13.0.1/ObjectInputStream.java:1561)
        at java.io.ObjectInputStream.readObject(java.base@13.0.1/ObjectInputStream.java:430)
        at ocsf.server.ConnectionToClient.run(ConnectionToClient.java:213)

"Thread-2" #17 prio=5 os_prio=31 cpu=1.51ms elapsed=162.98s tid=0x00007fc951819800 nid=0x9503 runnable [0x0000700005882000]
    java.lang.Thread.State: RUNNABLE
        at sun.nio.ch.SocketDispatcher.read0(java.base@13.0.1/Native Method)
        at sun.nio.ch.SocketDispatcher.read(java.base@13.0.1/SocketDispatcher.java:47)
        at sun.nio.ch.NioSocketImpl.tryRead(java.base@13.0.1/NioSocketImpl.java:262)
        at sun.nio.ch.NioSocketImpl.implRead(java.base@13.0.1/NioSocketImpl.java:313)
        at sun.nio.ch.NioSocketImpl.read(java.base@13.0.1/NioSocketImpl.java:351)
        at sun.nio.ch.NioSocketImpl$1.read(java.base@13.0.1/NioSocketImpl.java:802)
        at java.net.Socket$SocketInputStream.read(java.base@13.0.1/Socket.java:937)
        at java.net.Socket$SocketInputStream.read(java.base@13.0.1/Socket.java:932)
        at java.io.ObjectInputStream$PeekInputStream.peek(java.base@13.0.1/ObjectInputStream.java:2723)
        at java.io.ObjectInputStream$BlockDataInputStream.peek(java.base@13.0.1/ObjectInputStream.java:3050)
        at java.io.ObjectInputStream$BlockDataInputStream.peekByte(java.base@13.0.1/ObjectInputStream.java:3060)
        at java.io.ObjectInputStream.readObject0(java.base@13.0.1/ObjectInputStream.java:1561)
        at java.io.ObjectInputStream.readObject(java.base@13.0.1/ObjectInputStream.java:430)
        at ocsf.server.ConnectionToClient.run(ConnectionToClient.java:213)

"Attach Listener" #18 daemon prio=9 os_prio=31 cpu=1.58ms elapsed=0.12s tid=0x00007fc951849800 nid=0x6403 waiting on condition
[0x0000000000000000]
    java.lang.Thread.State: RUNNABLE

```

Output of the jstack command, with only the non-daemon processes (processes related to the program)

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE NAME
eclipse	11350	feyiadesanya	85u	IPv6	0xb4f7457bb89e7487	0t0	TCP localhost:59339->localhost:59340 (ESTABLISHED)
eclipse	11350	feyiadesanya	145u	IPv6	0xb4f7457bb89e4687	0t0	TCP feyiadesanya.hitronhub.home:59338->www.eclipse.org:https (CLOSE_WAIT)
java	11372	feyiadesanya	6u	IPv4	0xb4f7457bc2042eb7	0t0	TCP localhost:59340->localhost:59339 (ESTABLISHED)
java	11372	feyiadesanya	10u	IPv6	0xb4f7457bb89e57c7	0t0	TCP *:6992 (LISTEN)
java	11372	feyiadesanya	11u	IPv6	0xb4f7457bb89e40c7	0t0	TCP localhost:6992->localhost:59343 (ESTABLISHED)
java	11372	feyiadesanya	13u	IPv6	0xb4f7457bb89e4c47	0t0	TCP localhost:6992->localhost:59344 (ESTABLISHED)
java	11373	feyiadesanya	6u	IPv6	0xb4f7457bb89e6347	0t0	TCP localhost:59343->localhost:6992 (ESTABLISHED)
java	11375	feyiadesanya	6u	IPv6	0xb4f7457bb89e6ec7	0t0	TCP localhost:59344->localhost:6992 (ESTABLISHED)

Output of lsof -i command (equivalent to netstat -a) (Only displaying the ones relevant to the program)

Analysis:**Jps:**

Shows each process identifier (ID) for each JVM process

Jstat:

Thread - 0 Analysis

This thread represents the server, as stated where The state of this thread is RUNNABLE, which means the thread is occupying CPU and processing a task.

Thread - 1 Analysis

This thread represents one of the clients. The state of this thread is RUNNABLE, which means the thread is occupying CPU and processing a task.

Thread - 2 Analysis

This thread represents another client. The state of this thread is RUNNABLE, which means the thread is occupying CPU and processing a task.

Lsof (Netstat Equivalent):

We can see that the process IDs here correspond to the same ones we saw after executing the jstat command. The lsof -i command is equivalent to the netstat -a command with the local and foreign address being displayed under Name with an arrow between them. The process ID representing the server (11372) showcases port being established and listening for clients. The process ID's representing the two clients (11373 and 11375) can be seen connecting to the host we established 6992 (6XXX), along with the host connecting back to them (localhost 59343 and 59344).

Part 2:

Question 1 - Produce 6 use cases for the system

Use Case 1

Case Name	Presto Top Up
Actors:	Passenger
Goals	User wants to add funds to their presto card
Preconditions	Passenger notices the alert they received from the system on their presto card funds
Summary	<p>Passenger is alerted that they lack funds in their presto, which will prevent them from using public transport unless they add more.</p> <p>Assumptions: employees and administrators will use passenger accounts when wanting to use the public transport system themselves</p>
Normal Flow	<p>User Action</p> <ol style="list-style-type: none"> 1. Passenger receives and acknowledges to add funds to their presto card <p>System Response</p> <ol style="list-style-type: none"> 2. System prompts Passenger with the option to top up now or at a later time of their choosing <p>User Action</p> <ol style="list-style-type: none"> 3. Passenger chooses to top up now <p>System Response</p> <ol style="list-style-type: none"> 4. System brings user to a transaction page and ask for their payment details <p>User Action</p> <ol style="list-style-type: none"> 5. User inputs payment details <p>System Response</p> <ol style="list-style-type: none"> 6. System modifies the user's Presto Card shows changes in funds
Post-Conditions	Card now has enough funds to fund at minimum one ride on public transportation

Alternative Flow	<p>User Action</p> <ol style="list-style-type: none"> 1. Passenger decides to update their funds later 2. Doesn't ride public transport until they top up their presto card <p>System Response</p> <ol style="list-style-type: none"> 3. System will keep prompting the Passenger at regular intervals to add funds to their presto card
------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Use Case 2

Case Name	Passenger Route
Actors:	Passenger
Goals	Passenger needs to find a route to their destination
Preconditions	Passenger has a destination in mind and can only access it through public transportation
Summary	<p>Passenger checks all the possible routes to their specified destination and picks the most ideal route</p> <p>Assumptions: the most ideal route is the fastest in this case but many things might be taken into account by a passenger such as road conditions and stops they may make</p> <p>Passenger will leave immediately</p>
Normal Flow	<p>User Action</p> <ol style="list-style-type: none"> 1. Passenger decides on a destination and inputs their destination into the system <p>System Response</p> <ol style="list-style-type: none"> 2. The system locates all possible routes using all forms of public transport (trains, bus, and walking)

	<p>3. System displays the possible routes a user could pick from</p> <p>User Action</p> <p>4. Passenger picks the fastest route</p> <p>System Response</p> <p>5. System displays the route over a map of the city and displays the approximate time it will take to go through their specified route</p> <p>6. System gives user option on when they want to leave and shows the time they will leave and arrive</p> <p>User Action</p> <p>7. Passenger accepts and leaves immediately</p> <p>System Response</p> <p>8. As the Passenger travels to their destination, the system tracks their movements and provides instructions about the route</p>
Post-Conditions	User reaches their destination
Alternative Steps	<ol style="list-style-type: none"> 1. System is unable to locate a reasonable route for the Passenger 2. System displays this as an error message and provides other options a Passenger can take to reach their destination 3. Passenger chooses one of the other options

User Case 3

Case Name	Covid-19 Case Alert
Actors:	Passenger and Employees
Goals	Covid-19 case was identified and users of the system need to be alerted

Preconditions	Passenger or employee with Covid-19 confirms to the administrations that they went on a particular route will having the disease
Summary	<p>Systems tracks and updates users and employees on potential Covid-19 cases that were identified on their routes travelled</p> <p>Assumptions: Confirmed Covid-19 cases are reported by passengers or employees immediately when they receive news of getting the disease</p>
Normal Flow	<p>System Response</p> <ol style="list-style-type: none"> 1. Covid-19 Case is identified and reported to the system 2. System analyzes the route history of the user and identifies all routes that individual took over a period of one week 3. System sends out a message to all users of the system about the Covid-19 case and displays all the routes taken 4. The System specifically identifies user who have taken the particular routes within the specified time period and sends them a special alert 5. This alert detail the possible steps for quarantine and closest testing centers, encourage those displaying symptoms to stay home <p>User Response</p> <ol style="list-style-type: none"> 1. Employees and Passengers see the alert and respond accordingly
Post-Conditions	All users of the system are alerted of the case of Covid-19 and where it originated
Alternative Steps	<ol style="list-style-type: none"> 1. If the system cannot track the route of the affected individual, it will produce a message sent to all users of the system 2. The message will detail that a case of Covid-19 was identified in the public

	<p>transport system</p> <p>3. A special alert will details the possible steps for quarantine and closest testing centers, encourage those displaying symptoms to stay home</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Question 2: UML Diagram

Question 3

A functional requirement is a description of the service that a software must offer and specifies the behavior of the system. Non functional requirements describe how the system will work and optional implementations to the system that can be made. For our case study on the city's transport system, we would implement the following functional requirements and non-functional requirements.

Functional Requirements (10)

User cannot access their account from another device without two factor authentication

Verification email is sent whenever a new user registers for the first time on the software system

Users will be able to enter data about their age, address, education, and payment information

Users who are verified and designated as students, elderly, or government employees will be given free access to public transportation, which will be reflected in the system

Authenticated users and applications have defined access to rights to appropriate resources of the system, which can be overridden by an administrator

When the user pairs the system with a bluetooth enabled device the system will be able to communicate with the tracking and display systems of the device and utilize this to display the app contents appropriately

System will accurately display the date and time of the users local region following the format: month, date, year, time

The system will maintain a service log and, on startup will check, for scheduled services that are due, discontinuing services sessions that are not verified by an administrator

Payment system automatically charges passenger for a ride, once it is completed

Users can access a forum to give information about their complaints, recommendations of the app and public transportation quality

Non Functional Requirements (5) - Quality

The error rate for accepting and verifying users submitting payment details must not exceed 7%, regardless of the payment type they are using (presto, credit card, debit card)

Access to route availability webpage must be available to users 95% of the time during peak business hours EDT

The front page load time must be no more than three seconds for users on a connected on WI-FI or using an LTE mobile connection

The websites attendance limit must be able to support 5,000 users at a time, scalable to 10,000 by next year

Landing page for employees must support 1000 users every hour and have a response time of 5 seconds or less, when used on a Safari, Firefox, or Chrome browser, including the display of text and images

Non Functional Requirements (3) - Platform

The system must support iPhone devices running on versions of OS X 10.5 or higher

The system shall be developed for the Windows, Linux, and Macintosh operating system platforms

It shall be possible for the system to be installed by a user with no previous knowledge of the system but familiar with the operating system of the machine on which it will be installed.

Non Functional Requirements (2) - Process

To operate legally in the city, the system should be licensed with the local transport authority and approved by the city's government

The overall budget for the creation of the system will be subsidized by the city's government and the remainder of the funds will be acquired from shareholders