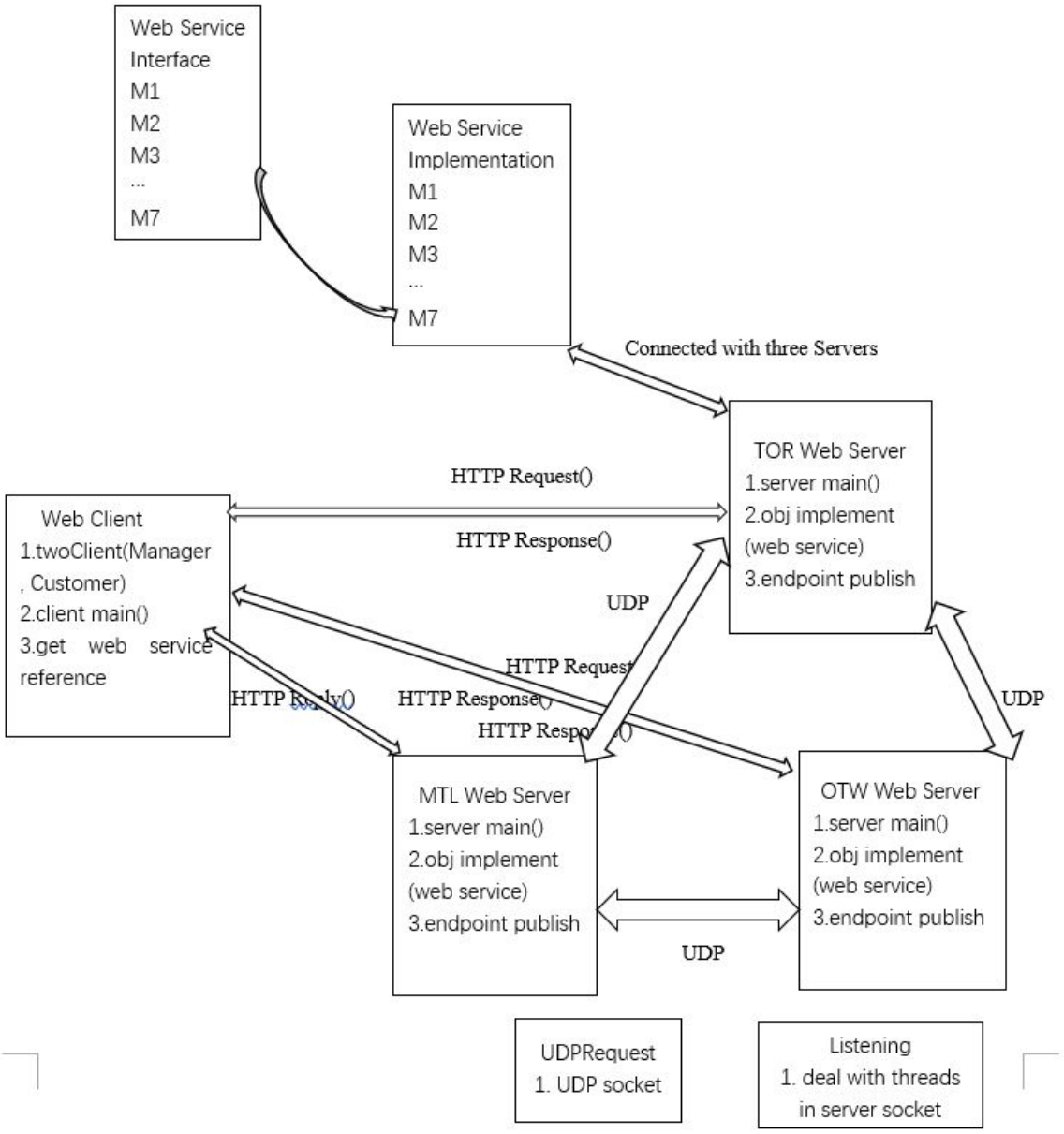
DEMS(Web Service) Design Documentation

StudentID: 40092514 Name: Wenhui Guo
StudentID: 40083064 Name: Yilin Li

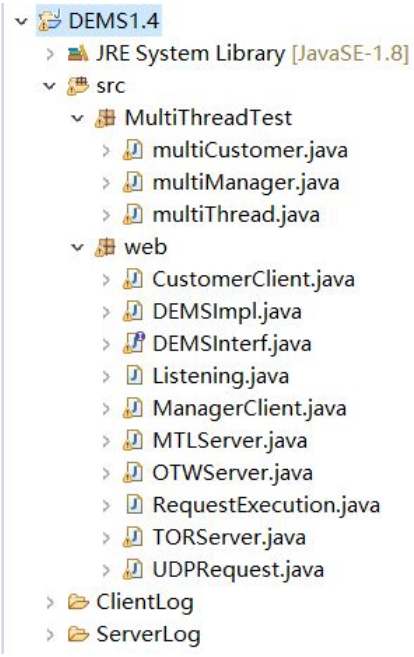
---

1. Techniques and architecture

Techniques: This assignment implements Web Service via standard network protocols SOAP and uses UDP/IP sockets to achieve inter-server communication.



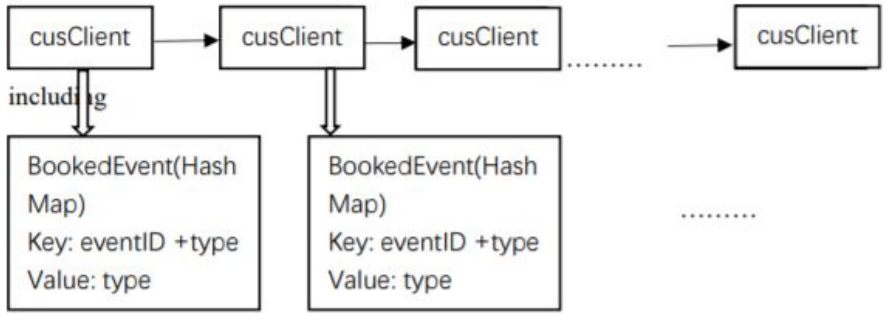
2. Project file structure



(1)Three Servers: MTLServer.java, OTWServer.java, TORServer.java
(2)Two Clients: ManagerClient.java, Customer.java
(3)Interface: DEMSInterf.java
(4)Implementation: DEMSImpl.java
(5)UDP: UDPRequest.java, RequestExection.java, Listening.java
(6)Multithreading test: multiThread.java, multiManager.java, multiCustomer.java
(7)Logs: ClientLog Folder, ServerLog Folder

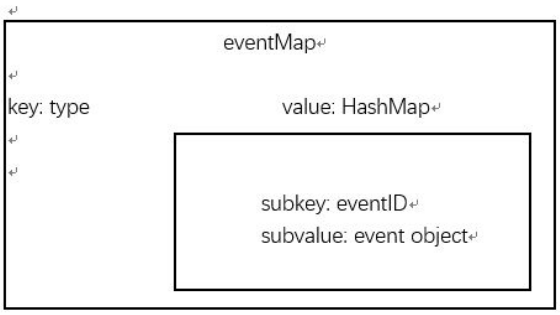
3. Data structure

(1) LinkedList: cusClients
(2) HashMap: bookedEvent for each customerClient



(3) LinkedList: manClients

(4) LinkedList: bookedCus for each event
(5) Two Level HashMap:



4. Important/Difficult part

The important part of this assignment is adapting the DEMS as the web service via standard network protocols SOAP and using UDP/IP sockets to achieve inter-server communication.

About Web Services, they are software components described via WSDL which are capable of being accessed via standard network protocols such as SOAP over HTTP. And SOAP is a protocol which uses XML for message exchange in support of remote method calls over the Internet. First, the client discovers the service and binds to the server. Next, the SOAP request is built to send the arguments which are needed by the service to the server-side. Then SOAP router routes the request to the appropriate server and server unpacks the request, handles it, computes the result and sent back it in the reverse direction.

About UDP programming, in term of sender, create a Datagram socket and bind it to UDP port, place data in byte array, create a datagram packet and specify data array and receiver's address, and invoke the send method with a reference to the packet; in term of receiver, create a datagram socket and bind it to a port, create a byte array to receive the data, create a datagram packet and specify the data array, and invoke the receive method of the socket with a reference the datagram packet.

5. Test case

Test ID	Case description	Function	Precondition	Test steps	Input data	Expected result	Test result	What to test
1	TORManager adds a new local city event	Add event	Starts three servers. Manager login.	Choose "add Event". Input eventID, eventType, capacity, topic	TORA070619 conference 10 IT	Add event successfully	As expected	Add new local city event
2	TORManager adds a	Add event	Starts three servers. Manager login.	Choose "add Event". Input	MTLA070619 conferen	Fail to add event.	As expected	Cannot add another

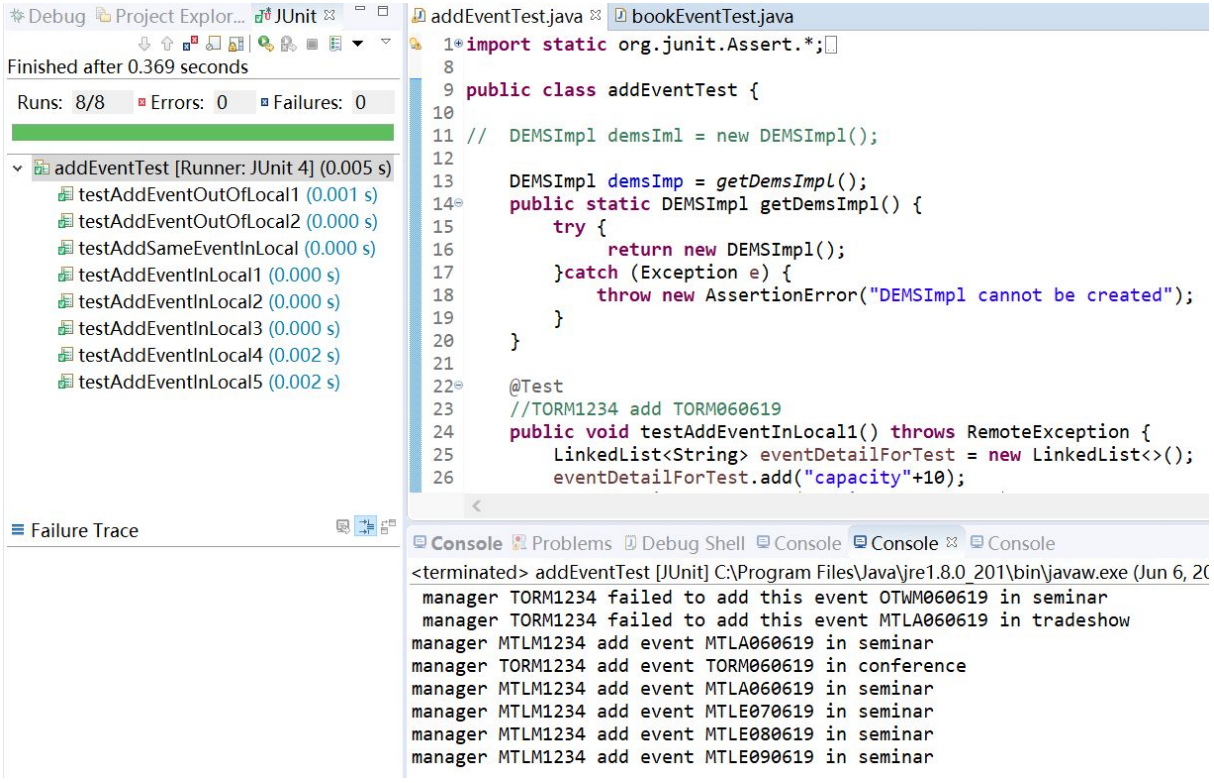
	new another city event			eventID, eventType, capacity, topic	ce 10 IT			city event
3	TORManager adds an existing event	Add event	Starts three servers. Manager login. The event exists.	Choose “add Event”. Input eventID, eventType, capacity, topic	TORA070619 conference 10 IT	Fail to add event. Its capacity plus one.	As expected	Cannot add event existing, but add its capacity
4	TORManager removes a local city event without customer reservation	Remove event	Starts three servers. Manager login. The event exists.	Choose “remove Event”. Input eventID, eventType	TORA070619 conference	Remove event successfully	As expected	Remove local city event
5	TORManager removes an another city event without customer reservation	Remove event	Starts three servers. Manager login. The event exists.	Choose “remove Event”. Input eventID, eventType	MTLA070619 conference	Fail to remove event	As expected	Cannot remove another city event
6	TORManager removes a local event with customer reservation	Remove event	Starts three servers. Manager login. Customer login. The event exists and customers booked it.	Choose “remove Event”. Input eventID, eventType	TORA070619 conference	Remove event successfully	As expected	Remove local city event and cancel it for all customers in it
7	Manager views the list of event availability	List event availability	Starts three servers. Manager login.	Choose “List event availability”. Input eventType	conference	Display the list	As expected	Display the list of available events in one type

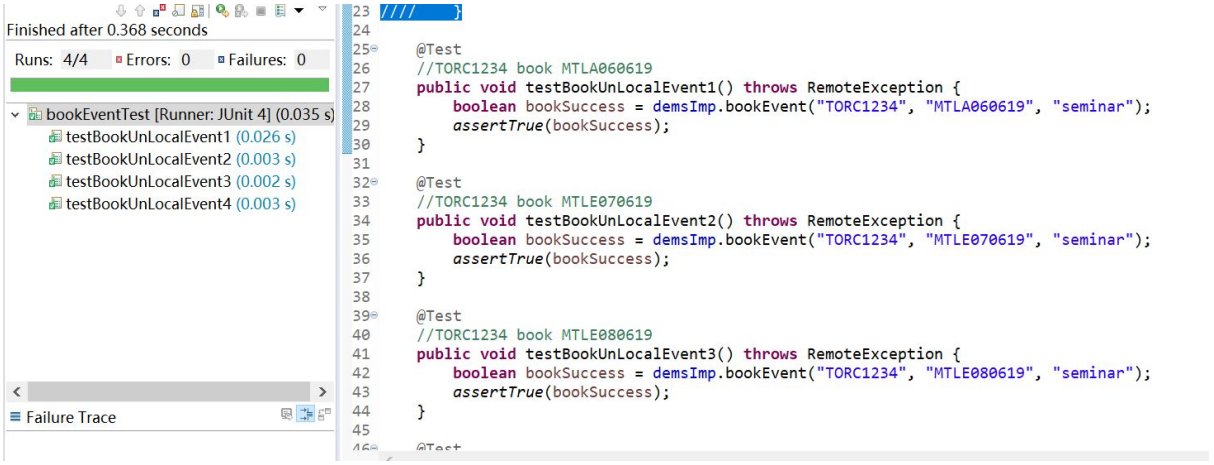
8	TORCustomer books a local city event	Book event	Starts three servers. Customer login. The event exists.	Customer chooses “book Event”. Input eventID, eventType	TORA070619 conference	Book event successfully	As expected	Book lical city event
9	TORManager books event for a local city customer	Book event	Starts three servers. Manager login. The event exists. Customer did not book it.	Manager chooses “book Event”. Input CustomerID, eventID, eventType	TORC0001 TORA070619 conference	Book event successfully	As expected	Manager helps its local customer to book event
10	TORManager books event for another city customer	Book event	Starts three servers. Manager login. The event exists.	Manager chooses “book Event”. Input CustomerID, eventID, eventType	MTLC0001 TORA070619 conference	Fail to book event	As expected	Manager don’t have power to help unlocal customer operate
11	TORCustomer books an event in another city, no more than three times a month	Book event	Starts three servers. Customer login. The event exists.	Customer chooses “book Event”. Input eventID, eventType	MTLA070619 tradeshow	Book event successfully	As expected	Coustomer has limitation about the number of booking unlocal event
12	TORCustomer books an event in another city, more than three times a month	Book event	Starts three servers. Customer login. The event exists. Customer has booked three other cities events in one month.	Customer chooses “book Event”. Input eventID, eventType	OTWA070619 tradeshow	Fail to book event	As expected	Coustomer has limitation about the number of booking unlocal event
13	Customer books an event	Book event	Starts three servers. Customer	Customer chooses “book	TORA070619 conferen	Fail to book event	As expected	When the event is not

	which has 0 capacity		login. The event exists. The number of people who booked the event is equal to its capacity.	Event". Input eventID, eventType	ce			available, customer can't book this event
14	Customer views the booking schedule	Get booking schedule	Starts three servers. Customer login.	Customer chooses " get booking schedule".		Display all events booked	As expected	Display all events booked of one customer
15	TORManager views the booking schedule of local city customer	Get booking schedule	Starts three servers. Manager login.	Manager chooses " get booking schedule". Input CustomerID	TORC0001	Display all events booked for that customer	As expected	Manager views local customer schedule
16	TORManager views the booking schedule of another city customer	Get booking schedule	Starts three servers. Manager login.	Manager chooses " get booking schedule". Input CustomerID	MTLC0002	Fail to get	As expected	Manager cannot view another city customer schedule
17	Customer cancels a booked event	Cancel event	Starts three servers. Customer login. The event exists. Customer booked this event.	Customer chooses " cancel event". Input eventID, eventType	TORA070619 conference	Cancel event successfully	As expected	Cancel booked event
18	Customer cancels an unbooked event	Cancel event	Starts three servers. Customer login. The event exists.	Customer chooses " cancel event". Input eventID, eventType	TORA070619 tradeshow	Fail to cancel event	As expected	Cannot cancel unbooked event
19	TORManager	Cancel	Starts three servers.	Manager chooses "	TORC0001	Cancel event	As expe	Manager cancels

	cancels an event for local city customer	even t	Manager login. Customer login. The event exists. Customer booked this event.	cancel event". Input CustomerID, eventID, eventType	TORA07 0619 conferen ce	succes sfully	cted	events for local customer
20	TORMan ager cancels an event for another city customer	Canc el even t	Starts three servers. Manager login. Customer login. The event exists. Customer booked this event.	Manager chooses " cancel event". Input CustomerID, eventID, eventType	MTLC00 02 TORA07 0619 conferen ce	Fail to cancel event	As expe cted	Manager cannot cancel events for another city customer
21	MTLCust omer swaps a new nonlocal event with an old local event when there are three nonlocal events and one local event in booking list	swap even t	Starts three servers. Managers login. Customer login. The events exist. Customer has booked three nonlocal events in a month and one local event.	Customer chooses "swap event". Input new eventID, new eventType, old eventID, old eventType.	TORA07 0719 conferen ce MTLA07 0719 conferen ce	Fail to swap events	As expe cted	When the customer has booked three nonlocal events in a month, he can not swap a new nonlocal event in that month with an old local booked event.
22	MTLCust omer swaps a new nonlocal event with an old nonlocal event when there are three	swap even t	Starts three servers. Managers login. Customer login. The events exist. Customer has booked three nonlocal events in a month.	Customer chooses "swap event". Input new eventID, new eventType, old eventID, old eventType.	TORA07 0719 conferen ce TORE07 0719 tradesho w	swap events succes sfully	As expe cted	When the customer has booked three nonlocal events in a month, he can swap a new nonlocal event

	nonlocal events in booking list							with an old nonlocal event.
23	Customer swaps a new event with an old event which was not booked	swap event	Starts three servers. Managers login. Customer login. The events exist.	Customer chooses “swap event”. Input new eventID, new eventType, old eventID, old eventType.	MTLA070719 conference MTLE070719 tradeshow	Fail to swap events	As expected	Customer cannot swap the old event which was not booked
24	Customer swaps a new event which has been booked with an old event.	swap event	Starts three servers. Managers login. Customer login. The events exist. Customer has booked an event.	Customer chooses “swap event”. Input new eventID, new eventType, old eventID, old eventType.	TORA070719 conference TORA070719 conference	Fail to swap events	As expected	Customer cannot swap the new event which has been booked




Finished after 0.368 seconds
Runs: 4/4
Errors: 0
Failures: 0
bookEventTest [Runner: JUnit 4] (0.035 s)
testBookUnLocalEvent1 (0.026 s)
testBookUnLocalEvent2 (0.003 s)
testBookUnLocalEvent3 (0.002 s)
testBookUnLocalEvent4 (0.003 s)
Failure Trace
//// }
@Test
//TORC1234 book MTLA060619
public void testBookUnLocalEvent1() throws RemoteException {
boolean bookSuccess = demsImp.bookEvent("TORC1234", "MTLA060619", "seminar");
assertTrue(bookSuccess);
}
@Test
//TORC1234 book MTLE070619
public void testBookUnLocalEvent2() throws RemoteException {
boolean bookSuccess = demsImp.bookEvent("TORC1234", "MTLE070619", "seminar");
assertTrue(bookSuccess);
}
@Test
//TORC1234 book MTLE080619
public void testBookUnLocalEvent3() throws RemoteException {
boolean bookSuccess = demsImp.bookEvent("TORC1234", "MTLE080619", "seminar");
assertTrue(bookSuccess);
}