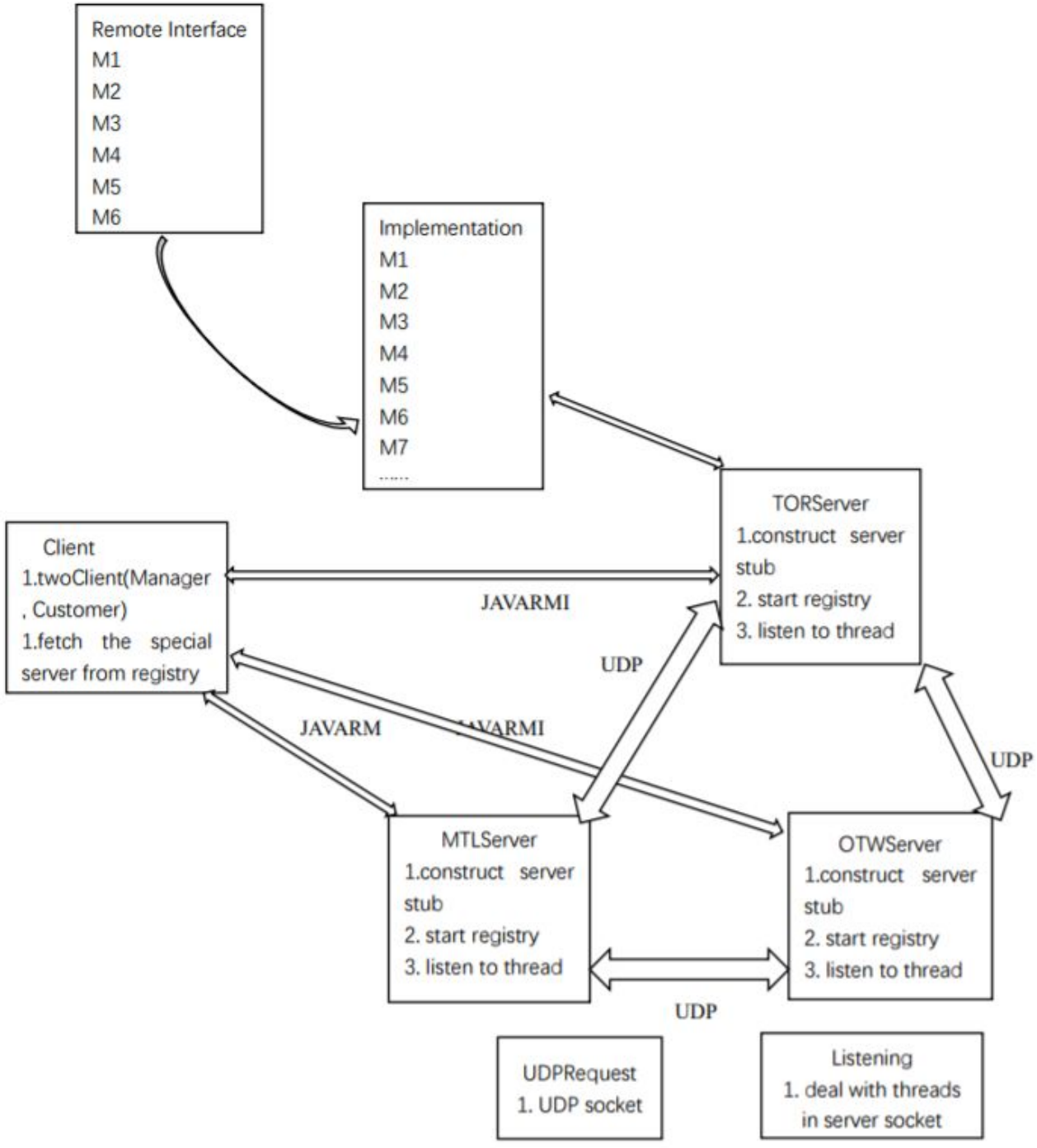
DEMS Design Documentation

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

---

1. Techniques and architecture

Techniques: This assignment uses Java RMI to let client and server communicate and use UDP/IP sockets to achieve inter-server communication.

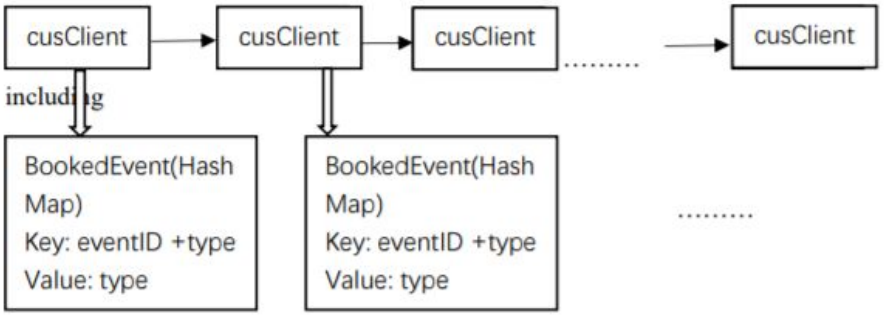


2. Project file structure

(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)Multithreading and UDP: UDPRequest.java, RequestExection.java, Listening.java
(6)Logs: ClientLog Folder, ServerLog Folder

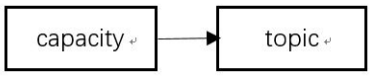
3. Data structure

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

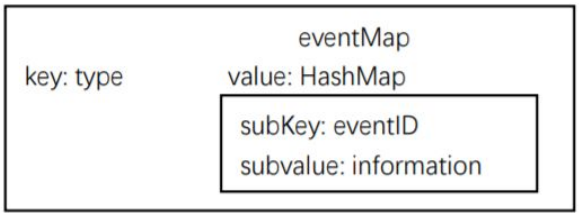


(3) LinkedList: manClients

(4) LinkedList: bookedCus for each event
(5) LinkedList: eventDetail for each event



(5) Two Level HashMap:



4. Function realization

(1) Add event

addEvent(String eventID, String eventType, String manID, LinkedList<String> eventDetail)
addEventLocal(String eventID, String eventType, String manID, LinkedList<String> eventDetail)

The *addEvent* invoke *addEventLocal* to add the event into event Hashmap if this event is not in the map. If the event exists, the capacity of it will plus one. If the type of event does not exist, add the type as a key of the map first.

(2) Remove event

removeEvent(String eventID, String eventType, String manID)

If there are customers in this event, call the *cancelEvent* function to cancel it for customers first. Then the event will be removed from the event Hashmap.

(3) List Event Availability

listEventA(String manID, String eventType)
listEventALocal(String eventType)

The function *listEventA* first call *listEventALocal* to get the result from the local city. Then it uses UDP request to communicate with the other two servers to get the result.

(4) Book event

bookEvent(String cusID, String eventID, String eventType)
insertBook(String cusID, String eventID, String eventType)
bookEventLocal(String cusID, String eventID, String eventType)

The function *bookEvent* determines if the customer is in the event. If the customer does not book it, the function will check the city where the event is located. If this event in another city, check if more than three events are booked within one month. Then it uses UDP to communicate with other servers.

(5) Get booking schedule

getBookingSchedule(String cusID)

The function uses the Hashmap of this customer to display all booked events.

(6) Cancel event

cancelEvent(String cusID, String eventID, String eventType)

cancelEventLocal(String cusID, String eventID, String eventType)
cancelForCustomer(String cusID, String eventID)

If the city of the event is same as the city of the customer, then *cancelEvent* function calls *cancelEventLocal* to cancel it. Otherwise, the function will use UDP to communicate with other servers.

5. Important/Difficult part

The important part of this assignment is using Java RMI to implement communication between clients and servers and using UDP/IP sockets to achieve inter-server communication.

About Java RMI, it makes the client be able to use the remote methods residing on the server.

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.

About multithreading, in order to achieve synchronization, we put the process of dealing with every request that server socket received into an independent thread. So we can have more than one client to operate the DEMS system at the same time. This part is achieved in Listening.

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 new another city event	Add event	Starts three servers. Manager login.	Choose "add Event". Input eventID, eventType, capacity, topic	MTLA070619 conference 10 IT	Fail to add event.	As expected	Cannot add another city event
3	TORManager adds an existing	Add event	Starts three servers. Manager login. The event	Choose "add Event". Input eventID,	TORA070619 conference	Fail to add event. Its	As expected	Cannot add event existing,

	event		exists.	eventType, capacity, topic	10 IT	capacity plus one.		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	Book even	Starts three servers.	Manager chooses	TORC0001	Book event	As expe	Manager helps its

	books event for a local city customer	t	Manager login. The event exists. Customer did not book it.	“book Event”. Input CustomerID, eventID, eventType	TORA070619 conference	succesfully	cted	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 succesfully	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 which has 0 capacity	Book event	Starts three servers. Customer login. The event exists. The number of people who booked the event is equal to its capacity.	Customer chooses “book Event”. Input eventID, eventType	TORA070619 conference	Fail to book event	As expected	When the event is not available, customer can’t book this event
14	Customer	Get	Starts three	Customer		Display	As	Display

	views the booking schedule	booking schedule	servers. Customer login.	chooses “ get booking schedule”.		all events booked	expected	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 cancels an event for local city customer	Cancel event	Starts three servers. Manager login. Customer login. The event exists. Customer booked this event.	Manager chooses “ cancel event”. Input CustomerID, eventID, eventType	TORC0001 TORA070619 conference	Cancel event successfully	As expected	Manager cancels events for local customer
20	TORMan	Canc	Starts three	Manager	MTLC00	Fail to	As	Manager

	ager cancels an event for another city customer	el even t	servers. Manager login. Customer login. The event exists. Customer booked this event.	chooses “ cancel event”. Input CustomerID, eventID, eventType	02 TORA07 0619 conferen ce	cancel event	expe cted	cannot cancel events for another city customer