

UseCase Specification

NetworkManagementTool

Group D2

Catelog

Use Case: CommunicationBetweenTwoUsers.....	1
Use Case: EmergencyResponder	3
Use Case: AccessSocialNetwork.....	5

Network Management Tool

Use Case: CommunicationBetweenTwoUsers

1 Brief Description

This Use Case will describe the flow of events NMT will undergo when two users communicate with each other..

2 Actor Brief Descriptions

UserA, UserB

3 Preconditions

NMTa and NMTb are installed on UserA and UserB's devices.

4 Basic Flow of Events

- 1 UserA is sending message to user B through C.
(the social network software,like Facebook. The CommunicationNetworkNode in our class diagram is actually a representation of this software and a component in our NMT software)
- 2 CommunicationNetworkNodeA detects the message and temporarily stores the message in MssgBuket.(Here I am using word 'detect', because whether or not our nmt will intercept this message and resend it is an implementation issue.)
- 3 UserB suddenly becomes unreachable for some reason.(He may shut down his communication node or something happened to his router)
- 4 No acknowledgement is received, CommunicationNetworkNodeA keeps trying sending the stored message.
- 5 After N times, CommunicationNetworkNodeA gives up and notifies UserA a failure.
- 6 CommunicationNetworkNodeA detects userA is about to shut down his computer.
CommunicationNetworkNodeA selects several CommunicationNetworkNodes that's mostly likely to be online at the same time as UserB and then sends the stored message to those CommunicationNetworkNodes.(They are just the same node as CommunicationNetworkNodeA, but just are assigned to be act as an server for CommunicationNetworkNodeA temporarily), here we call those nodes are CNs.(stands for Central nodes);

- 7 All CNs receive the message from CommunicationNetworkNodeA and store the message in their own MssgBuket.
- 8 All CNs keep monitoring the status of CommunicationNetworkNodeB, but UserB never comes back again.
- 9 UserA comes up and so is the CommunicationNetworkNodeA is online.
- 10 CommunicationNetworkNodeA sends request to those CNs for the status of those message it sent out earlier.
- 11 CNs send back the status of those message regarding to it's been sent or not. In this case, it's not. CNs now know CommunicationNetworkNodeA is up, they are not responsible for those message, so they delete those messages from their MssgBuket.
- 12 CommunicationNetworkNodeA keep monitoring the status of UserB.
- 13 UserB comes up and so is the CommunicationNetworkNodeB is online.
- 14 CommunicationNetworkNodeA detects that CommunicationNetworkNodeB has come up. Start send out its stored message for UserB.
- 15 CommunicationNetworkNodeB receives the message and sends an ack back to CommunicationNetworkNodeA.
- 16 Upon receipt, CommunicationNetworkNodeA notifies userA.

5 Alternative Flows

In step 8, UserB comes back before UserA comes back.

1. CNs detect that UserB has come back. So it sends out the message to UserB
2. UserB sends back an ack to all the CNs. (the list of CNs can be made when those CNs were first chosen by CommunicationNetworkNodeA, this list are sent to all CNs along with the message. So every CN knows each other.)
3. All CNs delete the message from their MssgBuket. Update the status of the message to be 'sent'
4. UserA comes up and so is the CommunicationNetworkNodeA is online.
5. CommunicationNetworkNodeA queries one of the CNs about the status of the message.
6. One CN sends back the status to CommunicationNetworkNodeA and other CNs, and delete all information about this message.
7. CommunicationNetworkNodeA notifies UserA the message it's sent and delete the message from its MssgBuket.

6 Post-conditions

UserA's message is sent to UserB.

Use Case: EmergencyResponder

1 Brief Description

This Use Case will describe the flow of events NMT will undergo when ER gets report from user.

2 Actor Brief Descriptions

1. User
2. ER- Emergency responder
3. RD-Remote database
4. CommunicationNetworkNode

3 Preconditions

ER reacts when gets a report from user.

4 Basic Flow of Events

1. The use case begins after emergency responder gets a report from a user.
2. ER stores the report into its local database through information node.
3. ER gets the information about the place where the emergency happens from the report.
4. ER attempts to get information about people related in this emergency.
5. ER sends a request to CommunicationNetworkNode to access RD via CommObj.
6. CommunicationNetworkNode sends request to RD and wait for information.
7. CommunicationNetworkNode gets people information from RD and sends it to ER via CommObj.
8. ER attempts to sends people notifications about the emergency via CommunicationNetworkNode.
9. CommunicationNetworkNode sends the message to CommunicationNetworkNode of user.
10. CommunicationNetworkNode of user receives the message and sends an acknowledgement back to CommunicationNetworkNode of ER.
11. CommunicationNetworkNode of user sends the notification to user.
12. User gets notification about emergency.

5 Alternative Flows

If in step 5 of the basic flow the CommunicationNetworkNode of ER does not receive information from RD, then

1. CommunicationNetworkNode waits N seconds for the information.
2. After N seconds, CommunicationNetworkNode of ER resends request to RD N times and waits for another N seconds.
3. After N seconds, if CommunicationNetworkNode of ER still doesn't get anything from RD, it stores request into its local database via information node.
4. If waiting time is greater than N minute, CommunicationNetworkNode of ER sends a notification to ER that information cannot be reached.
5. CommunicationNetworkNode of ER sends request to CommunicationNetwork, CommunicationNetwork will monitor RD.
6. CommunicationNetwork stores request to CommObj based on priority level.
7. When RD gets connected, CommunicationNetwork sends request to RD and after gets result, sends result to CommunicationNetworkNode node of ER and resume step 7 of basic flow.

If in step 11 of the basic flow the CommunicationNetworkNode of ER does not receive acknowledgement from CommunicationNetworkNode of user, then

1. CommunicationNetworkNode waits N seconds for the acknowledgement.
2. After N seconds, CommunicationNetworkNode of ER resends packets to CommunicationNetworkNode of user N times and waits for another N seconds.
3. After N seconds, CommunicationNetworkNode of ER still doesn't get acknowledgement, it stores user's packets into database via CommObj.
4. If waiting time is greater than N minute, CommunicationNetworkNode of ER sends a notification to ER that notification cannot be sent out.
5. CommunicationNetworkNode of ER sends packets to CommunicationNetwork, CommunicationNetwork will monitor user.
6. CommunicationNetwork stores packets to its database based on priority level.
7. When user gets connected, CommunicationNetwork sends notification to CommunicationNetworkNode of user and after gets acknowledgement from CommunicationNetworkNode of user, it sends notification to CommunicationNetworkNode of ER that user has received notification and resume step 12 of basic flow.

6 Post-conditions

7 Special Requirements

Use Case: AccessSocialNetwork

1 Brief Description

This Use Case will describe the flow of events NMT will undergo when User A attempts to login to social network.

2 Actor Brief Descriptions

1. User A - a internet user
2. SNN— social network node
3. CMN—communication node
4. INN – information network node

3 Preconditions

User A tries to login to social network(like Facebook).

4 Basic Flow of Events

1. The use case begins when User A attempts to access SN.
2. User A turns on his/her laptop as NMT is activated.
3. User A accesses social network through SNN.
4. User A login to social network by sending requests out through CMN.
5. CMN sends out the ack for logging in via CommObj.
6. CommObj stores the request information to database.
7. CMN gets log in respond from SN and send it to SNN.
8. CMN sends ack to CommObj and update user A's status via CommObj.
9. UserA logs in successfully.

5 Alternative Flows

If in step 7 of the basic flow the NMT of user A does not receive respond from social network, then .

1. CommObj checks the network status.
2. If network is connected, CommObj retrieves userA's login request from local database through INN and let CMN sends out request again. If network is disconnected, NMT keeps monitor network status.

3. When the network is recovered, CommObj retrieve userA's login request from local database through INN and let CMN sends out request again. Resume step 7 of basic flow.

6 Post-conditions

7 Special Requirements