**Resolved outage:**

resolvedOutage (outageID in number)

*Algorithm:* Mark the issue as resolved for the specific outage id.

The input for the feature is outage id for the issue. The record corresponding to the issue will be marked as resolved in ‘outage’ table and also record the ‘end\_time’ for the issue.

A message will be inserted into message table stating that the issue has been resolved.

*Input:*  1) outageID

*Output:* Message will be displayed saying that the issue has been resolved.

/\*Allow the system to report that an outrage has been resolved. Update the outrage status to resolved, record the resolved time. Insert a message to the message table for each account involved stating that the outrage has been resolved.  \*/

/\* This procedure takes in outage\_id as an input \*/

show errors;

Create or replace PROCEDURE resolved\_outage(outage\_ID in number) IS

/\* It loops through to find the affected houses’ account id(aid) using the outage\_id to inform them in the message table that their problem has been resolved \*/

cursor c1 is select aid from houses\_affected,account where houses\_affected.oid=outage\_ID  and houses\_affected.hid= account.hid;

account\_id account.aid%type;

/\* account\_id is a local variable used to store all the accounts that are affected by a particular outage \*/

BEGIN

open c1;

loop

/\*Once our query finds a record,it will fetch into the local variable \*/

fetch c1 into account\_id;

exit when c1%NOTFOUND;

/\*The outage table will be updated and a message will be inserted into the message table for every account that has been affected by the outage \*/

Update Outage SET status='Resolved', end\_time = sysdate where oid=outage\_ID;

Insert into Message Values (message\_id\_seq.nextval ,account\_id,'The ticket has been resolved',sysdate);

end loop;

close c1;

end;