

Nexmock (mocked messages) - how to add into cfl distro?

- <https://github.com/birdiecare/nexmock> - clone this repo next to the folder with cloned cfl distro.
- Add dockerfile in this cloned repo of nexmock:

```
# Use an official Node.js runtime as a parent image
FROM node:14-alpine

# Set the working directory in the container
WORKDIR /usr/src/app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install app dependencies
RUN npm install

# Copy the application code into the container
COPY . .

# Expose the port the app runs on
EXPOSE 5151

# Define the command to run your application
CMD ["sh", "-c", "PORT=5151 npm run start:dev"]
```

- replace scr/pricing.service.ts with:

```
import { Injectable } from '@nestjs/common';
import { prices } from '../constants/prices';

@Injectable()
export class PricingService {

  getPrice(phoneNumber?: string): number {
    if (!phoneNumber) {
      return 0;
    }

    phoneNumber = String(phoneNumber);
    if (!phoneNumber.startsWith('+')) {
      phoneNumber = '+' + phoneNumber;
    }

    const prefixes = Object.keys(prices);
    const prefix = prefixes.find(p => phoneNumber.startsWith(p));
    return prices[prefix];
  }
}
```

- In openmrs-distro-vxnaid within CFL folder - add 'docker-compose.nexmock.yml' file:

```
version: '2'

services:
  nexmock:
    build:
      context: ../../nexmock-app
      dockerfile: Dockerfile
    container_name: nexmock-container
    ports:
      - "5151:5151"
```

- update runInDevMode.sh:

```
#!/bin/bash
docker-compose -f docker-compose.run.yml -f docker-compose.db.yml stop
docker-compose -f docker-compose.build.yml -f docker-compose.run.yml -f
docker-compose.debug.yml -f docker-compose.db.yml -f
docker-compose.nexmock.yml up --build -d
sleep 10
curl -L http://localhost:8080/openmrs/ > /dev/null && notify-send -t 10000
-i face-surprise "Demo set up" "" &
docker-compose -f docker-compose.run.yml -f docker-compose.db.yml logs -f
--tail 1000
```

- You can run runInDevMode.sh
- To improve mocked messages you can set up new global property in <http://localhost:8080/openmrs/adminui/systemadmin/globalproperties/manageGlobalProperties.page#/list> -> please add 'messages.notificationTemplate.visit-reminder' and paste this code as the value of this property:

```

#set ($uuidClazz = $openmrsContext.loadClass('java.util.UUID'))
#set ($openmrs_id = $uuidClazz.randomUUID().toString())
#set ($integerClazz = $openmrsContext.loadClass('java.lang.Integer'))
#set ($stringClazz = $openmrsContext.loadClass('java.lang.String'))
#set ($simpleDateFormat =
$openmrsContext.loadClass('java.text.SimpleDateFormat').getDeclaredConstru
ctor($stringClazz).newInstance('yyyy-MM-dd'))
#set ($visitTypeIdInteger = $integerClazz.parseInt($visitTypeId))
#set ($visitPurpose =
$openmrsContext.getVisitService().getVisitType($visitTypeIdInteger).getNam
e())
#set ($patientName =
$patient.getPersonName().toString().replaceAll("[^A-Za-z0-9 ']", " "))
#set ($caregiverName =
$actor.getPersonName().toString().replaceAll("[^A-Za-z0-9 ']", " "))
#set($textToRead1 = "")
#if($patient.getId().equals($actor.getId()))
  #set($textToRead1 = "Hello $patientName, You have a")
#else
  #set($textToRead1 = "Hello $caregiverName, Your patient has a")
#end
#if($timeStarted)
  #set($visitTime = " at $timeStarted")
#else
  #set($visitTime = "")
#end
#set($textToRead2 = "visit scheduled for
$simpleDateFormat.format($simpleDateFormat.parse($dateStarted)) $visitTime
for the purpose of $visitPurpose.")
{ message:"$textToRead1 $textToRead2", openmrs_id:"$openmrs_id" }

```

- To set up message config based on nexmock you have to:
 - Log into CfL as admin
 - Go to the 'System Administration'
 - Go into 'SMS'
 - Go into 'SMS Configuration'
 - Upload config based on the bellowed JSON:

```
[
  {
    "name": "Nexmock2",
    "configurables": [
      "from",
      "username",
      "password",
      "basic_token"
    ],
    "outgoing": {
      "maxSmsSize": "160",
      "maxRecipient": "1",
      "millisecondsBetweenMessages": "1",
      "exponentialBackOffRetries": "true",
      "hasAuthentication": "false",
      "request": {
        "type": "POST",
        "urlPath": "http://nexmock:5151/sms/json",
        "recipientsSeparator": ", ",
        "bodyParameters": {
          "from": "[from]",
          "to": "[recipients]",
          "text": "[message]"
        },
        "headerParameters": {
          "Authorization": "[basic_token]"
        }
      },
      "response": {
        "multiLineRecipientResponse": "true",
        "singleRecipientResponse": "true",
        "extractSingleSuccessMessageId": "^ID: ([0-9a-zA-Z]+)$",
        "extractSingleFailureMessage": "^ERR: (.*)$",
        "extractSuccessMessageIdAndRecipient": "^ID: ([0-9a-zA-Z]+) To: ([0-9]+)$",
        "extractFailureMessageAndRecipient": "^ERR: (.*) To: ([0-9]+)$"
      }
    },
    "status": {
      "messageIdKey": "apiMsgId",
      "statusKey": "status",
      "statusSuccess": "003|004|008",
      "statusFailure": "005|006|007|009|010|012|014"
    },
    "incoming": {
      "messageKey": "text",
      "senderKey": "from",
      "recipientKey": "to",
      "msgIdKey": "moMsgId",
      "timestampKey": "timestamp"
    }
  }
]
```

- Add config based on the bellowed screenshot:

SMS Configurations

ADD CONFIGURATION Import additional configuration templates

▼ nexmock ★

Name:
nexmock

Template:
Nexmock

Maximum Retries:
1

Split Message Header:
Msg \$m of \$t

Split Message Footer:
...

Exclude footer from last split message
☒

Automatic Response Script

username:
username

password:
changeme

from:
14157386102

- Click the star right to the name of configuration. This one makes the configuration as the default one.

Connectcare
Empowering Patients & Health Systems

Home >> System Administration >> SMS >> Settings

SMS Configurations

ADD CONFIGURATION Import additional configuration templates

> nexmock ★

SAVE

- You can test this configuration in 'Send SMS'

- You can check if a message is sent and received on localhost:5151/list. If a message is on the list - it means that everything is good in terms of mocked messaging configuration. You can use it in performance testing.

Timestamp	From	To	Preview	Message
17 minutes ago	+987654321	4828619189571	Hello Test 579 Test 579, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4820784379443	Hello Test 170 Test 170, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4866896290515	Hello Test 372 Test 372, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4826892311283	Hello Test 165 Test 165, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4885457119259	Hello Test 377 Test 377, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4857481344635	Hello Test 574 Test 574, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4890383245215	Hello Test 786 Test 786, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4827119998165	Hello Test 776 Test 776, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4865388154578	Hello Test 988 Test 988, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4831266750063	Hello Test 175 Test 175, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4825744545578	Hello Test 155 Test 155, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4852139430951	Hello Test 983 Test 983, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4819095651224	Hello Test 1003 Test 1003, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4852671298428	Hello Test 796 Test 796, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details
17 minutes ago	+987654321	4866816954292	Hello Test 968 Test 968, You have a visit scheduled for 2023-11-21 at Evening for the purpose of Follow-up.	See details

- You can check messages when port 5151 on server is restricted by:
 - Log into server through ssh
 - Run `docker exec -it <container_id> /bin/bash` with **web** service to go into container
 - Within container run:

Unset

```
curl -X GET \  
  http://nexmock:5151/list \  
  -H 'Authorization: Basic \  
  <base64(username:password)>'
```

- Default credential for nexmock: user: username, password: changeme
- You should have the list of messages and you can verify if messages are sent and received by this mock tool.

IvrMock (mocked calls) - how to add into cfl distro?

Prerequisites

In order to use mocked IVR calls, you have to:

- create openMRS/CfL folder
- within newly created folder, clone [CfL distro](#)
- verify CfL module - `callflows-1.1.12.omod` (to verify, check version within `openmrs-distro-cfl/cfl/web/cfl-modules/`)

Installation on local environment or test server

To make a copy of this project on your local machine, please follow the next steps:

- clone [CfL distro](#) (if you don't have yet cloned)
- `https://github.com/SolDevelo/vonage-calls-sample-mock-server` - clone this repo 'ivrMock' next to the folder with cloned cfl distro.
- In openmrs-distro-vxnaid within CFL folder - add 'docker-compose.nexmomockivr.yml' file:

```
version: '2'

services:
  redis:
    image: "redis:latest"
    ports:
      - "6379:6379"

  ivrmock:
    build:
      context: ../../vonage-calls-sample-mock-server
      dockerfile: Dockerfile # Specify the Dockerfile name if it's not
named "Dockerfile"
    ports:
      - "7000:7000"
```

- update runInDevMode.sh:

```
#!/bin/bash
docker-compose -f docker-compose.run.yml -f docker-compose.db.yml stop
docker-compose -f docker-compose.build.yml -f docker-compose.run.yml -f
docker-compose.debug.yml -f docker-compose.db.yml -f
```

```
docker-compose.nexmock.yml -f docker-compose.nexmomockivr.yml up --build
-d
sleep 10
curl -L http://localhost:8080/openmrs/ > /dev/null && notify-send -t 10000
-i face-surprise "Demo set up" "" &
docker-compose -f docker-compose.run.yml -f docker-compose.db.yml logs -f
--tail 1000
```

- You can run runInDevMode.sh. Alternatively if you want to attach this new service into already running CfL distro you can run this by executing in this folder:

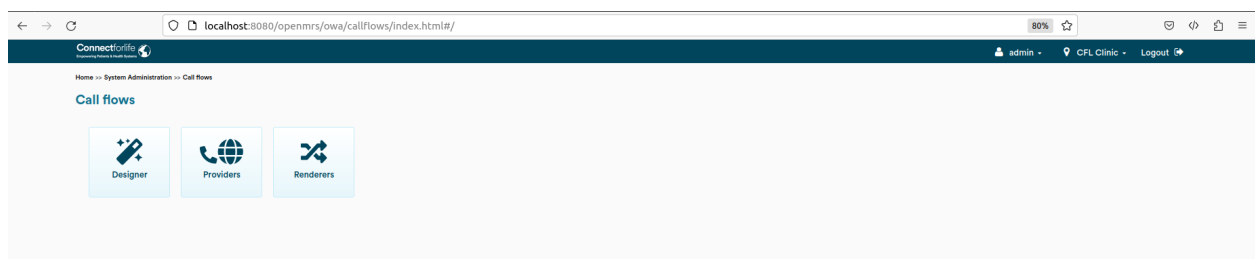
Unset

- `docker-compose -f docker-compose.nexmomockivr.yml up --build -d`

- now you should have running service with ivrMock app. Verify if app is running by executing `docker logs <container-name-or-id>` or `docker ps`.

Configuration on local app or app working in test environment

- Log into CfL as admin `http://<hostname>/openmrs/login.htm`
- Go into `http://localhost:8080/openmrs/owa/callflows/index.html#/providers` calls providers



- Add new provider
- Fill the form in the way as it is below:

Unset

Name: nexmo

a) Outgoing call URI template (optional):

http://ivrmock:7000/v1/calls/

b) Is authentication required?: Unchecked

c) Outgoing call HTTP method: Check POST

d) POST header parameters: Content-Type: application/json

e) POST parameters: {"to": [{"type": "phone", "number":

[phone]}], "from": {"type": "phone", "number":

489999999876}, "answer_url":

["http://web:8080/openmrs/ws/callflows/in/nexmo/flows/MainFlow.nexmo?numbertodial=[phone]&jumpTo=[internal.jumpTo]&callId=[internal.callId]&phone=[phone]"],

"event_url": ["http://web:8080/openmrs/ws/callflows/status/[internal.callId]?externalIdParamName=conversation_uuid&externalType=conversation_uuid"],

"event_method": "GET", "length_timer": 600}

f) Outbound Call Queue Configuration -> Call Limit: 0, Retry Sec: 0, Retry Attempts 0, Call after all Retry Attempts?: checked

g) Map of services that can be injected in IVR templates:

patientService:patientService,cflPersonService:cflPersonService,messagesService:messages.messagingService,

personService:personService,personDAO:personDAO,conceptDAO:conceptDAO,locationService:locationService,

messagingGroupService:messages.messagingGroupService,openmrsContext:context,patientTemplateService:messages.patientTemplateService,

healthTipService:messages.healthTipService,callService:callflows.callService,

adherenceFeedbackService:messages.adherenceFeedbackService,conceptService:conceptService

- Save provider configuration
- Create renderer in <http://localhost:8080/openmrs/owa/callflows/index.html#/renderers>

- Load the callflows by uploader in `http://localhost:8080/openmrs/module/metadatasharing/import/list.form`
- Click `import package`

[Home](#) | [Find/Create Patient](#) | [Dictionary](#) | [Cohort Builder](#) | [Reporting](#) | [Appointments](#) | [Administration](#)

Currently logged in as Super User | [Logout](#) | [My Profile](#) | [Join](#)

[Admin](#) | [Export Metadata](#) | [Import Metadata](#) | [Manage Tests](#) | [Configure](#)

Metadata Sharing

Import Metadata

Packages subscriptions:

Subscription	Status	
Emergency medicine	Unsubscribed	<input type="button" value="Subscribe"/>
	Unsubscribed	<input type="button" value="Subscribe"/>
	Unsubscribed	<input type="button" value="Subscribe"/>

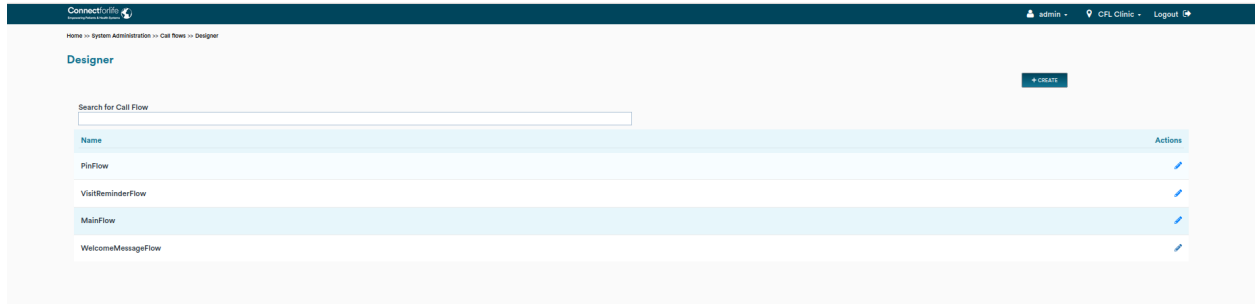
total registered servers: 3

- Browse the proper package in ZIP format

- click next
- in 3rd step check **Choose import mode: From Master**

- in 4th step check **4. Adjust import mode: Skip Assessing (Advanced)**. Leave the rest options as are as default

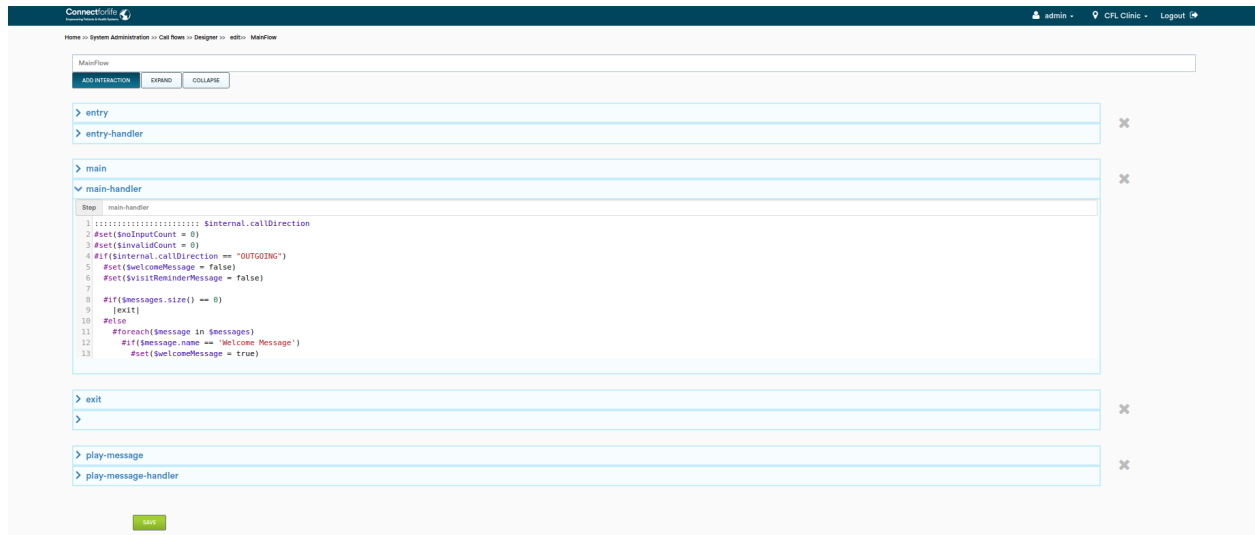
- Now you should have imported callflows. Verify this by going to **<http://localhost:8080/openmrs/owa/callflows/index.html#/designer>**



- In <http://localhost:8080/openmrs/owa/callflows/index.html#/designer/edit/MainFlow> please replace code by this one and save changes:

```
Unset
:::::::::::::::::::: $internal.callDirection
#set($noInputCount = 0)
#set($invalidCount = 0)
#if($internal.callDirection == "OUTGOING")
    #set($welcomeMessage = false)
    #set($visitReminderMessage = false)

    #if($messages.size() == 0)
        |exit|
    #else
        #foreach($message in $messages)
            #if($message.name == 'Welcome Message')
                #set($welcomeMessage = true)
                #set($welcomeMessageId = $message.messageId)
            #elseif($message.name == 'Visit reminder')
                #set($visitReminderMessage = true)
                #set($visitReminderMessageId = $message.messageId)
            #end
        #end
        |play-message|
    #end
#else
    |exit|
#end
```



- In the same page
<http://localhost:8080/openmrs/owa/callflows/index.html#/designer/edit/MainFlow>
 with callflows design in entry-handler replace code with:

Unset

```

::::::::::::::::::::::::::::::::::
::::: MainFlow params logger:::::
phone:::::::::::::::::: $phone
session.callid::: $params.get("session.callerid")
::::::::::::::::::::::::::::::::::
params:::::::::::::::::: $params
messages:::::::::::::::::: $messages
::::::::::::::::::::::::::::::::::
personId:::::::::::::::::: $personId
actorType:::::::::::::::::: $actorType
actorId:::::::::::::::::: $actorId
refKey:::::::::::::::::: $refKey
internal:::::::::::::::::: $internal
::::::::::::::::::::::::::::::::::

#set($callId = $internal.callId)
#set($pinAttempts = 0)
#set($maxPinAttempt = 3)

#if($refKey)
    #set($sourceId = $refKey)
    #set($sourceType = "SCHEDULED_SERVICE_GROUP")
    #set($serviceGroup =
$messagingGroupService.getById($Integer.parseInt($refKey)))
    #set($patientId = $serviceGroup.getPatient().personId)
patientId:::::::::::::::::: $patientId
service group::: $sourceId

```

```

#else
    #set($sourceId = $callId)
    #set($sourceType = "OTHER")
    #set($patientId = $actorId)
#end

#if(!$actorId && $personId)
    #set ($actorId = $personId)
#elseif($actorId)
    #set ($actorId = $Integer.parseInt($actorId))
#end

actorId:::::::::: $actorId

#set ($person = $personDAO.getPerson($actorId))

#if($person)
    #set($personStatus = "DEACTIVATED")
    #set($maxPinAttempt = 3)
    #set($pinAttempts = 0)
    ::::::::::::::::::::
    person attributes::::::::::
        #foreach($attribute in $person.getActiveAttributes())
            :::::::::::::::::::: $attribute.attributeType.name
            :::::::::::::::::::: $attribute.value
            :::::::::::::::::::: $attribute.voided
            #if($attribute.attributeType.personAttributeTypeId == 11 &&
            !$attribute.voided)
                #set($personStatus = $attribute.value)
            #end
        #end
    ::::::::::::::::::::
    |PinFlow.|
#else
    |exit|
#end

```


- Go into **`Manage Global Properties`** in **`System Administration`**
- Add value for **message.callConfig**: nexmo

ConnectLife
System Administration > Manage Global Properties

Edit Global Property

Name: messages.callConfig

Description: The string value representing the default value of call configuration used to make calls.

Value: nexmo

Audit Info
UUID: 9f26598f-9283-4eb1-b889-f872b598ebbb

CANCEL **SAVE**

- Update value for **messages.statusesEndingCallflow** GP to replace with **‘ANSWERED,UNANSWERED,MACHINE,BUSY,CANCELLED,FAILED,REJECTED,NO_ANSWER,TIMEOUT,completed,UNKNOWN,DISCONNECTED’**
- Update value for **messages.defaultFlow** GP to add **‘MainFlow’**
- [optional: for debug purposes] if you want to have more logs with **DEBUG** status for calls: please update **log.level** GP with **‘org.openmrs.api.info,org.openmrs.module.callflows:debug’**. In that case you have to either execute **‘runInDevMode.sh’** again or restart the web container.
- Now you should have configured calls based on the IVR mocked solution. You can test this by creating visits for patients.

Testing and verifying

You should create patients with visits and call messages in order to verify if it works. Please check docker logs for IVRMock container to see output.

Meanwhile on the database level in **‘cfl_calls_history’** table the records for particular callId should look like below. It means that the whole call flow is completed. This output should give a **‘green mark’** on the patient's visit calendar.

	startFlow	startNode	startTime	endFlow	endNode	endTime	providerTime	steps	config	actio
1	3	entry	[NULL]	3	exit	23-12-08 12:51:32	[NULL]	12	nexmo	7299
2	3	entry	[NULL]	3	exit	23-12-08 12:51:30	[NULL]	11	nexmo	7299
3	3	entry	[NULL]	3	play-message	23-12-08 12:51:28	[NULL]	10	nexmo	7299
4	3	entry	[NULL]	2	end	23-12-08 12:51:26	[NULL]	9	nexmo	7299
5	3	entry	[NULL]	2	repeat-message	23-12-08 12:51:23	[NULL]	8	nexmo	7299
6	3	entry	[NULL]	2	play-reminder	23-12-08 12:51:21	[NULL]	7	nexmo	7299
7	3	entry	[NULL]	2	entry	23-12-08 12:51:19	[NULL]	6	nexmo	7299
8	3	entry	[NULL]	3	play-message	23-12-08 12:51:17	[NULL]	5	nexmo	7299
9	3	entry	[NULL]	3	main	23-12-08 12:51:15	[NULL]	4	nexmo	7299
10	3	entry	[NULL]	1	validate-pin	23-12-08 12:51:13	[NULL]	3	nexmo	7299
11	3	entry	[NULL]	3	entry	23-12-08 12:51:11	[NULL]	2	nexmo	7299
12	3	entry	[NULL]	3		23-12-08 12:51:09	[NULL]	2	nexmo	7299
13	3	entry	[NULL]	3	entry	[NULL]	[NULL]	0	nexmo	7299
14	3	entry	[NULL]	3	exit	23-12-08 12:51:07	[NULL]	1	nexmo	7299
15	3	entry	[NULL]	3	entry	[NULL]	[NULL]	0	nexmo	7299
16	3	entry	[NULL]	3	entry	[NULL]	[NULL]	0	nexmo	7299

Test-1237 Test-1237 (30/M)

Patient ID: 280053
Patient Location: 4844416429058
Phone number: Activated
Status:

UPDATE THE STATUS

Calendar Overview

Patient

< December 2023 > month week day

MANAGE MESSAGES

SUN	MON	TUE	WED	THU	FRI	SAT
26	27	28	29	30	1	2
3	4	5	6	7	8	9
	19:35 Visit reminder					
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

- Display:
- ☒ Welcome Message
 - ☒ Ad hoc Message
 - ☒ Visit reminder