

Panchkula Welare Trust - Intercace Document - 3.0

Purpose of this Document

The purpose of this document is to describe the mechanism how medical reports will be made available to patients on their mobiles.

Figure 1(a) shows how the reports need to flow from the laboratory software to the Intermediary to Cloud Server which will then make it available to the mobile devices.

Figure 1(b) explains how this will be accomplished. The reports will flow from the laboratory software to a DLL (called sendreport.dll) which will then transmit to intermediary software (Interface to Cloud Server) which will then transmit it to Cloud server from where reports are available to the mobile devices.

For development and modification purposes of the Laboratory software, we need a test bench so that independent testing can be done on the modifications of the laboratory software without dependence on the many components that are there in Figure 1(b). This is accomplished using testbench.exe as illustrated in Figure 1(c). The sendreport.dll will send the reports in the same manner to testbench.exe. With that the Laboratory software changes can be tested.

Figure 1(a)

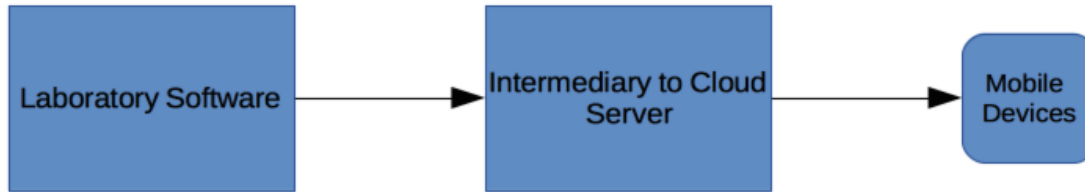


Figure 1(b)

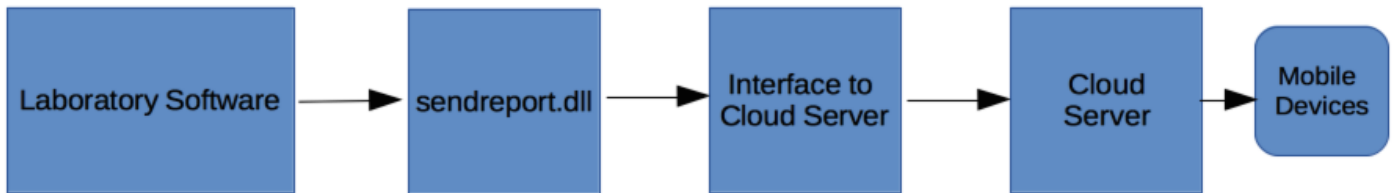
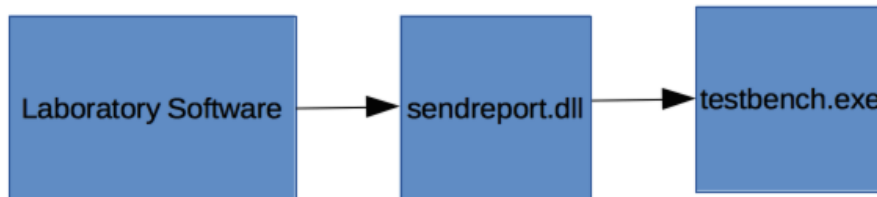


Figure 1(c)



Requirements for the Software at the Lab

- Provide a way to specify in the laboratory software (a) where the Interface to Cloud server (or testbench.exe) is installed. Either the IP address of the testbench server or the name of the server can be specified here. (b) the port on which the Interface to Cloud Server (or testbench.exe) is running. This complete information is provided by the URL that would be sent as the first parameter to sendreport.dll. In case this URL is unspecified, sendreport.dll should not be called.
- At the time of patient registration, the software will need to provide a check-box to indicate if report is to be sent to patient's mobile or not. By default, this will be on (i.e. report will be sent to the mobile). But the operator can choose to uncheck this check box. In that case the reports will not be sent to patient mobile.
- When the report is ready to be made available to patient the software will call the function sendreport() with the pdf of the report file which will then be transmitted to the patient by the interface to the cloud server.

How to run testbench.exe and test against that

- On the development machine (or a separate machine), let's call this machine X, run `testbench.exe` by running `testbench.exe` on the Windows command line.
- Install `sendreport.dll` by copying to windows\system32 of the development machine (Y).
- On the development machine open Visual Basic and then call `sendreport.dll` and use its functions to transmit reports to `testbench.exe`.
- As each report is given to `sendreport.dll` one can also see the corresponding data that is transmitted to it for debugging purposes.

Description of Components

sendreport.dll

This DLL exports two functions: `sendreport()` and `getresult()`. `sendreport()` is to be called when report is to be made available for publication to patient. In case some error happens, `getresult()` can be called to retrieve the description of the error.

Note that as per the requirements of windows, you have to put `sendreport.dll` so that it is discoverable by the VB program. e.g. one option is that you can put `sendreport.dll` in windows\system32 directory.

VBProject

This is a sample project to show how `sendreport.dll` can be sent and errors retrieved in case they happen. As you can see in the sample code, one first needs to declare each function to be used from `sendreport.dll` and then use it as required.

testbench.exe

This `testbench.exe` is to be run so that testing can be done using `sendreport.dll`.

You can run `testbench.exe` on the same computer or a different one (note you have to provide the URL to this computer accordingly). By default `testbench.exe` makes its services available on port 8000 but one can choose a different port by running

```
testbench.exe --port 8800
```

Also there is an option available so that deliberate errors are sent back by `testbench.exe` for alternative invocations. This is there so that the scenario that if an error occurs, the transmission of that particular medical report should be retried again.

```
testbench.exe --errors
```

To see all the options, use the following command:

```
testbench.exe -h
```

Details of functions in sendreport.dll

sendreport(url, testdate, phone, description, name, email, pdffile, sid, pid)

Parameters:

- URL: The URL where testbench.exe is running. Note that this URL is to be picked from configuration parameters and must not be hardcoded. Note that this must correspond to the place where testbench.exe must be running. So if testbench.exe is running on IP address 10.0.2.15 on port 8000, the URL should be http://10.0.2.15:8000/
- testdate: Example string 2016-05-26T05:50 (which means that this test was done at 05:50 am of 26-may-2016). Time is expected to be in UTC. This is the date and time when the test sample was collected.
- phone: A string comprising 12 digits of mobile number of patient i.e. the number +91-9876147766 will be sent as 919876147766.
- description: some details about the report. e.g. type of tests done. Note this must be brief (<= 30 characters). So what is typically sent when SMS are sent can be used here.
- name: Name of patient.
- email: email of patient if it is available. If not available, send "".
- pdffile: the path to the PDF file which contains the report. The file must exist otherwise sendreport.dll will return an error.
- sid: the SID of the report.
- pid: the patient id of the report. If not available, send "" (empty string).

Return Value:

0 if successful, non-zero if unsuccessful.

getresult()

Parameters: None

Return Value: A string describing the error encountered.

Important Note

Note that sendreport.dll is not a COM component so please don't try to register it using regsvr32. It will fail. sendreport.dll is a plain DLL that needs to be used in VB using the syntax `declare function` in VB. Please see the VB example provided on how to use this DLL.

Also note that sendreport.dll must be loadable as per standard windows rules (See [https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx)). A simple way to handle this is to put the sendreport.dll in the same place where the program that invokes it will reside.