IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1315)
Available online at: https://www.ijariit.com

Remote control tool to test the interfaces of IoT hub

Shreevidya S.
shreevidyas25@gmail.com
RV College of Engineering, Bengaluru, Karnataka

Nagaraju P.
nagarajup@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

Manoj Kumar Podha
manoj.kumar.podha@philips.com
Philips Innovation Campus, Bengaluru, Karnataka

Abstract— The proposed Remote-Control service tool allows remote connection of the medical devices with the IoT hub software. The hub provides proactive services to the associate devices and these devices are connected to the hub via the proposed tool. The tool realizes the features offered by the hub and is capable of performing registration of the associate devices via unique identification numbers such as serial number and material number, or the system identifier along with attribute update feature. The tool is deployed via CICD pipeline using Azure DevOps. The functional correctness of the code and its comprehensive analysis are done by unit testing and generation of code coverage report respectively.

Keywords— *Angular, Azure DevOps, code coverage, Electron, Jasmine, Registration, Simulator*

1. INTRODUCTION

Remote device management and service capabilities are critical to the maintenance and serviceability of medical devices thus providing fast and efficient troubleshooting, remote configuration capabilities. IoT for Services solution satisfies the end-to-end need of the remote device management and serviceability for the Philips Healthcare Systems. The designed remote-control service tool ensures the proper working of various devices by proactive analysis thereby reducing the cost and time of service.

The objective of this paper is to design a remote-control service tool that will monitor the medical devices in the field i.e., hospital and provide proactive services to the customers and thus improve remote resolution rate and minimize the field visit rates. Various associates (medical devices) such as MR machine, CT scanner, ultrasound scanner, Xray machine etc must be serviced regularly to ensure its proper maintenance as any damage to it would result in huge loss to the hospitals. In addition, when there is a problem encountered in the device, the field service engineer has to go to the location and service the device which leads to waste of time and money. So, it is necessary to constantly monitor the device remotely and perform a proactive analysis to avoid any damage to the device. Unique features of the project include enabling remote connectivity, service capability, system reachability, device registration, software upgradation-Installation of patch, s/w download etc, device profile update, device file upload and automatic deployment of software, all being the functionalities offered by the IoT hub software. The designed remote-control tool which is a simulator, realises all the functionalities mentioned above and it is used as a validation tool during the verification of the IoT hub software.

The outcome of the paper is to enable the remote-control service tool to achieve capabilities in Philips medical devices to keep devices connected and reachable.

2. LITERATURE REVIEW

The literature survey briefly introduces the course of advancement in the design of Remote-Control Tool. In this section, the review of the current works carried out in the field of IoT for providing proactive services are discussed.

Isaac O Olalere describes the Remote Condition observance (RCM) of elevators with conditions to boost proactive maintenance and reduced manning [1]. It employs 2 elevator parameters like acoustics and vibration, by activity IoT device for fault indication and remote knowledge acquisition. exploitation this technique, breakdown maintenance of the system is optimized to proactive maintenance. Proactive solutions also can be provided for home observance which will be supported the perceptive of the particular home automation events [2]. By analyzing the user habits, reinforcement algorithmic rule proposes an answer providing automatic ideas which might adapt to the user's talents. This technique reduces the hassle and movement of actions. It helps older folks to own Associate in Nursing autonomous life.

[3] provides a completely unique recommendation engine for personalized suggested wearables and IoT solutions for any given individual. It uses a mathematical optimization model that's developed to advocate the best solutions for a personal, so facultative proactive health observance. during this paper, the matter of proactive observance of one's health via personalized set of wearable technologies and IoT solutions to be used is addressed. a brand-new approach called Cassandra, that combining design-time and run-time analysis techniques, will "look ahead" within the close to execution future, and predict potential failures is introduced by Pengcheng Zhang [4]. Since

unsafe run-time changes compromises correct execution of the whole system, advanced monitors should be outlined with the power to predict and forestall the potential errors happening within the future. this will be eventually custom-built to the OSGi part framework.

Marco Angelini [5] has planned PERCIVAL, a unique visual analytics atmosphere that contributes to situational awareness and monitor security events that can happen on the system. It permits security operators to know each the static and therefore the dynamic risk level of the system they monitor. As such, it provides higher understanding of the risks visage by the system within the absence of attacks and permits security operators to watch security events that happen on the system. Min Zhu [6] proposes a wireless communication system victimization 2 legitimate monitors, one monitor is employed because the cooperative relay, accustomed improve the general data transmission rate, the opposite receiving suspicious data and causation ECM signals. this is often principally accustomed get the utmost eavesdropping rate by collectively style the transmit beamforming of 2 monitors.

The problem of predicting whether each time window will infringe an aggregate performance constraint, at a series of checkpoints within the window can be addressed by performance monitoring [7]. Here, 3 types of measures are often estimated: what performance outcome every instance can yield, what percentage instances can begin within the remainder of the window, and what their performance outcomes are going to be by using regression technique. associate analysis is dispensed on knowledge set, analysis setting, parameters setting and results, etc. To mitigate the problems of smart city scenario involving a large crowd, Bikash Choudhary [8] proposes a service replication theme to boost service response-time, service availableness and system wide resource utilization. It uses a twin threshold primarily based proactive sensing mechanism for distinctive the services to be replicated. However, the performance gain depends on the putting in place and the adaption of huge range of system parameters. Abnormality detection [9] in station being a typical IoT application are often addressed by proactive knowledge service model that encapsulates stream detector knowledge into services. By the analysis of event correlations, service hyperlinks are often realized that offers proactive real time interaction with services. It helps to make software-defined ‘sensors’ which will collaborate with associate degree other in an IoT context.

[10] proposes the group-based Incentive and Penalizing Schemes for Proactive democratic information Sensing (GRIPS-PPDS) in IoT Network. GRIPS-PPDS applies the minimum set cowl theorem to pick the nodes that more sense and mixture the information with high precision rate, consistency and dependability. The planned model implements the relaxed and rigid modes for proactive information sensing in IoT network. supported attributes like quality of knowledge, consistency, information accuracy rate, dependability and node trait, the GRIPS-PPDS theme defines the motivation and penalizing factors to minimize the energy consumption, optimize the coverage region and make the network secure.

A large variety and complexness of personalized services are often dealt by the proactive personalized service investing advanced Event process (CEP) [11]. First, a directed acrylic graph will be realized for complex event pattern. Then, clustering algorithms and complex event pattern partitioning is proposed. Finally, the proposed approach, BCEPCare is used to realize a prototype partitioning. By investing the advantage of CEP in information streaming, a stratified fog-cloud computing CEP design for personalized service to accelerate latent period and cut back resource waste is additionally planned [12]. beginning with the planned design includes fog layer, sensing element layer and cloud layer, a clump approach to optimize the fog and cloud computing is introduced. Finally, the design by name FogCepCare is enforced within the model system.

A positive insureTech feedback is delivered by employing a proactive fintech model that uses IoT intelligence [13]. FinTech model speaks of actual price and offers a horny service price and client satisfaction. On Shelf availableness (OSA) being a key metric to confirm shopper expertise [14], is enabled by machine-controlled web of things and web of everything. To avoid the consumers filling the shopper’s list while not encountering out-of-stock scenario, the present method gaps is showing intelligence machine-controlled to reinforce the OSA while not the human intervention.

Usman Ashraf [15] proposes a proactive network strengthening resolution known as PROSE that focuses on planning resilient IoTs by proactively distinguishing important nodes within the network in order that they'll be protected by putting in backups. Simulation results of associate economical heuristic algorithmic program validates that PROSE identifies the foremost vulnerable IoT nodes.

From the above papers, it is understood that proactive services using IoT plays an important role in providing an end to end solution in providing remote services to the medical devices.

3. OBJECTIVES

The objectives of this paper are:

1. Study of various frameworks and languages required for the development of the tool.
2. Design a Remote-Control service tool that realizes all the features offered by the IoT hub software. The various features of the hub are:
   a) Software installation-Upgradation
   b) Device registration by serial number and material number
   c) Device Registration by system identifier
   d) Device attribute update
   e) CICD workflow using Azure DevOps
3. Integrate the cloud calls with the application
4. The code is unit tested along with the generation of code coverage report.

4. DESIGN AND CONCEPT

The IoT hub offers many interfaces such as registration, file upload, device profile upload etc. These interfaces have to be tested before the actual deployment of the hub in the location. Since testing of interfaces using actual devices require too much of investment in cost and maintenance, manual testing using postman tool where the REST calls are made by hitting the URI could be preferable. The disadvantage of using the postman is that there can be errors while testing due to wrong data entry or format. This can be solved by using a simulator, where all the input data is hidden inside and hence there will be no errors while testing. The general concept of the hub deployment is shown in the figure 1. The designed tool is used to successfully test the services that the hub is capable of providing remotely.

Simulator is a virtual version of a real-time object that performs all its operations. The key characteristics and behaviors of the actual tool that are likely to occur can be reproduced in the test conditions. The workflow of development of the tool is shown in the figure 2.

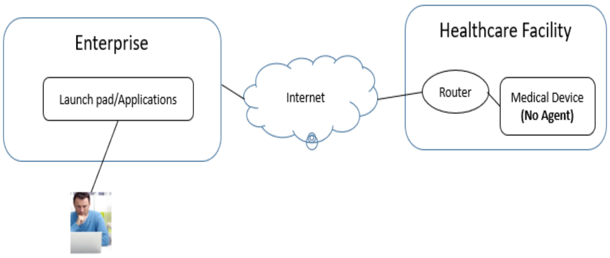


Figure 1: General Concept

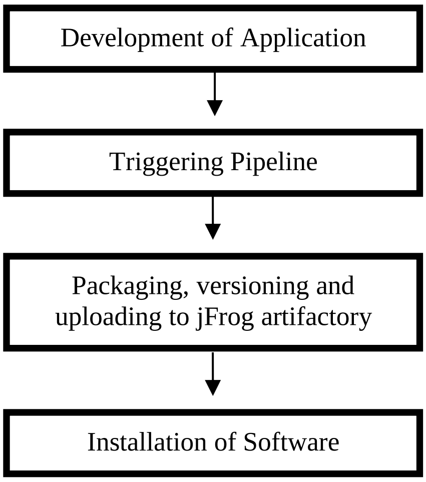

Development of Application
Triggering Pipeline
Packaging, versioning and uploading to jFrog artifactory
Installation of Software

Figure 2: Workflow of development of the tool

5. SOFTWARE INSTALLATION/ UPGRADATION

Any application that is developed needs to be distributed so that it can be deployed in a different system. This requires the executable file to be created followed by the generation of OS specific installers. Also, the application needs to be upgraded whenever a feature is added to the application. The following tools are required for the software installation/ upgradation.

A. Electron

Electron is an open-source framework developed and maintained by GitHub. It allows for the development of GUI applications (desktop applications) using web technologies such as HTML, CSS, JavaScript etc. It combines chromium rendering engine and the NodeJS runtime. An electron application consists of several processes. It has one main process which runs the application logic, and many renderer processes rendering HTML and CSS which are basically the windows that appear on the user screen.

B. Electron Builder

Electron-builder framework gives a complete solution for packaging and building an electron app that is ready for distribution. Packaging refers to creation of executable files and building implies generation of installers based on the operating system. The tool is compatible for MacOS, Windows and Linux with an auto update support.

C. Electron Updater

Electron updater module allows automatic updating of an application. A new version number will be overwritten when the application is updated. The version id incremented based on the valid argument passed like patch, minor, major, etc. A patch change means that we increment the rightmost digit by one. Changing software from 1.0.0 to 1.0.1 is a small change, and usually implies a bug fix. A minor change means that the software version increases from 1.0.0 to 1.1.0. We deal with more severe changes as we increase the middle digit by one. This number should increase when new functionality is added to the software, and it should still be backward compatible with the 1.0.0 version. With a major change, the version number increases from 1.0.0 to 2.0.0. Things might have changed so much that constructs have been renamed or removed. It might not be compatible with earlier versions.

D. Electron Log

Logging refers to the events that reflect the different aspect of an application. It is the easiest way to troubleshoot and diagnose the application. Logs give data concerning the severity of the matter, similarly as insights into its root cause. The severity can be categorized into logging levels which helps in categorize application activity so we can easily pinpoint errors. Electron-log is a simple logging module for Electron or NW.js application. Zero being the highest priority and 3 lowest, Electron-log supports the following log levels shown in the table 1.

Table 1: Log levels

Logs	Priority Level
Error	0
Warn	1
Info	2
Trace	3

6. DEVICE REGISTRATION

Device registration is a process of onboarding an associate device to the Philips remote services by registering it with or via the IoT hub. Registration will be done once there is a mutual authentication of the certificate of SSL certificate between client and the server.

A. Generate Certificates

Connection between client and the server will be enabled once there is a mutual authentication. Mutual authentication is a verification scheme to ensure data security. Verification schemes are based on either passwords or certificates. In this project, mutual authentication is done through Philips certificate verification. The protocols used for verification are:

- SSL protocol: Secure Sockets Layer (SSL) protocol is an encryption-based protocol principally used to guarantee authentication, privacy and data integrity. SSL can be enforced solely by the websites that have SSL certificate. It is stored and displayed on a website's or application's server.
- HTTPS protocol: It is a secured form of HTTP. A website implementing SSL certificate will have HTTPS instead of HTTP.

B. Make REST calls

POST, GET methods can be used to register a device, get status of the registration and so on. The status of the registration can be obtained both by using the nametag of the device and the jobid. Jobid is obtained when the user tries to register the device either by serial number and material number, or the system identifier. The registration status can be registered, deregistered, registration failed. These details will be updated in the hub management portal. A popup message will be shown when the user tries to register or tries to get the status of registration based on nametag or jobid. A detailed description can also be obtained from the popup window. The list of status of registration can be as follows:

1. **Not Registered**: When a device is added to the dashboard, the initial registration status of the device will be in not registered state.
2. **In Progress**: When the user hits on the register button, the status will change to the in-progress state.

3. **Registered**: Registration of a device can be done in two ways, either based on the serial number or the system identifier. A device can be registered with a particular nametag having either serial number or system identifier. The nametag will be unique to every device that is added.
4. **Registration Failed**: The registration of a device might fail either because the server is down or when there is any authorization error.

7. DEVICE ATTRIBUTE UPDATE

The devices can be added to the remote-control service tool by selecting the desired registration type, and based on the type of registration, the user has to select the required business name, product type, serial number and material number or the system identifier, timeout etc. Once the device id added, it gets updated in the dashboard where the user can edit the details of the particular device selected. The edited details can be viewed on the user interface by selecting on the device.

8. CICD USING AZURE DEVOPS

Deployment of the software is carried out using Azure pipeline cloud service that is capable of continuous integration and continuous delivery. The pipelines are basically used to build and test the code and ship it to the target of choice. Azure has many advantages like controlling version, different language and application types, deployment target etc. The pipeline consists of two stages – build pipeline and release pipeline. Build is the stage where the application is compiled while release stage involves delivering the application to the repository. All the build operations are configured on the system that is considered as agent or the build server. There can be different agents performing different operations or same agent performing different operations. An agent can have different tasks and these different set of tasks make a pipeline. Each task holds four distinct folders for artifactory, binary, source and test results. Whenever any change is updated in the repository, the pipeline will be automatically triggered and then the content in the agent system will be refreshed. Whenever the pipeline triggered, a build will be generated and uploaded to the jFrog artifactory. It contains the installer and the supporting binary files, which can be directly downloaded and used by any user. Figure 3 shows workflow of the CICD using Azure DevOps.

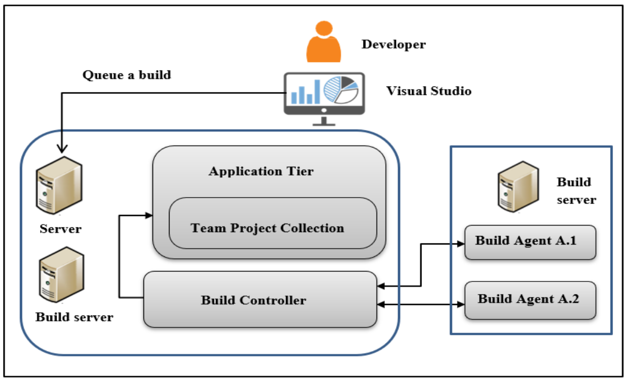


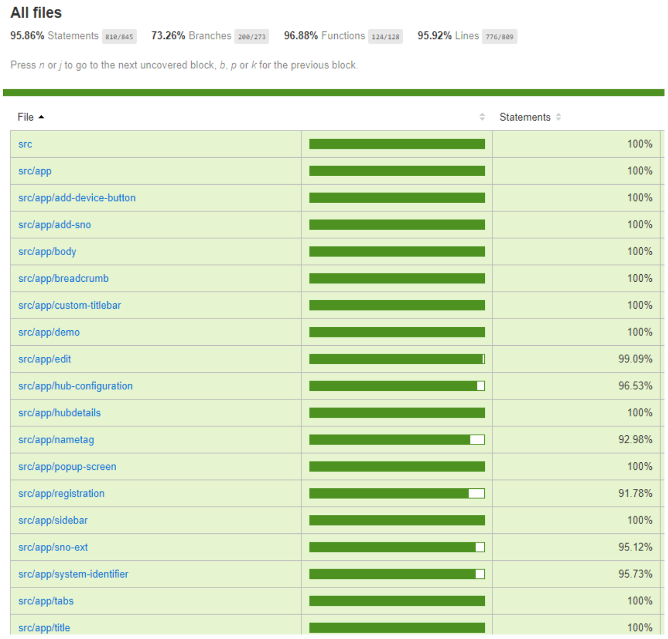
Figure 3: CICD workflow using Azure DevOps

9. SOFTWARE TESTING

Software testing involves various processes to evaluate the designed software product in order to meet the required conditions. Unit testing is a part of software testing strategy usually carried out by the developers themselves and hence concerned with functional correctness. Jasmine is a popular Behavior Driven Development (BDD) framework which runs on any JavaScript enabled platform. It is an open-source framework mainly used to perform unit testing. Jasmine has an inbuilt test runner by which we can run the browser tests using a simple JavaScript runner tool called Karma.

Angular application can be tested in the angular testing utility which consists if classes and functions that are required to build a test environment. Testbed is a class that configures and initializes the environment, while the function configure Testing Module provides all the necessary dependencies for the component. The test cases are run in the karma testing environment.

Code coverage is a testing metric using which helps in comprehensive analysis of a software. Coverage determines if the lines of code of a software is successfully validated in the test procedure. It is useful in maintaining the quality of the code as well as assessing the test performance. It also helps in dead code detection and elimination which is beneficial to the developers. The code coverage report of the developed tool is as shown in the figure 4.



All files

95.86% Statements 810/845   73.26% Branches 200/273   96.88% Functions 124/128   95.92% Lines 776/809

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File ▲		Statements
src		100%
src/app		100%
src/app/add-device-button		100%
src/app/add-sno		100%
src/app/body		100%
src/app/breadcrumb		100%
src/app/custom-titlebar		100%
src/app/demo		100%
src/app/edit		99.09%
src/app/hub-configuration		96.53%
src/app/hubdetails		100%
src/app/nametag		92.98%
src/app/popup-screen		100%
src/app/registration		91.78%
src/app/sidebar		100%
src/app/sno-ext		95.12%
src/app/system-identifier		95.73%
src/app/tabs		100%
src/app/title		100%

Figure 4: Code coverage Report

10. CONCLUSION

The designed tool is successfully able to register the associate devices both by the serial number and material number and the system identifier. The status of the registration can be obtained both via nametag of the device and the unique id obtained during the registration of the device. The tool is successfully tested using jasmine framework along with the generation of the code coverage report.

11. FUTURE SCOPE

The **scope** of this paper is:

1. The Remote-Control service tool monitors the medical devices in the field and provide proactive services to customer.
2. The tool realizes all the features offered by the IoT hub software.
3. Improve remote resolution rate and minimize the field visit rates.
4. It can be used as a validation tool during the verification of the IoT hub software.

REFERENCES

[1] Isaac O Olalere, and et.al, "Remote Condition Monitoring of Elevator's Parameters for Optimized Maintenance Using IoT

Technology," 2018 *IEEE Canadian Conference on Electrical and Computer Engineering*, 2018.

[2] T.B.T Truong, Frizon de Lamotte, "Proactive Remote Healthcare Based on Multimedia and Home Automation Services," 5th Annual *IEEE Conference on Automation Science and Engineering*,2019.

[3] Shubhi Asthana, and et.al, "A Recommendation System for Proactive Health Monitoring Using IoT and Wearable Technologies," *IEEE, 978-1-5386-1999-5*, 2017.

[4] Pengcheng Zhang, and et.al, "Run-time systems failure prediction via proactive monitoring," *IEEE*, 978-1-4577-1639-3/11, 2011.

[5] Marco Angelini, and et.al, "PERCIVAL: proactive and reactive attack and response assessment for cyber incidents using visual analytics," *IEEE, 978-1-4673-7599-3/15*, 2015.

[6] Min Zhu and et.al, "Legitimate Monitoring via Cooperative Relay and Proactive Jamming," *IEEE*, 2019.

[7] Francesco Folino and et.al, "A Prediction Framework for Proactively Monitoring Aggregate Process-Performance Indicators," 2020 *IEEE 19th International Enterprise Distributed Object Computing Conference*, 1541-7719/15, 2020.

[8] Bikash Choudhary, Subhrabratha Choudhary and Animesh Dutta, "A proactive context-aware service replication scheme for Adhoc IoT scenarios," *IEEE Transactions on network and Service Management*, vol. 16, pp. 1797-1811, 2019.

[9] Shouli Zhang, Chin Liu, and et.al, "A proactive data service model to encapsulating stream sensor data into service," *14th Web Information systems and Applications conference*, 978-1-5386-4807-0, 2018.

[10] M Bala Krishna, "Group-based incentive and penalizing schemes for proactive participatory data sensing in IoT networks," *2018 IEEE 4th World forum on Internet of Things*, 2018.

[11] Shuqing He and et.al, "Proactive Personalized services in large-scale to IoT based healthcare application," *2018 IEEE International Conference on Web Services*, vol. 10, 2018.

[12] Shuqing He and et.al, "Proactive Personal Services through Fog-Cloud computing in large scale IoT-based HealthCare Application," *China Communications*, vol. 14, pp. 1-16 2017.

[13] Zahraa Marafie, Kwey-jay Lin and others "ProActive FinTech: using Intelligent IoT to deliver positive InsurTech Feedback," *2018 IEEE 20th Conference on Business Informatics (CBI)*, 2018.

[14] Rajesh Vargheese and Hazeem Dahir, "An IoT/IoE enabled architecture framework for precision on shelf availability: Enhancing proactive shopper experience," *2015 IEEE International Conference on Big Data, 2015.*

[15] Usmaan Ashraf, "PROSE: Proactive Resielence in internet of things: Targeted attacks and countermeasures," *IEEE Sensors Journal*, vol. 18, pp. 10049-10057, 2018.