PUSH TECHNOLOGY INTEGRATION IN ONLINE SERVICE THE PRINTING EQUIPMENT MAINTENANCE

***T. Neroda**, PhD in Engineering,*
Associate Professor in Department of Automation and Computer Technologies
Ukrainian Academy of Printing

The development of the field of providing services to printing companies is gradually occupying its well-deserved market niche. The main task of such support institutions is the prompt provision of maintenance services and adjustment of multi-purpose equipment for printing infrastructure [1]. To ensure the basic functions and elasticity and performance increase there is a need to use the system of operational communication of services for printing equipment [2].

To expand the functionality of the designed online service for printing equipment, it was decided to integrate push-messaging technology into corporate application. Their main purpose is to provide communication between mobile terminals and directly to server platform. Such categorized small information blocks may appear on end terminal screen where alert area is implemented. The push message pops up on top of all open windows and lingers on device screen for a short time. Modern mobile platforms allow to receive such messages even when application is closed. Thanks to push-messages technology availability in online service of printing equipment maintenance, adjusters will no longer need to periodically run a mobile application to obtain up-to-date information about current requests, as it allows the server to self send data to mobile devices. By interactively transmitting data without time costs to verify information flows, push messaging technology also synchronizes already transmitted data if it is updated.

The functionality of a number of modules was expanded in project to integrate push messaging technology into the online service printing equipment maintenance. First of all, it is the central server of institution providing support services, which sends push-messages (Fig. 1, vertex #1). Following, the server of subscriber mobile application sends a target notification to push message server. It is also necessary to ensure a constantly running process in the operating system of end mobile device that communicates with the push-message server.

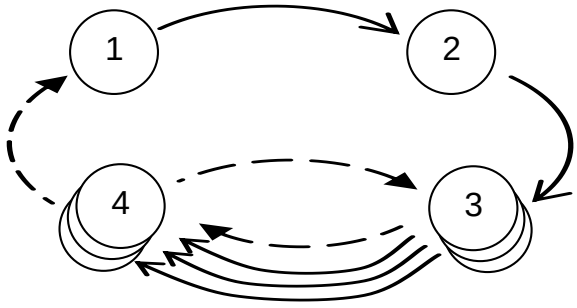

Figure 1 – Operation principle of push-messages in online service of printing equipment maintenance

Finally, the main changes took place in the functional module of mobile application, which supports push notifications. Because this application is cross-

platform, from server side there was a need to share the saved unique device identifiers across platforms and send requests to appropriate push servers. It was also taken into account that for each platform is ought to use the appropriate push server (vertex #3): for iOS it is Apple APNs [3], for Android – Google Cloud Messaging [4] and so on.

To reduce the time spent on the implementation of presented functionality, it was decided to use Google Firebase Cloud Messaging (vertex #2). It is a cross-platform solution to exchange cloud messages and messaging from Google, which allows developers to send push messages to end users of applications on different platforms through Firebase Notification Composer or through APIs provided by Firebase [5].

Its key feature is universal targeted notification, which allows to address push messages to a specific device, devices set, or disparate devices accepted in the publishing-subscription infrastructure for a specific corporate role. Thus, the classic notifications sending allow delivering messages with data and regulating the production behavior of an individual corporate community. In turn, sending messages from client applications gives the opportunity in addition to confirmations allows to deploy chats or return notifications from end devices to server via a reliable and efficient Firebase Cloud Messaging connection channel.

After the end user gives permission to the application to receive push notifications, the process of registering a mobile device can be divided into three stages. First of all, the application from the set of corporate end terminals (vertex #4) sends its unique ID and device number to the push message server (address edge #4—#3). These two unique numbers form a complex individual identifier that is registered and sent back from the application server in response to the registration request (address edge #3—#4).

Finally, the application sends the received identifier to the web server for storage in the corporate database (address edge #4—#1). Next to this identifier, in the process of creating a push message, the web server generates relevant content along with a list of unique identifiers of devices that should receive messages, and sends this data using a special API to the Firebase Cloud Messaging server (information edge #1—#2), which distributes the received list on mobile platforms and sends them to the appropriate push servers (information edge #2—#3). Push messaging servers, in turn, forward these messages to mobile devices to inform the relevant staff about the available workload in a timely manner.

References

1. Neroda, T. (2019). Designing of multilevel system the distributed resources administration in polygraphically oriented network infrastructure. *Computer technologies of printing*, (42), 64-72.

2. Neroda, T. (2022). Integration's means study of protection mechanisms for polygraphically oriented network infrastructure. *Automation and computer-integrated technologies in industry and education*, (10), 32-34.

3. Notifications Overview – Apple Developer. developer.apple.com/notifications.

4. Cloud Messaging | Google Developers. developers.google.com/cloud-messaging.

5. Firebase Status Dashboard. firebase.google.com/support.