

СТВОРЕННЯ СИСТЕМИ ПОШИРЕННЯ PUSH-ПОВІДОМЛЕНЬ

***Abstract.** The model of the system was implemented in the work. It allows to trace the quality and the passed way to deliver the notification. In order to create a system for transmitting push-messages, model variants of use, function trees were constructed. The results can be used to create a push message system for customers who have mobile devices. The scientific novelty consists in the development of a methodological basis for assessing push-technologies; developing a model information system for the transmission of push-notification.*

***Keywords:** push-notification, message transfer, user, information system, mobile technology.*

Вступ і постановка задачі.

В даний час рівень операційних систем мобільних пристроїв дозволяє технологічно використовувати різні форми отримання інформації та контенту через додатки і багато в чому вже не залежати від провайдера телекомунікаційних послуг. Більш того, використання відкритих інформаційних технологій в програмуванні збільшує конкуренцію на ринку програмного забезпечення, що створює передумови до підвищення прикладного рівня розробок і дозволяє користувачеві розширити функціональні можливості своїх пристроїв. Такі можливості для розробки нових напрямків сьогодні представлені більшістю виробників операційних систем. При цьому основні гравці на ринку (Google, Apple) намагаються максимально безпечно відкрити доступ до пристроїв ззовні для зв'язку з абонентом. Одним з перспективних напрямків в цій області можна виділити створення і розвиток системи push-повідомлень для безпосередній відправки повідомлень, графічного матеріалу і іншої інформації за допомогою мережі Інтернет на додатки мобільного пристрою, веб-додатки тощо. У цьому випадку користувач системи отримує можливість цілеспрямованого одностороннього спілкування з зацікавленими особами.

1. Основні характеристики push-технологій

Push-технологія використовується як в глобальній мережі Internet, так і в корпоративних Intranet-мережах і призначена для примусової

доставки за допомогою Web-каналів різноманітної інформації (в тому числі, різної інформації освітнього призначення). Поширення повідомлень документів - саме та область, в якій push-технологія найбільш ефективна. Web-канали - це чудовий транспорт для поширення інформації, яка змінюється щотижня, щодня, щогодини або навіть ще більш динамічно. Освітні Web-канали дистанційного навчання, які приймають нові уроки кожен день, - це лише один приватний приклад, наочно демонструє можливості застосування Push-технології. За допомогою push-систем можна здійснювати розсилку різного роду повідомлень, оголошень, пересилання в фоновому режимі великих за обсягом файлів, автоматичне оновлення програмного забезпечення, автоматичне завантаження будь-яких часто оновлюваних матеріалів і даних.

Технологія дозволяє реалізувати на сервері різні варіації доставки інформації:

- одночасна розсилка інформації тисячам користувачів по єдиним централізованим запитом;

- диференційована розсилка інформації за ролями чи групами користувачів;

- розсилка інформації окремому користувачеві.

Користувачі отримують необхідну інформацію, попередньо встановивши відповідні параметри клієнтської частини при підписці на канал; надсилати запит на цю інформацію не потрібно. При цьому web-сервер стає активним джерелом інформації, "примусово поширюючи" її по мережі і звільняючи користувача від необхідності пошуку і "витягування" потрібної інформації.

Форми відображення інформації, що надаються клієнтським програмним забезпеченням (ПЗ) push-систем, можуть бути різноманітними. Подання інформації може здійснюватися: на персональну Web-сторінку, в окреме вікно push-застосунку, інформація може бути передана клієнту електронної пошти.

Push-повідомлення - це короткі спливаючі повідомлення, які з'являються на екрані мобільного телефону і повідомляють про важливі події та оновлення. [1,2] Вони підтримуються практично в кожній мобільній операційній системі (ОС). Самі по собі вони є засобом представлення інформації користувачеві і можуть бути інтерактивним елементом, що дозволяє переглянути інформацію докладніше після здійснення кліка по елементу. Вони не залежать від наявності або відсутності Інтернету, і можуть відображатися, наприклад, в залежності від положення телефону в просторі (наприклад, використовуючи GPS), або просто повідомляти користувача телефону про поточний час доби. Однак ми в даній

роботі розглядаємо push-повідомлення в зв'язці з Інтернет сервісом, що забезпечує ведення Інтернет-розсилки.

Доставка повідомлень в системах push-повідомлень здійснюється мережею Інтернет, використовуючи клієнт серверну модель. Мобільний клієнт (subscriber) за допомогою встановленої програми підписується на розсилку push-повідомлень, а сервер публікацій (publisher), по настанню певної події, здійснює розсилку підписалися клієнтам. Таким чином, адресатом повідомлення завжди є деякий застосунок. Важливо відзначити, що push-повідомлення дозволяють серверам розсилки повідомляти користувачів про настання подій, навіть якщо застосунок в даний конкретний момент неактивно. Досить, щоб застосунок було встановлено, і діяла підписка.

Відправка на сервері, як правило, здійснюється при настанні деякої події. Сама подія відстежується деяким окремим додатком, і може бути будь-яким: отримання повідомлення від адміністратора, або знаходження деякої кількості клієнтських пристроїв поблизу (може використовуватися пасивний Wi-Fi або Bluetooth моніторинг). Системи, що дозволяють деяким чином обробляти інформацію, в сукупності з системами доставки цієї інформації і пов'язані з цим апаратні, програмні та людські ресурси, називаються інформаційними системами [5].

Спочатку системи створювалися автономними: ініціатором події міг бути або адміністратор системи, або виконання зумовленої події. Проте наступним етапом у розвитку концепції розсилок на базі push-повідомлень стала поява сервісів, які надають інтерфейс програмування додатків (application programming interface, API) для налаштування власних розсилок. Сервіси надають своїм користувачам, які будуть підписані на розсилку, можливість налаштування контенту розсилки, а так само завантаження клієнтських додатків, реалізованих для різних мобільних платформ (iOS, Android тощо).

Ці сервіси є досить популярними, проте у них є один недолік з точки зору більшості комерційних підприємств – це необхідність володіння навичками програмування. Коли у будь-якої організації виникає потреба в організації розсилки (в тому числі це може бути розсилка для вузького кола осіб - наприклад, для співробітників компанії), то необхідність наймати програміста, який має навички створення мобільних додатків, до того ж для декількох мобільних операційних систем – це додаткові витрати, яких хотілося б уникнути.

Тому виникає потреба в подальшому ступені розвитку сервісів - це сервіси, які, крім представлення API для створення і ведення розсилки, надають

мобільний застосунок, що працює з даними API і встановлюється на мобільний пристрій клієнта, а також панель управління розсилкою push-повідомлень. Такі сервіси є вже готовим бізнес-рішенням для організацій, згаданих вище: при необхідності вести розсилку певному колу осіб, потрібно просто зареєструватися, далі при необхідності здійснити оплату, скачати клієнтську програму і налаштувати розсилку в особистому кабінеті панелі управління. Такі сервіси вже можна назвати реалізацією інформаційної системи.

Вочевидь, якщо суб'єктом підписки на push-повідомлення є застосунок, то така система повинна пропонувати не тільки систему управління розсилками, але і мобільний застосунок для прийому повідомлень. При цьому виникає один важливий момент. Щоб уникнути програмування, мобільний застосунок має бути універсальним. Але це може суперечити політиці конкретних компаній (не використовувати ті ж рішення, що і конкуренти). Відповідно, універсальний мобільний застосунок має якимось конфігуруватися. Тоді кожен бізнес-користувач зможе створити "власну" версію мобільного застосування і поширювати її вже від свого імені, як свій власний застосунок. У разі створення такого мобільного додатка для підписаних клієнтів буде вирішена проблема з програмуванням [2]. Іншим вирішенням цієї проблеми є саме відмова від програмування. Це буде деякий аналог системи управління контентом (CMS) для Веб, який охоплює всі етапи, включаючи мобільний застосунок.

2. Порівняльна характеристика сервісів на базі push-технологій

Сервіси push-повідомлень можуть бути реалізовані на основі різноманітних платформ. Для аналізу їх застосовності представимо характеристики деяких відомих платформ, які використовуються для реалізації сервісу проведення інформаційних розсилок на основі push-повідомлень (табл.1). Для порівняння будемо використовувати такі характеристики, як: підтримка різних операційних систем, наявність консолі для управління розсилкою, можливість конфігурувати повідомлення та засоби створення додатку.

Наведені в табл. 1 сервіси дозволяють вирішити наступні задачі:

для створення розсилки необхідно створити облікові записи, після чого контент розсилки можна задати в консолі управління;

є готові рішення для інтеграції сервісу з мобільним додатком: від бібліотеки, що вбудовується в вихідний код програми, до готового додатка, налаштованого на взаємодію з конкретним сервісом;

Порівняльна характеристика інформаційних систем
для технології push-повідомлень

Назва сервісу	Підтримувані пристрої	Консоль управління розсилкою	Можливості налаштування	Засоби створення мобільного додатку і доставки його користувачам
1	2	3	4	5
Amazon Simple Notification Service	iOS, Android, Fire OS, Windows тощо	Стандартна	Можна вказати тему, текст, звук, посилання для переходу і зображення	Готовий застосунок і програмний код для вбудовування
Windows Azure Notification Hubs	Windows 8, Windows Phone, iOS, Android	Стандартна	Можна вказати тему, текст, звук, посилання для переходу і зображення	Програмний код для вбудовування і тестовий застосунок
Urban Airship Push-notifications	Windows 8, iOS, Windows Phone, Android	Стандартна		Програмний код, вбудовується в застосунок
Jeapie	iOS, Android, Windows та інші	Стандартна	Візуальний конструктор повідомлень	Готовий вихідний код для вбудови в застосунок . Дається шаблон застосуноку.
Push2Press	iOS, Android	Як окрема CMS, так і плагін для WordPress	Стандартні	Можливість сконструювати застосунок онлайн, завантажити готовий шаблон, і зібрати його в настільному середовищі

Закінчення табл.1

1	2	3	4	5
OpenPush	iPhone, Android, Blackberry, Windows Phone тощо	Стандартні	Стандартні	Для поширення надається готовий шаблон застосунку
Nudge	iPhone, Android, Blackberry, Windows Phone тощо	Стандартні	Стандартні	Для поширення надається готовий шаблон застосунку

супровід розсилки: як для платного веб-сервісу, так і сервісу з відкритим вихідним кодом.

зберігання повідомлень, доставка та захист інформації здійснюється сервісом.

В той же час існують проблеми, які залишилися невирішеними. З боку клієнта застосунок може бути або стандартним (готове рішення від сервісу), або написаним користувачем сервісу самостійно з використанням бібліотек сервісу або шаблонів. Але якщо клієнт хоче унікальний застосунок, ці сервіси не вирішать таку проблему.

В даний час ця технологія є одним з варіантів надання контенту користувачам, наприклад, вартості акцій на біржах, прогнозу погоди, або будь-якого іншого інформаційного повідомлення, і не вимагає від користувача прийняття дій щодо відправника. Дана технологія активно починає використовуватися в силу мінімальних необхідних витрат для її запуску.

Розвиток push-технологій в мобільних пристроях передбачає, що основний застосунок знаходиться в неактивному режимі, і покликана насамперед подолати ряд обмежень, з якими стикаються користувачі і розробники:

обмежений ресурс операційної системи;

відносно високий рівень оплати послуг за контент провайдером (SMS, MMS);

високий рівень конкуренції на ринку мобільних операційних систем і як наслідок, розвиток додаткових безкоштовних сервісів;

складність реалізації взаємодії з телекомунікаційною інфраструктурою.

Дані фактори створюють основу для популярності мобільних додатків, які сприяють розвитку інфраструктури для передачі push-повідомлень.

Ці служби підтримують дуже економічне з'єднання з мобільними пристроями і здійснюють доставку push-повідомлень. ОС мобільного пристрою перенаправляє повідомлення відповідного застосунку. У додатках під Windows Phone повідомлення з'являються у верхній частині екрана. У додатках під Android та iOS повідомлення з'являються у верхній частині екрану в панелі повідомлень.

Однак ці сервіси надають саме API, під який можна запрограмувати застосунок, який з ним взаємодіє. Також необхідно самостійно створити сервіс для управління розсилкою. Але це не завжди зручно, що і зумовило появу інформаційних систем, які будуть розглянуті далі.

3. Архітектура інформаційних систем для push-повідомлень

У всіх розглянутих рішеннях користувачеві надається можливість здійснювати такі види розсилок:

1) Широко умовна розсилка. Після вказівки контенту розсилки, сервер запитує список зареєстрованих клієнтів, після чого їм здійснюється розсилка.

2) Персоналізована (індивідуальна) розсилка. При вказівці контенту розсилки публікатор також вказує унікальний номер одержувача (зі списку наявних та зареєстрованих користувачів), і за цим номером здійснюється доставка

3) Сегментована розсилка. Середнє між першими двома видами: публікатор повинен вказати умову на дані одержувача. Сервер вибирає тих підписаних клієнтів, дані яких задовольняють умові, після чого їм здійснюється доставка.

Інформаційні системи push-повідомлень ставлять перед собою такі основні завдання:

1) надати користувачеві інструмент для створення тематичної розсилки;
2) надати можливість здійснювати будь-який з 3-ьох видів розсилок (див. вище), а для сегментованої розсилки надати якомога більш гнучкий спосіб вибірки підписаних клієнтів для відправки повідомлень;

3) надати якомога більш прозорий спосіб підписи клієнтського пристрою на розсилку. Зокрема можливість помістити в мобільний пристрій користувачів за стосунок для підписки на розсилку.

Для вирішення завдань з цього списку, користувачам надаються такі інструменти:

Кожному користувачеві-публікатору інформаційної системи виділяється віртуальний особистий кабінет, в якому користувач має консоль управління розсилками. Найбільш типові можливості консолі: створення розсилки з певної тематики, відправка повідомлення (одним з трьох видів push-розсилки), перегляд статистики відкритих push-повідомлень тощо.

Для здійснення різних видів розсилок використовуються мітки, які присвоюються кожному окремому одержувачу. Це можуть бути як довільні теги, так і універсальні характеристики, такі як версія операційної системи мобільного телефону. Для персоналізованої розсилки можуть використовуватися унікальні ідентифікатори, які генеруються самим мобільним додатком, або ж персональні параметри, що настроюються користувачем - наприклад, день народження власника телефону.

Для того, щоб одержувач міг підписатися на розсилку, необхідно створити застосунок, яка б виконувала підписку в технічному сенсі, і доставити цей застосунок кінцевому споживачеві. Для цього застосовуються різні підходи:

надати програмний код, який потрібно впровадити в власний мобільний застосунок. Таку можливість надають практично всі сервіси, і це є найбільш гнучким способом доставки, проте в той же час найбільш трудомістким;

надати готовий до завантаження «тестовий» застосунок. На противагу попередньому підходу, цей підхід практично не вимагає зусиль з боку користувача-публікатора інформаційної системи, однак позбавлений гнучкості в конфігуруванні застосунку;

надати більш-менш конфігуруємий в особистому кабінеті шаблон мобільного додатку, який користувач-публікатор може завантажити собі на комп'ютер. Після допрацювання за необхідністю застосунок може бути зібрано у відповідному для шаблону середовищі розробки і викладено як свій власний.

Основний спосіб донести створений готовий застосунок до користувача – це викласти його в основний, спеціалізований для кожної операційної системи сховище застосунків. Такі сховища завжди знаходяться в публічному доступі і дають розробникам можливість завантажувати туди свої застосунки. Також публікатор, який бажає поширити свій застосунок, може відправити його по електронній пошті або викласти десь в публічному доступі посилання для завантаження застосунку.

4. Основні мобільні платформи для push-повідомлень

Найбільш поширеними платформами, що реалізують push-повідомлення є Google, Apple і BlackBerry.

Система хмарних повідомлень Google (Google Cloud Messaging, далі GCM) - служба, побудована з використанням ресурсів Інтернету, яка дозволяє організувати розсилання даних зі стороннього сервера користувачам пристроїв з операційною системою Android. Повідомленнями є легковагові дані сягом 4 кВ, які відправляються сервером при настанні певного типу подій. Дана служба управляє чергами повідомлень, що виникають, і доставкою їх до цільової програми.

Можна виділити наступні основні особливості GCM-служби [4]:

- 1) можливість використовувати сторонній сервер додатків для відправки даних;
- 2) застосунок на мобільному пристрої в момент отримання повідомлення може бути неактивним. У разі відповідного налаштування додатка система запустить цю програму для отримання повідомлення в фоновому режимі;
- 3) встановлена операційна система Android версії 2.2 і вище;
- 4) наявність Google-акаунта для пристроїв з операційною системою версії нижче Android 4.0.4.

Таким чином, для використання мобільного застосунку на стороні клієнта не пред'являються високі вимоги, що, безумовно, дозволяє популяризувати технологію.

Для організації повноцінної роботи система повинна включати в себе наступний мінімальний набір компонент: мобільний пристрій, сторонній сервер додатків для відправки даних через службу GCM, службу GCM - служба доставки повідомлень до додатка клієнта.

При роботі в системі використовуються наступні параметри доступу :

ідентифікатор відправника (Sender ID) - номер проекту, реєстрований розробником в сервісах Google. Використовується в момент реєстрації Android-дodatки для ідентифікації пристрою під час надсилання повідомлення;

ідентифікатор додатку (Application ID) - найменування додатку, яким адресовано повідомлення;

реєстраційний ідентифікатор (Registration ID) - номер, формований GCM сервером для Android-дodatки, яке дозволяє приймати повідомлення. Даний ідентифікатор використовується стороннім сервером додатків для розсилки даних;

акаунт користувача в системі Google (Google User Account); маркер авторизованого відправника (Sender Auth Token) - ключ, який використовується сервером додатків для доступу до служб Google.

Загальна архітектура push-повідомлень Google для мобільних пристроїв включає основні етапи процесу, які можна поділити на три основні групи: реєстрація, відправка і отримання повідомлення [3].

1. Реєстрація в GCM: в момент реєстрації активується реєстраційний інтент (`com.google.android.c2dm.intent.REGISTER`), який містить ідентифікатори відправника і додатки; у випадку успішної реєстрації GCM сервер повідомляє додатком реєстраційний ідентифікатор; для завершення реєстрації Android-застосунок відправляє реєстраційний ідентифікатор серверу завтосунка.

2. Відправлення та отримання повідомлення: сервер програми відправляє повідомлення GCM-сервера; Google ставить в чергу і зберігає повідомлення, коли пристрій вимкнений або недоступний; коли пристрій доступний, GCM служба перенаправляє повідомлення мобільному додатку; спеціальна служба операційної системи Android (notification service) ретранслює повідомлення цільовим застосункам.

Дана система призначена для запобігання лавиноподібного потоку повідомлень користувачеві і загальної оптимізації роботи мережі і батареї мобільного пристрою. Заснована вона на використанні обмеженого числа токенів для додатку, які витрачаються в міру надходження повідомлення. На кожне повідомлення система віднімає один токен. Токени в свою чергу поповнюються з певним часовим лагом. У разі вичерпання ліміту повідомлення переносяться в буферну чергу GCM служби, що може створювати тимчасові затримки у доставці повідомлень. В цілому можна відзначити, що використання даної архітектури дозволить створити досить просту і легко масштабовану систему повідомлень.

Push-технологія Apple (далі APN) - служба, розроблена компанією Apple, використовує постійно відкрите зашифроване IP з'єднання для пересилання повідомлень зі стороннього сервера Apple-пристроїв. Повідомлення можуть містити значки, звуки або текст. Сервіс був запущений в червні 2009р. При цьому кожне повідомлення обмежена розміром в 256В. Для перевірки справжності push-запитів з ІОС додатки Apple використовує цифрові сертифікати з відкритим ключем. Якщо повідомлення надходить на пристрій із зупиненим додатком, з'являється сигнал про надходження нових даних.

В цілому схема роботи push-технології Apple збігається з раніше розглянутої службою Google і включає в себе наступні етапи:

у момент активації програми через діалогове вікно в iOS-додатку система запитує дозвіл користувача на отримання повідомлень;

якщо дозвіл отримано, iOS-застосунок підключається до служби Apple Push Notification (APNs) за рядком унікального ідентифікатора. Для цього на пристрої встановлений додаток (його можна представити як аналог телефонного номера одержувача в традиційному сценарії обміну повідомленнями);

iOS-застосунок передає ідентифікатор на сервер-додатків;

у разі необхідності серверного додатком здійснити відправку push-повідомлення, APN перевіряє справжність push-сервера і використовує ідентифікатор для вказівки одержувача повідомлення;

якщо пристрій абонента знаходиться в режимі онлайн, він приймає і обробляє повідомлення. Якщо пристрій недоступний, повідомлення ставиться в чергу і доставляється, як тільки пристрій вийде на зв'язок.

При цьому система початкової аутентифікації пристроїв та сторонніх серверів влаштована складніше і здійснюється з використанням протоколу TLS. В даному випадку APN служба після встановлення з'єднання з нею спрямовує на мобільний пристрій сертифікат, який проходить перевірку, після чого сертифікат самого пристрою перенаправляється для перевірки в APN. У разі успішної верифікації встановлюється TSL з'єднання.

Кожне повідомлення, яке відправляється в службу APN, має супроводжуватися ідентифікатором пристрою, отриманим з клієнтського застосунку. Даний ідентифікатор розшифровується для оцінки валідності повідомлення та отримання ID пристрою призначення.

Архітектура системи push-технології BlackBerry включає в себе ініціатора push-повідомлень і пристрій BlackBerry з встановленим застосунком, який здатний приймати такі повідомлення (табл. 2) У даній технології передбачається, що ініціатор не чекає запиту на відправку контенту від одержувача. Максимальний розмір даних не повинен перевищувати 8Кб. Компанія BlackBerry стверджує, що повідомлення миттєво надходять на додатки користувача.

Архітектура є клієнт-серверним рішенням. Бібліотеки серверної і клієнтської частин взаємодіють з різними компонентами системи для забезпечення доставки контенту. Робота системи в цілому аналогічна роботі розглянутих раніше push-технологій і складається з наступних етапів [5]:

постачальник контенту (ініціатор) надсилає push-запит;

інфраструктура BlackBerry (PPG) повертає відповідь;

Основні компоненти push-технології BlackBerry

Компоненти	Опис
Push-ініціатор (Push Initiator)	Застосунок на серверній частині, яка створює повідомлення (запити і відповіді) з використанням бібліотеки серверної частини і передає на PPG
Серверна бібліотека (Server-side library)	Необхідне програмне API для взаємодії з PPG в форматі XML з боку сервера
Push проксі-шлюз (Push Proxy Gateway - PPG)	Процеси PPG проштовхують повідомлення, які надходять від ініціатора, а також формують код відповіді про доставку. Як PPG виступає інфраструктура компанії BlackBerry
Клієнтська бібліотека (Client-side library)	Необхідне програмне API для взаємодії з PPG з боку клієнта
Клієнтський push-застосунок (Push-енаблед application)	Застосунок на клієнтському пристрої BlackBerry, який здійснює підписку на сервері PPG, повідомлення надходить контенту в фоновому режимі

PPG проштовхує дані в мобільний пристрій BlackBerry;
 мобільний пристрій повертає відповідь PPG;
 відповідь-підтвердження надсилається ініціатору повідомлення;
 підтвердження про отримані дані прочитаного повідомлення повертається на сервер PPG.

Всі розглянуті технології надають можливість розробникам використовувати бібліотеки для побудови різних архітектур стороннього сервера застосунків для одночасної роботи з інфраструктурою Google, Apple, BlackBerry та інших.

Взаємодія зі службою APN може бути організовано за допомогою бібліотеки, яка зарекомендувала себе, а саме JPNS з відкритим вихідним кодом. Основні методи інтерфейсу даної бібліотеки являють собою статичні методи класу Push для повідомлень кожного типу і виконують перетворення повідомлення в формат JavaScript Object Notation (JSON), що приймається серверами APNs.

5. Створення системи для передачі push-повідомлень

Створення системи для передачі push-повідомлень передбачає насамперед визначення функціональних вимог. Для визначення функціональних вимог до системи була розроблена діаграма варіантів використання адміністратора клієнтської частини системи передачі push-повідомлень (рис.1). Основними варіантами використання адміністратора клієнтської частини є: авторизація, формування подій, управління підписками на події, визначення ролі користувача.

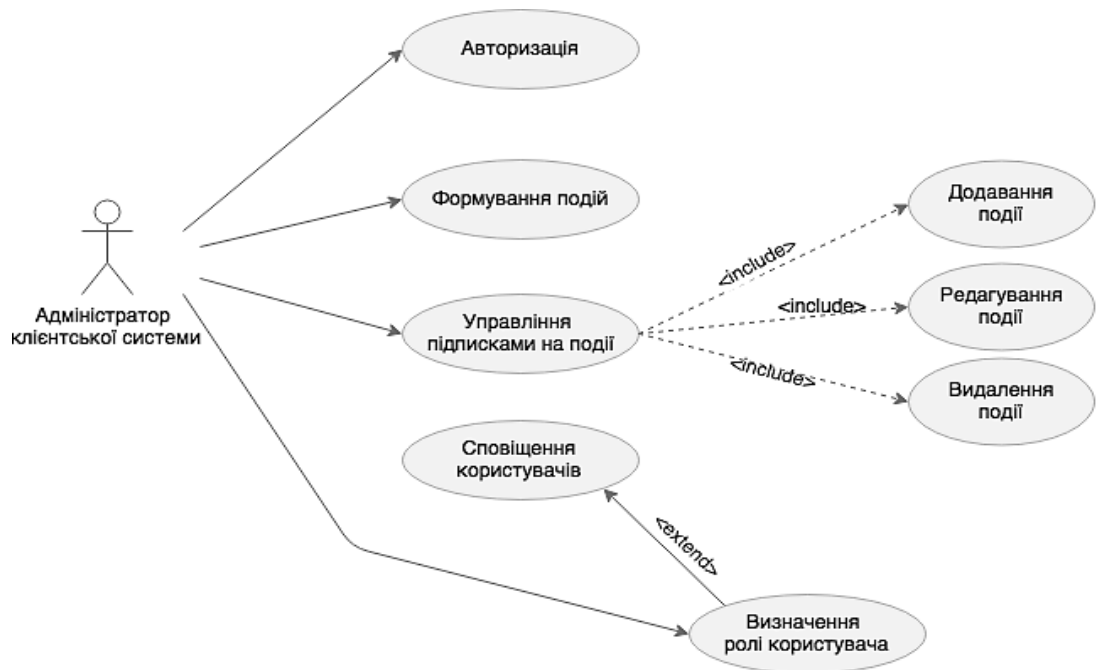


Рис. 1. Діаграма варіантів використання клієнтської частини системи передачі push-повідомлень

На рис. 2 представлена діаграма дерева функцій процесу сповіщення клієнтів, який включає: підписку на сповіщення, обробку інформації про користувача та доставку повідомлення



Рис.2. Діаграма «Дерево функцій» для процесу сповіщення клієнтів

Характеристика бізнес-процесу «Доставка сповіщень користувачам» представлена в табл. 3.

Таблиця 3

Характеристика бізнес-процесу «Доставка сповіщень користувачам»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Доставка сповіщень користувачам
Основні учасники	API, Web-клієнт, менеджер
Вхідна подія	Оформлення підписки на подію
Вхідна інформація	Підписка на подію
Вихідна подія	Сповіщення клієнту
Вихідні інформація	Візуальне сповіщення
Клієнт бізнес-процесу	Користувач

Оснoву передачі сповіщень користувачам складає система, яка дозволила б вирішити поставлені завдання, які можна розділити на три складові (рис. 3):

веб-сервіс, що надає користувачеві-публікатору взаємодіяти з системою, це особистий кабінет, що включає в себе як візуальний конструктор мобільного застосунку, так і консоль управління розсилкою;

шаблон мобільного додатка, що настраюється в візуальному конструкторі, а також система збирання застосунку з цього шаблону;

система доставки повідомлень. Потрібна система, яка дозволяла б здійснювати відправку повідомлень на конкретні зібрані додатки.

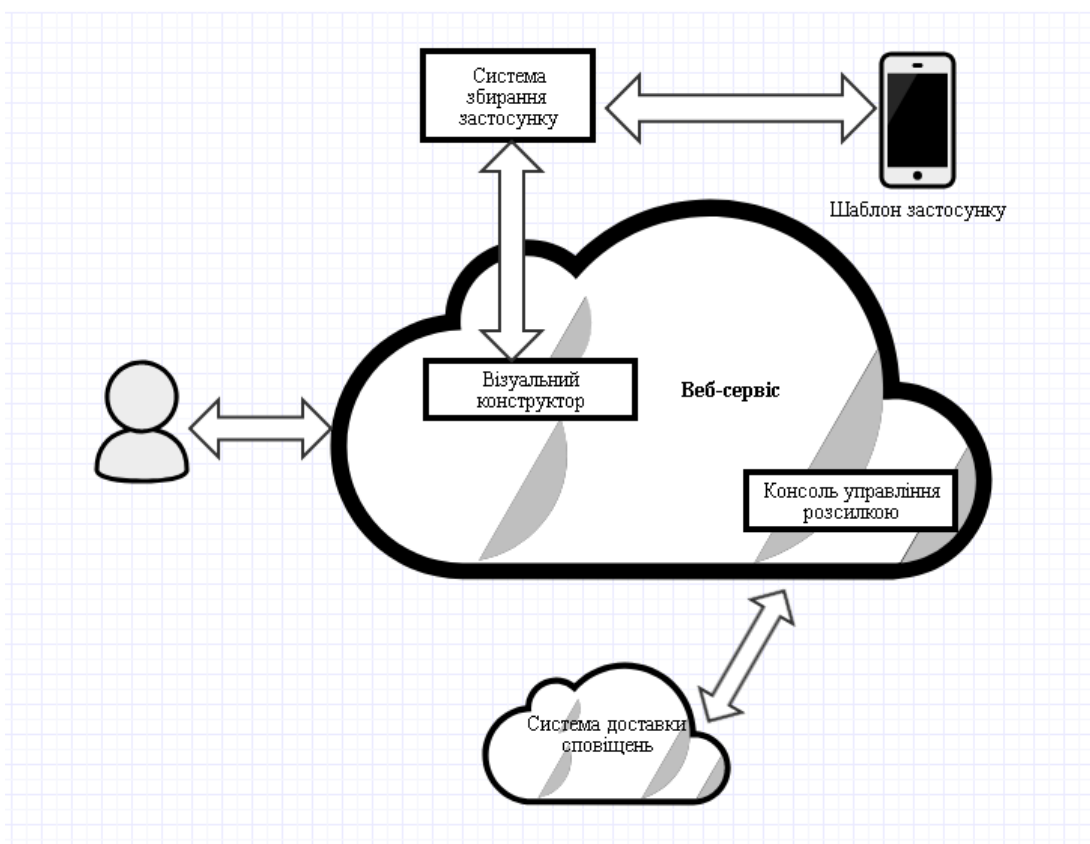


Рис. 3. Схема веб-сервісу системи доставки сповіщень

Учасник, який бажає стати публікатором розсилки, повинен зареєструватися, щоб отримати власний акаунт і увійти в особистий кабінет

Відносини між мобільними додатками, створюваними користувачем, і розсилками, які він буде публікувати, характеризуються зв'язками один-до-багатьох, тобто спочатку користувач створює сутність застосунку, яка буде поширюватися серед користувачів, а потім для цього застосунку він створює розсилки, на які цей застосунок буде підписано. У кожній розсилці може відправлятися не обмежене число повідомлень. Для того, щоб створений застосунок можна було публікувати у вільному доступі, він повинен бути підписаний унікальним ключем розробника. Ключі також можуть створюватися користувачем в особистому кабінеті і прив'язуватися до додатків зв'язком один ключ до багатьох додатків.

Стандартний алгоритм дій рядового користувача-публікатора після реєстрації можна розділити на дві фази: створення мобільного додатку для

поширення повідомлень між підписниками; створення та управління розсилкою. Після створення застосунку, користувач може здійснити в ньому розсилку.

Висновок

Широке поширення мобільних технологій дає можливість використовувати нові види доставки повідомлень клієнтам, до яких відносяться технології push-повідомлень. В роботі були розглянуті і проаналізовані найбільш популярні сервіси і платформи, що реалізують технології push-повідомлень. Проведений аналіз платформ для побудови системи push-повідомлень основних великих компаній-розробників операційних систем для мобільних пристроїв Google, Apple, BlackBerry показав, що пропонується ними архітектура легко масштабована, дозволяє використовувати сторонні сервери.

В роботі був запропонований підхід до створення системи передачі push-повідомлень на основі веб-сервісу, що надає користувачеві-публікатору можливість взаємодіяти з системою. Перевагою цього підходу є використання шаблону мобільного додатка, що настроюється в візуальному конструкторі, а також система для збирання застосунку на основі цього шаблону.

Література

1. Намиот Д. Е. Twitter как транспорт в информационных системах / Д. Е. Намиот // International Journal of Open Information Technologies. 2014. – С. 42-46.
2. Павлов, А. Д. Системы для поддержки push-уведомлений / А. Д. Павлов, Д. Е. Намиот // International Journal of Open Information Technologies. 2014. – С. 37-44.
3. Amazon SNS [Electronic resource]. – Access mode : <http://aws.amazon.com/sns/>
4. Android: Getting Started with Push [Electronic resource]. – Access mode : <http://docs.urbanairship.com/build/android.html>
5. Creating Our Web Push Service [Electronic resource]. – Access mode : <https://medium.com/the-guardian-mobile-innovation-lab/creating-our-web-push-service-91da44b38539>