УДК 519.6:004.93

к.т.н. **Козіна Ю.Ю.** (ОНПУ)
Авілов В.В. (ОНПУ)
Козін А.О. (ОНПУ)

СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ АНАЛІЗУ СУЧАСНИХ ІНСТРУМЕНТІВ ТА ПІДХОДІВ ДО РОЗРОБКИ КРОСПЛАТФОРМНИХ ВЕБ-ДОДАТКІВ

Існуючі методики створення кросплатформних додатків, як правило, вимагають велику кількість трудових ресурсів. Тому, для вирішення даної проблеми, в цій роботі, на основі аналізу новітніх методик та інструментів кросплатформної розробки,була запропонована архітектурна схема інформаційної системи. В основі її створення знаходится використання взаємодії двох безкоштовних фреймворків (Node.js і Titanium SDK). Для демонстрації переваг даного підходу була розроблена клієнт-серверна система програмного забезпечення, що працює на операційних системах android і ios, а так само на платформах, що підтримують веб-розробку. У результаті було досягнуто скорочення часу розробки.

Ключові слова: кросплатформні веб-додатки, клієнт-серверна архітектура, фреймворк.

В умовах бурхливого росту інформаційних технологій все більшу популярність і потрібність набувають додатки, що розробляються для використання їх на різних платформах – кросплатформені програми [1]. Це обумовлено появою та розвитком всіляких пристроїв, на яких вони можуть функціонувати. Одним з швидко розвинених напрямків в галузі багатоплатформеності в даний час є розробка додатків, що працюють на різних операційних системах, таких як десктопні - Windows, Mac OS, мобільні - iOS, Android. Однак, існуючі методики їх розробки орієнтовані на створення додатків під конкретну операційну систему, що обмежує універсальність їх застосування. Наприклад, додатки, розроблені під систему Windows не завжди будуть працювати на мобільних платформах. А з урахуванням зростаючого попиту на мобільні додатки, актуальність вирішення даного протиріччя зростає. Крім того, досягнення високої якості розробки кросплатформних додатків може бути забезпечено ефективною організацією процесу їх автоматизованого тестування.

Вимоги практики розробки кросплатформних додатків, що функціонують на десктопних і мобільних платформах, а також організація процесу їх автоматизованого тестування обґрунтовують актуальність даної роботи.

Постановка завдання. Метою роботи є розробка кросплатформного клієнт-серверного програмного забезпечення (Web сайт і додатки для Android і Ios), використовуючи зв'язку фреймворків для скорочення часу розробки.

Для досягнення поставленої мети вирішені наступні завдання:

– аналіз існуючих підходів розробки кросплатформного програмного забезпечення;

– аналіз основних фреймворків для розробки веб-додатків;

– огляд методологій розробки програмного забезпечення, що використовуються в роботі;

– проектування і розробка кросплатформної інформаційної системи «Calypso», що складається з веб і мобільних додатків;

– дослідження характеристик відсотка переносимості коду.

Розв'язання завдання. У теперішній час зростає потреба в додатках, які одночасно працюють на декількох платформах, а так само не рідкісні випадки, коли додаток пишеться під одну платформу і потім переноситься на іншу. Разом зі зростанням інтересу, зростає і кількість інструментів і підходів до кросплатформної розробки. Розглянемо ці підходи.

1) Нативний підхід. Перевагами є гарантія того, що додатки на цій платформі будуть працювати саме так як замислювалося тому, що додаток пишеться рідною мовою. Головним мінусом і основною причиною, чому цей підхід рідко використовується це, те що трудовитрати на розробку і розвиток будуть рости пропорційно кількості підтримуваних

платформ. Однак, якщо кросплатформність спочатку передбачити в архітектурі додатка, то ці витрати можна істотно скоротити.

2) Web підхід. Основною ідеєю цього підходу виступає написання веб-сайту з адаптивним дизайном. Плюсами даного підходу є проста реалізація, підтримка дуже великої кількості платформ (будь-який пристрій з web браузером). К мінусам даного підходу можна віднести: необхідне постійне інтернет підключення; не враховується специфіка роботи пристрою, що в свою чергу негативно впливає на швидкодію додатка, а так само на зручність роботи з ним; відсутність єдиних стандартів в браузерах [2].

3) Підхід використання кросплатформних мов програмування. Сильні сторони:

– підтримка більшості популярних платформ, також і мобільних;
– багато готових програмних бібліотек і готових рішень;
– потрібна мінімальна зміна коду для різних платформ (може бути взагалі не потрібна);

Слабкі сторони:

– відсутня підтримка Ios;
– як правило, відсутність нативних елементів управління. Це означає, що додаток буде мати однаковий вид на всіх платформах, і може виглядати незвично для користувача.

4) Підхід використання фреймворків для крос платформної розробки.

Фреймворк (англ. Framework - каркас, структура) - програмна платформа, яка визначає структуру програмної системи; програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту [3].

Ідея цих фреймворків полягає в тому, що весь код пишеться якоюсь популярною мовою, прописуються правила для конкретних платформ, якщо потрібно і потім фреймворк компілює або інтерпретує цей код в нативні програми під кожну обрану платформу. Так само дозволяє зберегти нативний зовнішній вигляд.

Загальні схеми існуючих інструментів та підходів до крос платформної розробки та основних фреймворків показані на рис. 1,2.

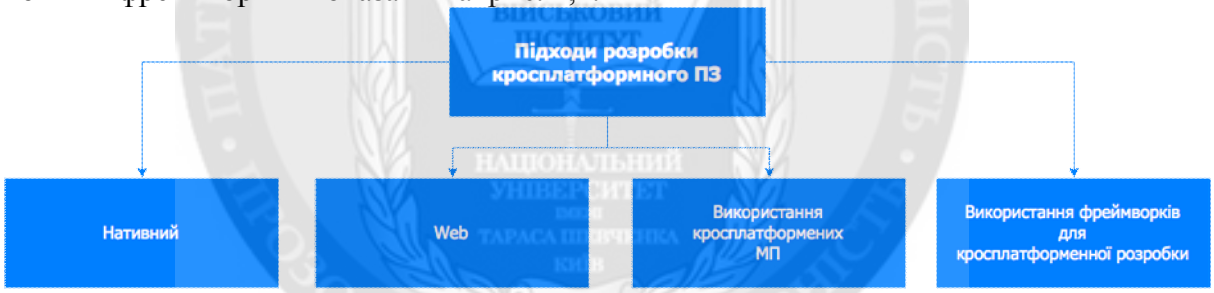

Рис. 1. Аналіз існуючих технологій розробки кросплатформного програмного забезпечення

Рис. 2. Аналіз основних фреймворків кросплатформної розробки

Проведено аналіз основних веб-фреймворків. Вони спрощують розробку і позбавляють від необхідності написання рутинного коду. Багато фреймворків спрощують доступ до баз даних, розробку інтерфейсу, і також зменшують дублювання коду. На рисунку 3 приведені найпопулярніші на 2015 рік веб-фреймворки і основні відповідні їм мови програмування.

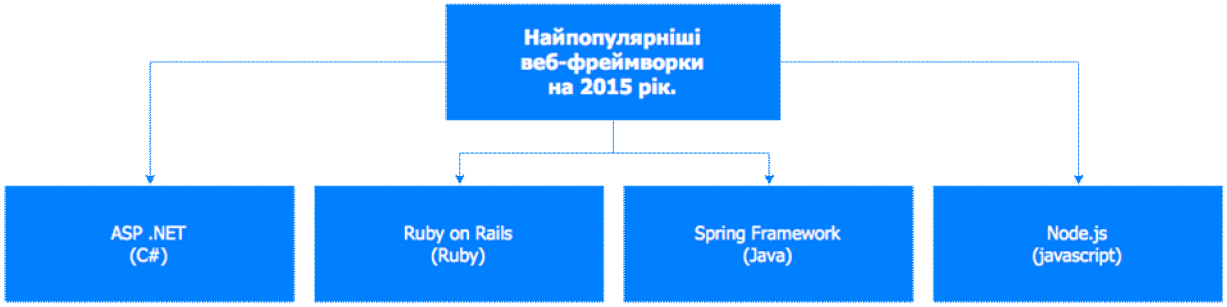

Рис. 3. Аналіз основних веб-фреймворків кросплатформної розробки

Основні критерії до обґрунтування вибору підходу до розробки, інструментів та методології програмування при реалізації кросплатформних додатків обрано такі:

– високий відсоток переносимості коду;
– продуктивність;
– безкоштовність продуктів;
– швидкість розробки.

На підставі аналізу підходів до кросплатформної розробки і на підставі критеріїв, оберемо підхід що використовує фреймворки.

Далі зробимо вибір з розглянутих фреймворків кросплатформної розробки, який так само базується на відповідності сильних і слабких сторін фреймворків та критеріях. Результатом вибору буде: Titanium SDK. Цей фреймворк повністю безкоштовний. Має високу переносимість коду, поступається тільки PhoneGap за цим параметром, проте виграє у нього в продуктивності. Швидкості розробки сприяє велика кількість готових рішень «з коробки». Однак якість продукту може постраждати через те, що використана мова javascript асинхронна та в численних вкладених розгалуженнях програмісту легко помилитися.

Так як нашим завданням є клієнт-серверне програмне забезпечення, то нам також потрібен інструментарій для створення серверної частини – веб-сервіс зі своїм API. Хоча Titanium SDK і дозволяє написання веб-додатків, але його функціонал призначений для самостійних додатків і написання веб-сервісу на ньому буде не доцільно. Результатом вибору буде – Node.js. Але залишається така ж уразливість з якістю кінцевого продукту із за асинхронності як і у випадку з Titanium SDK.

Для усунення вразливостей пов'язаних з помилками при написанні коду, будемо використовувати таку методологію програмування як BDD– розробка через поведінку (англ. Behavior-driven development, BDD) [4].

Проектована інформаційна система буде створена на основі клієнт-серверної архітектури. Роль сервера виконує додаток на Node.js, клієнти – додатки що виконуються на різних платформах (мобільні та web) написані на Titanium SDK.

Схема архітектури зображена на рисунку 4.

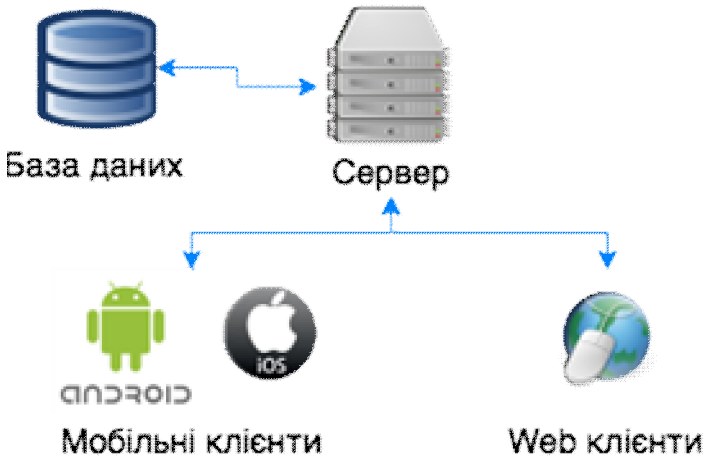

Рис. 4. Схема архітектури розроблюваної системи

Клієнти посилають запити на різні операції з об'єктом користувача або мікро повідомлень, сервер в свою чергу працює з базою даних і повертає потрібні дані клієнтам. Спілкування між клієнтами і сервером йде через протокол передачі гіпертексту HTTP (англ. HyperText Transfer Protocol).

Використовувані в розробляється системі методи HTTP:

– OPTIONS. Надсилається клієнтами для отримання списку дозволених методів;
– GET. Використовується для отримання запитуваного вмісту;
– POST. Застосовується для передачі даних до серверу;
– DELETE. Запит на видалення якихось даних.

Використовуваний формат даних для передачі в обидві сторони – JSON.

Розроблено серверну ті клієнтські частини інформаційної системи. Розглянемо основні функції клієнтської частини: надання графічного інтерфейсу користувачу для роботи з системою, а так само відправка запитів на сервер і обробка відповідей. Насамперед реалізуємо модель користувача на клієнтської частині, це необхідно для зручної синхронізації даних з сервером. Схема моделі буде повторювати схему моделі з сервера. Так само для роботи синхронізації потрібно написати так званий REST, всередині якого прописується правила відправки запити на такі дії з моделлю як:

– створити;
– видалити;
– прочитати.

Після того як синхронізація даних реалізована було створено графічний інтерфейс додатку.

У точці входу додатка пропишемо одне правило, якщо у внутрішньому сховище є поле з секретним ключем, то намагаємося отримати модель з сервера по ньому, якщо ключ не підходить або його немає, перенаправляємо користувача на вікно входу. У вікні входу повинні бути два текстових поля, одне для імені користувача і друге для пароля, і дві кнопки, перша «signin» – для входу, друга «signup» – для відкриття вікна реєстрації. Далі створимо вікно реєстрації, воно буде містити три текстових поля, для імені користувача, для пароля і для підтвердження пароля, дві кнопки «signup» і «signin» – для створення нового користувача і для відкриття вікна входу відповідно. Так само в цьому вікні перевіряється збіг паролів для того, щоб виключити помилку введення пароля користувачем. Якщо реєстрація пройшла успішно користувача аутентифікує і переадресує у вікно профілю. Останнім вікном нашого додатка буде вікно профілю користувача, воно містить текстове поле з ім'ям користувача і стрічку мікро повідомлень (рис. 5).

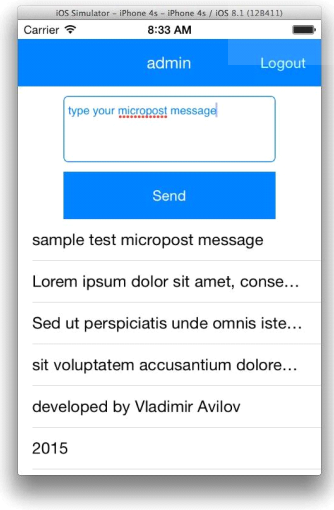

Рис. 5. Вікно профілю користувача

Висновки. У роботі виконано аналіз новітніх методик та інструментів кросплатформної розробки. Була запропонована архітектурна схема інформаційної системи, що створена на основі використання взаємодії двох безкоштовних фреймворків. Розроблена кросплатформна інформаційна система «Calypso», яка реалізує архітектуру «клієнт-сервер» та працює на платформах ios, android, і так само в будь-якому web браузері. При цьому отримані наступні результати: досягнуто відсоток переносимості коду в 94.7%, що є відмінним результатом. Якби ми писали код окремо під кожні платформи, то було б імовірно у 2,84 рази більше рядків коду. Тобто час, витрачений на розробку, за рахунок використання цієї зв'язки фреймворків та архітектури, було скорочено у 2,84 рази.

ЛІТЕРАТУРА:

1. Технології та засоби досягнення кросплатформності [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Багатоплатформність.

2. Дронов В.А. HTML 5, CSS3 и Web 2.0. Разработка современных Web-сайтов / В.А. Дронов . – СПб: БХВ-Петербург, 2011. – 416 с.

3. Денисюк А. В. Актуальные проблемы качества программного обеспечения /А. В. Денисюк, В. В. Любченко // Электротехнические и компьютерные системы. –К. : Техника. – 2013. – Вып. 09(95) – С. 142 – 148.

4. Introducing bdd [Електронний ресурс] – Режим доступу: http://dannorth.net/introducing-bdd/ .

REFERENCE:

1. Tehnologii' ta zasoby dosjagnennja krosplatformnosti [Elektronnyj resurs] – Rezhym dostupu: https://uk.wikipedia.org/wiki/Багатоплатформність.

2. Dronov V.A. HTML 5, CSS3 i Web 2.0. Razrabotka sovremennyh Web-sajtov / V.A. Dronov . – SPb: BHV-Peterburg, 2011. – 416 s.

3. Denisjuk A. V. Aktual'nye problemy kachestva programmnogo obespechenija /A. V. Denisjuk, V. V. Ljubchenko // Jelektrotehnicheskie i komp'juternye sistemy. –K. : Tehnika. – 2013. – Vyp. 09(95) – S. 142 – 148.

4. Introducing bdd [Elektronnyj resurs] – Rezhym dostupu: http://dannorth.net/introducing-bdd/ .

Рецензент: д.т.н.. проф. Барабаш О.В., Державний університет телекомунікацій

к.т.н. Козина Ю.Ю., Авилов В.В., Козин А.О.

СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ НА ОСНОВЕ АНАЛИЗА СОВРЕМЕННЫХ ИНСТРУМЕНТОВ И ПОДХОДОВ К РАЗРАБОТКЕ КРОСПЛАТФОРМННЫХ ВЕБ-ПРИЛОЖЕНИЙ

Существующие методики создания кроссплатформенных приложений, как правило, требуют значительного количества трудовых ресурсов. Поэтому, для разрешения данной проблемы,в этой работе, на основе анализа новейших методик и инструментов кроссплатформенной разработки, была предложена архитектурная схема информациионной системы. В основе ее создания находится использование взаимодействия двух бесплатных фреймворков (Node.js і Titanium SDK). Для демонстрации преимуществ данного подхода была разработана клиент-серверная система программного обеспечения, которая работает на операционных системах android и ios, а также на платформах, которые поддерживают веб-разработку. В результате было достигнуто сокращение времени разработки.

Ключевые слова: кроссплатформенные веб-приложения, клиент-серверная архитектура, фреймворк.

Ph.D. Kozina Y.Y., Avilov V.V., Kozin A.O.

CREATION OF AN INFORMATION SYSTEM BY ANALYZING MODERN TOOLS AND APPROACHES TO THE DEVELOPMENT OF WEB APPLICATIONS KROSPLATFORMNNYH

Existing methods of creating cross-platform applications typically require large amounts of labor. Therefore, to solve this problem, in this thesis, on the basis of the analysis of new methodic and tools of cross-platform development, we offered the architectural scheme of information system. At the base of its creating is using of interaction of two free frameworks (Node.js and Titanium SDK). For demonstrating the advantages of this approach has been developed client-server software system that runs on operating systems android and ios, and on platforms supporting the WEB. As result was reducing of development time.

Keywords: cross-platform web-applications, client-server architecture, framework.