

Висновки. В даній роботі був розроблений мобільний додаток для зберігання та відображення інформації про розробників програмного забезпечення. Даний додаток дозволяє оптимізувати роботу HR-менеджерів під час пошуку та найму працівників.

Список літератури

1. Matt N. *iOS 10 Programming Fundamentals with Swift* / Neuburg Matt., 2016. – 620 с.
2. *Realm: Building Modern Swift Apps with Realm Database* / Marin Todorov, 2018

УДК 004.42

*Гнатюк М. А., студент 3 курсу спеціальності 122 «Комп'ютерні науки»
Антонов Ю. С., к.ф-м.н., доцент, доцент кафедри інформаційних технологій*

ПРОБЛЕМА ВИКОРИСТАННЯ МЕРЕЖЕВОГО ІНТЕРФЕЙСА BLUETOOTH ПРИ РОЗРОБЦІ ДОДАТКІВ ПІД ПЛАТФОРМУ ANDROID

Донецький національний університет імені Василя Стуса, м. Вінниця

На сьогоднішній день мобільні пристрої обладнані багатьма мережевими інтерфейсами передачі даних. Одним із таких інтерфейсів є Bluetooth [1]. Його реалізація в екосистемі Android не завжди є тривіальним завданням. Також у вільному доступі майже відсутня інформація з цього питання, а у тих випадках коли вона наводиться, то надаються лише неповні фрагменти, які не розкривають проблему в цілому.

У цій роботі наводиться короткий огляд основних компонентів [2] та способів їх використання для встановлення з'єднання та зчитування даних з інших мобільних телефонів, моніторів, медіа-програвачів, тощо.

Платформа Android надає розробнику багаті комунікаційні можливості [3]. Для роботи з Bluetooth до складу Android входить потужний API, що дозволяє легко проводити сканування навколишнього простору на предмет наявності готових до з'єднання пристроїв, передачу даних між пристроями та багато іншого.

Робота з Bluetooth складається з чотирьох етапів: установка налаштувань bluetooth адаптера, пошук доступних для з'єднання пристроїв, установка з'єднання, передача даних [5].

Bluetooth API розташовується в пакеті `android.bluetooth`. До його складу входить кілька класів:

1. *BluetoothAdapter* – відповідає за роботу з встановленим в телефоні Bluetooth модулем. Об'єкт цього класу є в будь-якій програмі, що використовує Bluetooth.
2. *BluetoothDevice* – клас, що асоціюється з віддаленим Bluetooth

пристроєм. Примірник цього класу використовується для з'єднання через *BluetoothSocket* або для запиту інформації про віддаленому пристрої (ім'я, адреса, клас, стан).

3. *BluetoothSocket* – інтерфейс, аналогічний TCP сокетам. Це точка з'єднання, що дозволяє обмінюватися даними з віддаленим пристроєм через *InputStream* і *OutputStream*.
4. *BluetoothServerSocket* – представляє відкритий сокет сервера, готовий до обробки вхідного запиту. Для того щоб з'єднати два Android пристрою, одне з них повинно відкрити сокет за допомогою цього класу. Коли віддалений пристрій відправить запит на з'єднання, *BluetoothServerSocket* поверне об'єкт *BluetoothSocket*.

За допомогою класу *BluetoothAdapter*, Ви можете знайти віддалений пристрій, запустивши сканування або надіславши запит список спарених пристроїв [4].

Щоб з'єднати два пристрої, ви повинні написати серверну і клієнтську частину коду. Одне з пристроїв має відкрити серверний сокет, а друге – форматувати з'єднання, використовуючи MAC адресу сервера.

Після успішного з'єднання, кожне з з'єднаних пристроїв має об'єкт *BluetoothSocket* за допомогою якого легко реалізувати передачу / прийом даних:

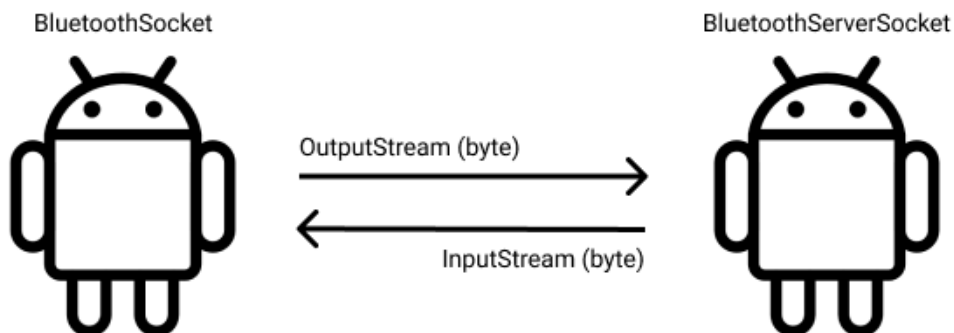


Рисунок 1 – Процес передачі даних

1. За допомогою методів *getInputStream()* і *getOutputStream()* об'єктів *InputStream* і *OutputStream*, ми маємо можливість передавати інформацію іншому пристрою у вигляді масиву байтів.

2. Читати і писати дані в потік за допомогою методів *read(byte [])* і *write(byte [])*.

Щоб закінчити роботу із підключеним пристроєм потрібно визвати метод *close()* об'єкта *BluetoothSocket*.

Отже, розглянувши основні компоненти роботи з мережевим інтерфейсом Android можна відокремити об'єкт *BluetoothSocket*, завдяки якому фактично відбувається вся передача інформації між пристроями. Також слід не забувати закривати *BluetoothSocket* для економії енергії.

Список літератури

1. *Bluetooth*. Wikipedia. URL: <https://ru.wikipedia.org/wiki/Bluetooth>

2. *Bluetooth Network Connection and Pairing. Electronics Notes. URL: <https://www.electronics-notes.com/articles/connectivity/bluetooth/network-pairing-connection.php>*
3. *Bluetooth Communication. Android Developers. URL: <https://developer.android.com/guide/topics/connectivity/bluetooth>*
4. *Bluetooth Basics. Sparkfun. URL: <https://learn.sparkfun.com/tutorials/bluetooth-basics/all>*
5. *How To Use Android BLE to Communicate with Bluetooth Devices. Medium. URL: https://medium.com/@shahar_avigezer/bluetooth-low-energy-on-android-22bc7310387a*

УДК 004.41:378.1

*Дідевич К. С., студентка 4-го курсу спеціальності 124 «Системний аналіз»
Мельников О. Ю., к. т. н., доцент, доцент кафедри інтелектуальних систем прийняття рішень*

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОБОТИ З ОСВІТНІМИ ПРОГРАМАМИ ТА СТАНДАРТАМИ ВИЩОЇ ОСВІТИ

Донбаська державна машинобудівна академія, м. Краматорськ

Стандарт вищої освіти – це сукупність норм, які встановлюють основну мету і задачі освіти, вимоги до змісту освіти, рівню підготовки фахівців, визначають спосіб діагностики якості вищої освіти. Кожен заклад вищої освіти на основі затвердженого стандарту для кожної спеціальності розробляє освітню (освітньо-професійну або освітньо-наукову) програму, яка може затверджуватися або змінюватися щорічно. Створення освітніх програм на основі існуючих стандартів, їх порівняння та вдосконалення є, звісно, творчим процесом, але необхідність постійного оновлення матеріалів та проведення оцінки якості вимагає залучення інформаційних систем та технологій [1].

Освітня програма містить загальні та спеціальні компетентності, програмні результати навчання, освітні компоненти (перелік дисциплін навчального плану), структурно-логічну схему та матриці відповідності компетентностей і програмних результатів навчання (ПРН) компонентам програми (тобто дисциплінам). Програмні результати навчання в освітній програмі повинні забезпечити, як правило, не менш 50% компетентностей СВО. Число компетентностей і ПРН може бути більше або рівним своїм аналогам зі СВО. Як правило, освітньо-професійна програма складається у закладі вищої освіти та може затверджуватися або змінюватися щорічно [2].

Було сформульовано задачу створення програмної системи, що дозволяла б працювати зі списком формованих компетенцій і по предметах, і програмних результатах навчання. Система повинна надавати можливість імпортувати всі наявні дані, вносити зміни до будь-якого розділу, при цьому працювати з даними XLS-формату.