УДК 004.42

О. С. Капличний
М. А. Очкуров

ПРОГРАМНИЙ ЗАСІБ ВЕДЕННЯ ОБЛІКУ ОБСЛУГОВУВАННЯ АВТОМОБІЛЯ ТА ЙОГО ТЕСТУВАННЯ

Вінницький національний технічний університет

Анотація
Розроблено програмний засіб ведення обліку обслуговування автомобіля та виконано його тестування.
Ключові слова: обслуговування автомобіля, дані автомобіля, Bluetooth, облік, тестування.

Abstract
A software tool for keeping track of car maintenance was developed and tested.
Keywords: car maintenance, car data, Bluetooth, accounting, testing

В наш час інтеграція мобільних пристроїв у всі сфери не є новинкою, кожна поважаюча себе компанія випускає свій додаток, в якому є певний функціонал для її клієнтів. Не є виключенням для поширення мобільних технологій і Інтернету й автомобільна індустрія. На даний момент у власників автомобілів є додатки, які дозволяють відслідковувати паркування машини, вмикати обігрів, заводити, провітрювати салон в спекотний день і багато чого іншого [1, 2]. Невід'ємною частиною експлуатації автомобіля є його обслуговування, саме від нього залежить можливість довгого користування та відсутність проблем у власника транспортного засобу. Тому було вирішено створити програмний засіб, який би автоматизував ведення обліку обслуговування автомобіля, створював нагадування про необхідність обслуговування, та міг би підказати найближчі станції технічного обслуговування (СТО), на яких можна провести чергове обслуговування транспортного засобу.

Загальна структура будь якого веб-проекту зазвичай майже однакова, кожен із таких проектів має головну сторінку, на якій є хедер, певний контент і футер [3]. Окрім головної сторінки завжди є інші сторінки, в яких також міститься певний контент, окремою частиною додатка є сховище та сторінка помилок, щоб показувати її у випадку якогось непередбачуваного моменту. Такий підхід покладено у основу для розробленого додатку ведення обліку обслуговування автомобіля.

При вході і успішній авторизації користувачу буде показано екран додавання його автомобіля. На екрані будуть поля для введення основної інформації про його автомобіль: марка, модель, рік виробництва та тип палива.

Для внесення даних про автомобіль користувач буде потрапляти на головний екран. Тут розташована мінімальна інформація про автомобіль, щоб не навантажувати користувача зайвою інформацією, як це було зроблено в деяких аналогах. На головному екрані користувач буде бачити:

— марку та модель автомобіля;
— середній пробіг за поїздку;
— середній час поїздки;
— останній день діагностування;
— заставки у вигляді плиток із переходом до деталізації по певних параметрах, таких як журнал обслуговування, паливо, діагностика автомобіля, управління витратами (рис 1).

Знизу головного екрану розміщена панель із індикатором підключення плати, яка отримує інформацію від машини. В індикаторі є три поділки, які загораються при:

— додаванні автомобіля в додаток;
— підключення додатка до пристрою для зчитування даних через порт OBD2;
— отримання актуальних даних з машини.

Для реалізації функціоналу під'єднання додатку, що установлений на телефоні користувача, до пристрою зчитування інформації з автомобіля було використано сторонню бібліотеку.

При натисканні на кнопку під назвою «Діагностика автомобіля» відкривається сторінка із переходом на підключення до автомобіля. На цій сторінці прописана загальна інформація по кодах помилок і кнопка переходу до карти розміщення СТО. Внизу сторінки знаходиться велика кнопка «Діагноз», по якій відбувається власне саме підключення мобільного додатку до автомобіля.

При натисканні на кнопку «Ремонтна майстерня» відкривається Google карти із введеним в них словом «Сервіс» для миттєвого знаходження сервісних центрів. Далі користувач натискає на «Сервіс» і надається номер телефону щоб можна було подзвонити на обрану СТО.

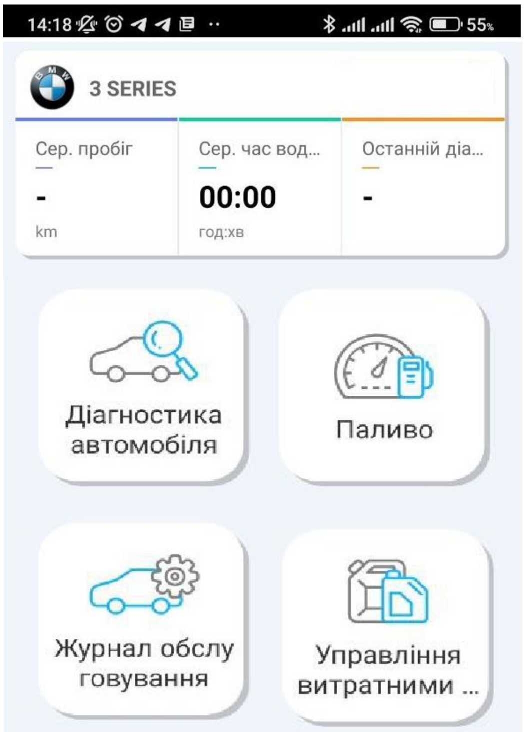

Рис.1— Фрагмент екрану головного вікна додатку

Оскільки робота з Bluetooth [4] в даному додатку є ключовим функціоналом, то для її використання була проведена деяка оптимізація. Так як технологія Bluetooth постійно працює у фоновому режимі, то була звернута увага на енергоефективність її використання, для зменшення використання енергії та надмірного розрядження телефону. Для цього була використана бібліотека react-native-ble-plx. Після налаштування даного пакету були прописані запити на дозволи користуванням Bluetooth на пристрої, на якому запускатиметься розроблене програмне забезпечення.

Очікуваними результатами по проведених роботах із налаштуванням роботи Bluetooth є стабільна і передбачувана робота додатка із даною технологією та уникнення помилок при непередбачуваних раніше обставинах. Встановленням та налаштуванням останнього пакету було оптимізовано енергоспоживання самого процесу роботи із безпровідною технологією, в результаті чого смартфон користувача буде набагато менше споживати енергії у фоновій роботі.

Тестування є важливим заключним етапом розробки програмного забезпечення, від цього етапу залежить наскільки якісний продукт надійде на ринок та наскільки першим користувачам сподобається даний продукт.

Для перевірки налаштування роботи Bluetooth потрібно було просто сканувати пристрої Bluetooth. Для цього додаємо швидкий одноразовий код до компонента головної сторінки. Даний код використано тільки для тестування та потім він був видалений перед кінцевою збіркою прототипу додатка. Для отримання класу менеджера Bluetooth необхідно під імпортованим повідомленням ввести const manager = new BleManager(). Таким чином була створена змінна, в яку за допомогою конструктору класу BleManager був присвоєний клас для роботи із налаштованим вище функціоналом. На сторінці головного екрану було додано наступний код:

```
const scanForPeripherals = () => {
manager.startDeviceScan(null, null, (error, scannedDevice) => {
```

```
console.log(scannedDevice)
})
}
```

Після чого було створено кнопку та у її властивість onPress внесено наступний код:

```
onPress={() => {
  dispatch(bluetoothPeripheralsFound(['AA:DD:CC:DD']));
  scanForPeripherals()
}}
```

Результат роботи по тестуванню додатку отримано після натискання на створену кнопку в терміналі розробника. На цьому етапі відбулось сканування пристроїв і в термінал розробника виведено результат сканування, де приводиться список пристроїв із їхніми даними, які ми можемо використовувати в своїх власних цілях.

При розробці даного програмного засобу були використані певні міри оптимізації по споживанню енергії смартфоном. Для перевірки ефективності даної доробки було проведено ряд тестів. Були проведені заміри, на скільки відсотків смартфон розряджається за ніч без запущеного додатка, який працює у фоні, із запущеною старою версією додатка та із запущеною оптимізованою версією додатка. Для чистоти експерименту телефон був заряджений до ста відсотків і відлік починали від конкретної години та закінчували конкретною годиною. Під час проведення дослідження встановлені програми та їх кількість не змінювались.

Сам смартфон протягом всього часу перебування в стані спокою не має вмикатись та повинен перебувати в “режимі польоту’’, оскільки неоднорідність сповіщень та кількість даних, які можуть знадобитись телефону за цей період, може бути не однакова. Також перед проведенням кожного із дослідів був увімкнений режим без обмежень системи. Для цього було перейдено в налаштування, обрано програму, з якої потрібно знати обмеження, вибрано налаштування «Зберігач батареї» та обрано пункт «Без обмежень».

Для створення програмного продукту було використано такі технології: TypeScript [5] та React Native [6] для написання фронтенду, Expo як основа створюваного додатку та система для налагодження розроблюваного продукту, Firebase як повноцінна платформа з власним API та базою даних, мова розмітки CSS та менеджер станів Redux. Було розроблено інтерфейс користувача, який адаптується під операційну систему та фізичні розміри екрану. Розробку здійснено у інтегрованому середовищі розробки WebStorm.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кращі програми для автомобіліста. [Електронний ресурс] — Режим доступу до ресурсу: http://autopark.pp.ua/515-krasch-programi-dlya-avtomoblsta.html.
2. ТОП-10 найкращих мобільних додатків для сучасних водіїв. [Електронний ресурс] — Режим доступу до ресурсу:https://enigma.ua/articles/top-10-naykrashchikh-dodatkiv
3. Пасічник О. Г. Основи веб-дизайну. / О. Г. Пасічник, О. В. Пасічник, І. В. Стеценко : [Навч. посіб.]. — К.: Вид. група BHV. — 2009. — 336 с.
4. Технологія Bluetooth [Електронний ресурс] — Режим доступу до ресурсу https://uk.wikipedia.org/wiki/Bluetooth.
5. TypeScript. [Електронний ресурс] — Режим доступу до ресурсу https://www.jetbrains.com/help/webstorm/typescript-support.html.
6. React Native. [Електронний ресурс] — Режим доступу до ресурсу: https://en.wikipedia.org/wiki/React_Native

Олег Сергійович Капличний — студент групи 1КІ-21м, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: olegkapl7@gmail.com.

Микола Андрійович Очкуров — старший викладач кафедри обчислювальної техніки, Вінницький національний технічний університет, м. Вінниця.

Oleg Kaplychnyi — students, Department of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: olegkapl7@gmail.com.

Mykola A. Ochkurov — Senior lecturer of the Computer Techniques Chair, Vinnytsia National Technical University, Vinnytsia.