

*Пузіно М. В.
Криворізький національний університет
Рибальченко О. Г.
старший викладач, Криворізький національний університет*

МОБІЛЬНИЙ КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС ПРОГРАМИ УПРАВЛІННЯ АВТОМАТИЧНОЇ ДРОН- ДОСТАВКИ

У роботі були виявлені особливості та виділені основні вимоги до мобільного користувачького інтерфейсу систем контролю автоматичної доставки дронами.

Через пандемію коронавірусу SARS-CoV-2 наразі усі масові заходи скасовуються задля попередження розповсюдження вірусу, у більшості країн світу введені карантинні обмеження. Більшість компаній перевели своїх співробітників на віддалену роботу, населенню рекомендовано залишатися вдома. Через це протягом останнього року очевидно зростає попит на послуги компаній, які спеціалізуються на послугах доставки як товарів та продуктів додому, так і на промисловій доставці. Промислова доставка являє собою доправлення багажу з одного пункту мережі доставки до іншого переважно без будь-якої участі людини (окрім моніторингу процесу системним адміністратором та можливим запитам з клієнтської частини системи). Це, по-перше, перешкоджає розповсюдженню вірусу, якщо розглядати цю проблему в реаліях сьогодення, та, по-друге, зменшує необхідність використання людської праці в цілому.

Саме через вищезазначену проблематику наразі у промисловій галузі в нашій країні є актуальним запровадження промислових дронів, які будуть використовуватися для швидкого перевезення продукції без додаткових витрат. Для їх керування не завжди є достатнім використання користувачького інтерфейсу, який надається за замовчуванням від виробника, особливо тоді, коли замовник такої доставки потребує більшого рівня захисту або ж додаткового функціоналу. Тож доволі часто необхідним є створення власних дронів з вузькоспеціалізованим мобільним інтерфейсом для конкретних заданих цілей.

Для визначення необхідних елементів інтерфейсу зробимо перелік основних вимог, які повинен задовольняти інтерфейс:

1. Інтуїтивність – інтерфейс має бути інтуїтивно зрозумілим для того, щоб з моменту початку використання додатку користувач міг з легкістю знайти необхідні меню або функції (навіть без попереднього досвіду користування);

2. Інформативність – інтерфейс повинен надавати достатню кількість інформації із мінімальними зусиллями з боку людини;

3. Лаконічність – інтерфейс не повинен бути перевантаженим для того, щоб користувач міг з легкістю виділити необхідну йому інформацію.

Знаючи вимоги до інтерфейсу, перелічимо основні функції, що він має задовольнити:

1. Мапа для виводу маршруту руху дрона, точок зльоту та посадки;

2. Виведення інформації про стан апарату (заряд батареї, швидкість руху, місцезнаходження, висота польоту тощо);

3. Пошук місць посадки за адресою;

4. Автоматична побудова маршруту польоту з пункту А в пункт Б з урахуванням висоти;

5. Авторизація користувача.

Крім того, бажано щоб створений додаток був багатоплатформовим, тобто міг бути скомпільований як для систем Android, так і для систем IOS (за вимогою замовника).

Саме тому ідеальним рішенням є використання Java Script фреймворку React Native [1]. Він не тільки дозволяє створити мобільний додаток для декількох платформ, але й надає можливість використовувати велику кількість допоміжних бібліотек для розробки інтерфейсу, наприклад react-native-maps [2] для створення мапи Google чи Mapbox; react-native-axios [3] для відправлення HTTP запитів (наприклад на API Google для побудови маршруту). Окрім бібліотек, рекомендовано використовувати популярну вільну об'єктно-реляційну систему управління базами даних PostgreSQL [4] при створенні, адже вона має наступні переваги:

1. Підтримка БД необмеженого розміру;

2. Потужні і надійні механізми транзакцій і реплікації;

3. Розширювана система вбудованих мов програмування і підтримка завантаження С-сумісних модулів;

4. Спадкування;
5. Легка розширюваність.

ВИСНОВКИ

Системи контролю дрон-доставки наразі являються затребуваним програмним продуктом у зв'язку з зростаючою популярністю автоматизованих екосистем перевезення вантажів. Саме тому перелічені особливості користувацького інтерфейсу подібних додатків можна вважати фундаментальними у даній галузі.

ЛІТЕРАТУРА

1. React Native. [Електронний ресурс] / Режим доступу: <https://reactnative.dev/>
2. react-native-maps. [Електронний ресурс] / Режим доступу: <https://www.npmjs.com/package/react-native-maps>
3. react-native-axios. [Електронний ресурс] / Режим доступу: <https://www.npmjs.com/package/react-native-axios>
4. PostgreSQL: The world's most advanced open source database. [Електронний ресурс] / Режим доступу: <https://www.postgresql.org>

Гребенюк Б. В.

Криворізький національний університет

Рибальченко О. Г.

старший викладач, Криворізький національний університет

КОНТЕЙНЕРИЗАЦІЯ. ВІРТУАЛЬНІ МАШИНИ ПРОТИ КОНТЕЙНЕРІВ

У доповіді було розглянуто питання запуску коду в ізольованому середовищі, проведено порівняння методів віртуалізації та контейнерізації, розглянутий інструмент контейнерізації Docker.

Інколи необхідно бути впевненим в безпечності деякого коду, в даному випадку поширеною практикою є запуск коду в ізольованому середовищі. Це робить неможливим «шкідливий» вплив програми на зовнішні процеси.