УДК 004.9:159.925:316.6

РОЗРОБЛЕННЯ МОБІЛЬНОГО ДОДАТКУ З ВИКОРИСТАННЯМ ПАТТЕРНУ MODEL-VIEW-CONTROLLER

Д.С. Галака, студент
Київський національний університет технологій та дизайну
В.Г. Резанова, кандидат технічних наук, доцент
Київський національний університет технологій та дизайну

Ключові слова: Model-View-Controller, програмне забезпечення, мобільний додаток, архітектура, кросплатформеність, Xamarin.

Мобільні додатки є невід'ємною частиною сучасного цифрового життя. З кожним днем вони стають все більш необхідними для розваг, навчання, роботи та спілкування. Мобільна розробка — це актуальна сфера розробки програмного забезпечення.

Кросплатформна мобільна програма – це програма, яка працює на кількох мобільних платформах (Android, iOS). Для спрощення розробки кросплатформенних мобільних додатків і, отже, для зменшення витрат на розробку та підтримку, було запропоновано декілька інфраструктур. Серед них крос-компіляторні фреймворки мобільної розробки, такі як Xamarin від Microsoft, що перетворюють код програми, написаний на проміжній мові, на рідний код для кожної бажаної платформи.

Поставлено задачу: використовуючи кросплатформений інструмент розробки Xamarin, створити мобільний додаток для проведення тестувань за програмою мульти-інтелекту різними мовами та перегляд їх результатів і подальший аналіз даних. Додаток має бути доступний на мобільному пристрої з операційною системою Android чи iOS. Мають бути виконані наступні функціональні вимоги: до додатку: - Користувачі мають мати можливість обирати мову і переглядати доступні тести.

- Додаток повинен надати можливість проходити обрані тести з підтримкою різних типів питань (з однією чи кількома відповідями, відкриті питання тощо).

- Додаток має підтримувати збереження прогресу проходження тесту, включаючи те, щоб користувач міг відновити тест після виходу з додатку.

- Після завершення тесту користувач повинен мати можливість переглядати свої результати.

- Додаток повинен надати детальний звіт про відповіді користувача на кожне питання та загальний результат.

- Відповіді користувачів мають зберігатися безпечно та конфіденційно.

- Додаток має забезпечити надійне збереження даних навіть у випадку втрати зв'язку або закриття додатка.

- Додаток повинен мати зручний та інтуїтивно зрозумілий інтерфейс для забезпечення зручності взаємодії з користувачем під час проходження

тестів.

Для організації архітектури додатку використвємо шаблон *Model-View-Controller (MVC)*:

- *Модель (Model):* Містить логіку обробки даних, включаючи збереження тестів і результатів проходження.

- *Вид (View):* Представляє інтерфейс користувача, який відображає тести і отримує відповіді від користувача.

- *Контролер (Controller):* Керує взаємодією між моделлю та видом, обробляє введення користувача і виконує логіку проходження тестів.

Такий поділ полегшує модульну розробку, дозволяючи ефективне повторне використання коду та паралельну розробку. У контексті мобільних додатків, зокрема на пристроях Android, *MVC* служить фундаментальною структурою для керування інтерфейсами користувача та взаємодією. Через обмежену обчислювальну потужність і апаратні ресурси мобільних пристроїв, широкомасштабні програми часто неможливо розгорнути безпосередньо на цих пристроях. Замість цього пропонується збалансована архітектура *MVC*, яка розділяє функціональність між клієнтською та серверною сторонами, оптимізуючи продуктивність і використовуючи можливості розпізнавання контексту мобільного пристрою. Таким чином, архітектура *MVC* є невід'ємною частиною розробки мобільних додатків, забезпечуючи структурований підхід до проєктування та покращуючи керованість додатків. Його збалансована реалізація особливо важлива в контексті мобільних пристроїв з обмеженими ресурсами, що забезпечує оптимальну продуктивність і взаємодію з користувачем.

Широке визнання шаблону *MVC* та його еволюція у відповідь на технологічний прогрес підкреслюють його незмінну актуальність у індустрії програмного забезпечення.

Список використаних джерел

1. Alessandro Del Sole Real-World Xamarin.Forms Succinctly // Syncfusion: 2021. - 132 p.

2. Williams M. Xamarin Blueprints // Packt Publishing, 2016. – 295 p.

3. Johnson Paul F. Cross-platform UI Development with Xamarin.Forms // Packt Publishing, 2015. – 330 p.

4. Ryan Davis From Xamarin Native to Xamarin.Forms: Reaping the Rewards without the Risk // CODE Focus Magazine : 2019 - Vol. 16 - Issue 1 - .NET Core 3.0.

5. D. Hermes Building Xamarin.Forms Mobile Apps Using XAML // APress, 2019. – 445 p

6. Bilgin C. Mobile Development with .NET: Build cross-platform mobile applications with Xamarin.Forms 5 and ASP.NET Core 5, 2nd Edition // Packt Publishing, 2021. – 572 p.