

РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ ОБЛІКУ ТА ОПЛАТИ КОМУНАЛЬНИХ ПОСЛУГ

Вінницький національний технічний університет

Анотація

У роботі розроблено мобільний програмний продукт для автоматизації обліку помешкань, збереження показників лічильників та розрахунку оплати комунальних послуг. Система дозволяє користувачеві керувати декількома об'єктами нерухомості, вносити показники та отримувати автоматичний розрахунок суми до сплати. Реалізація виконана на базі фреймворку Flutter із використанням локальної бази даних SQLite.

Ключові слова: інформаційні системи, мобільний застосунок, Flutter, SQLite, комунальні платежі, автоматизація обліку.

Abstract

The paper presents a mobile software product for automating property accounting, storing meter readings, and calculating utility payments. The system allows users to manage multiple real estate objects, enter readings, and receive automatic calculations of the amount due. The implementation is based on the Flutter framework using a local SQLite database.

Keywords: information systems, mobile application, Flutter, SQLite, utility payments, accounting automation.

Вступ

Цифрові рішення все активніше впроваджуються у повсякденне життя, і однією з важливих складових побуту є облік та своєчасна оплата комунальних послуг. Більшість користувачів стикаються з проблемою розрізненості сервісів та ризиком помилок при ручних розрахунках. Актуальність роботи зумовлена потребою у створенні єдиного зручного інструменту для автоматизації процесу контролю витрат на комунальні послуги.

Мета роботи - проектування та реалізація мобільного застосунку, що забезпечує швидкий доступ до інформації про особові рахунки, автономність збереження даних та автоматизацію обчислень.

Результати дослідження

В основу розробки покладено кросплатформенну технологію Flutter (мова Dart) [4], що дозволяє створювати додатки для Android та iOS з єдиною кодовою базою [1]. Для забезпечення автономної роботи системи без постійного підключення до мережі Інтернет використано локальну базу даних SQLite [2].

Архітектура застосунку побудована за модульним принципом і включає:

- Інтерфейс користувача (UI) - набір екранів для взаємодії з системою (головний екран, екран внесення показників, історія платежів).
- Моделі даних - опис структур об'єктів (Property, Meter, Payment).
- Рівень роботи з даними - модуль для виконання CRUD-операцій у базі даних.

Система реалізує повний цикл управління платежами: від додавання нового помешкання та реєстрації лічильників до формування підтвердження про оплату через сервіс сповіщень.

Процес внесення показників включає валідацію даних сервісом MeterService та автоматичне обчислення суми через PaymentService. Проведений аналіз показав, що такий підхід мінімізує кількість дій користувача та підвищує точність розрахунків [1].

Діаграма випадків використання.

На рисунку 1 представлена діаграма випадків використання мобільного застосунку для автоматизації обліку та оплати комунальних послуг. Вона відображає основні сценарії взаємодії користувачів із системою та розподіл функціональних можливостей між різними ролями.

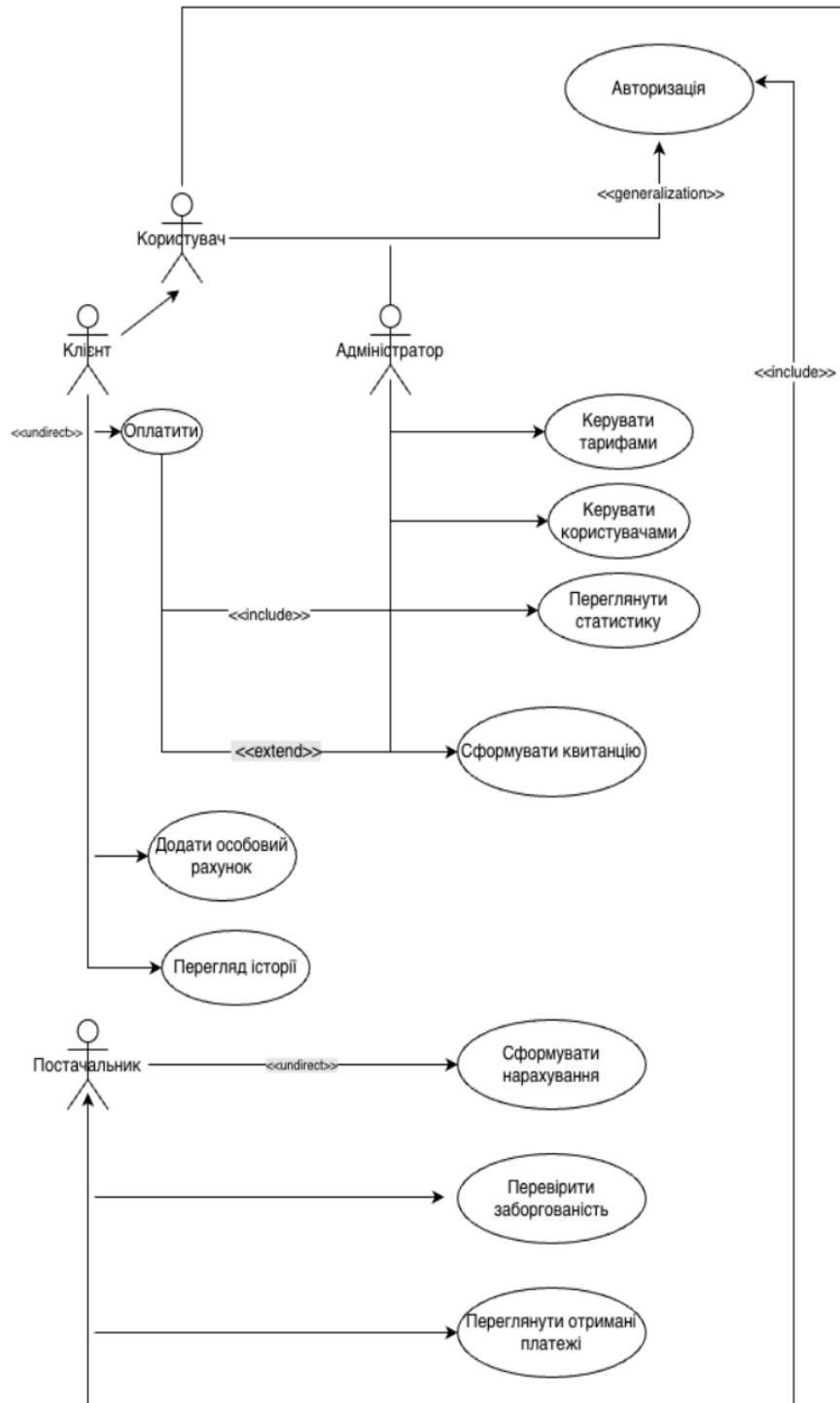


Рисунок 1 – Діаграма випадків використання (Use Case Diagram)

На рисунку 2 зображена діаграма послідовності процесу внесення показників лічильників, розрахунку платежу та підтвердження оплати. Вона демонструє взаємодію між користувачем та основними сервісами системи, такими як MeterService, DatabaseService, PaymentService та

NotificationService.

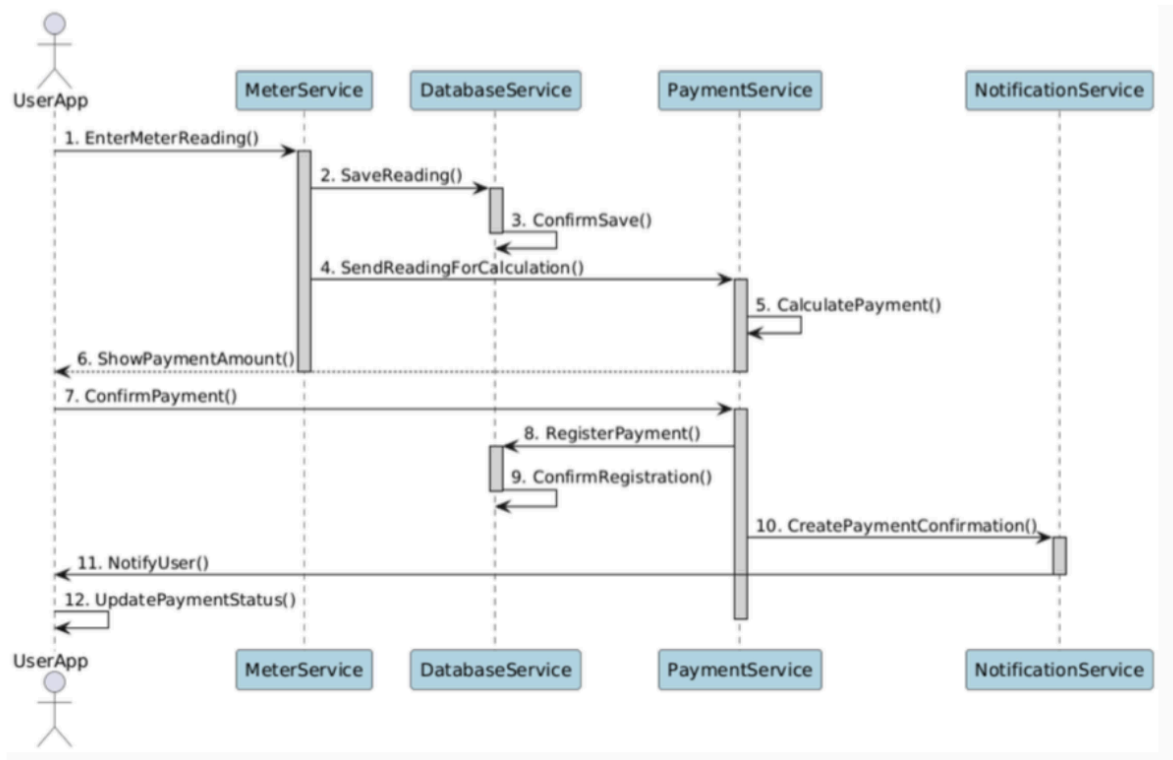


Рисунок 2 – Діаграма послідовності розрахунку платежу та підтвердження оплати

Опис процесу:

1. Введення показників: користувач через мобільний застосунок (UserApp) вводить нові показники лічильників (EnterMeterReading).
2. Збереження даних: дані передаються до сервісу MeterService, який ініціює їх збереження в базі даних через DatabaseService (SaveReading).
3. Підтвердження збереження: DatabaseService підтверджує успішне збереження даних (ConfirmSave).
4. Передача на розрахунок: MeterService передає введені показники до PaymentService для виконання розрахунку (SendReadingForCalculation).
5. Розрахунок платежу: PaymentService обчислює суму до сплати на основі тарифів та показників (CalculatePayment).
6. Відображення результату: результат повертається користувачу у вигляді суми до оплати (ShowPaymentAmount).
7. Підтвердження оплати: користувач підтверджує оплату (ConfirmPayment).
8. Реєстрація платежу: PaymentService передає інформацію про оплату до DatabaseService (RegisterPayment).
9. Підтвердження запису: DatabaseService підтверджує успішну реєстрацію платежу (ConfirmRegistration).
10. Формування підтвердження: PaymentService створює підтвердження платежу та передає його до NotificationService (CreatePaymentConfirmation).
11. Сповіднення користувача: NotificationService надсилає повідомлення користувачу про успішну оплату (NotifyUser).
12. Оновлення статусу: система оновлює статус платежу в інтерфейсі користувача (UpdatePaymentStatus).

На рисунку 3 представлена діаграма станів системи платежів, яка відображає життєвий цикл обробки комунальних нарахувань та оплати в мобільному застосунку.

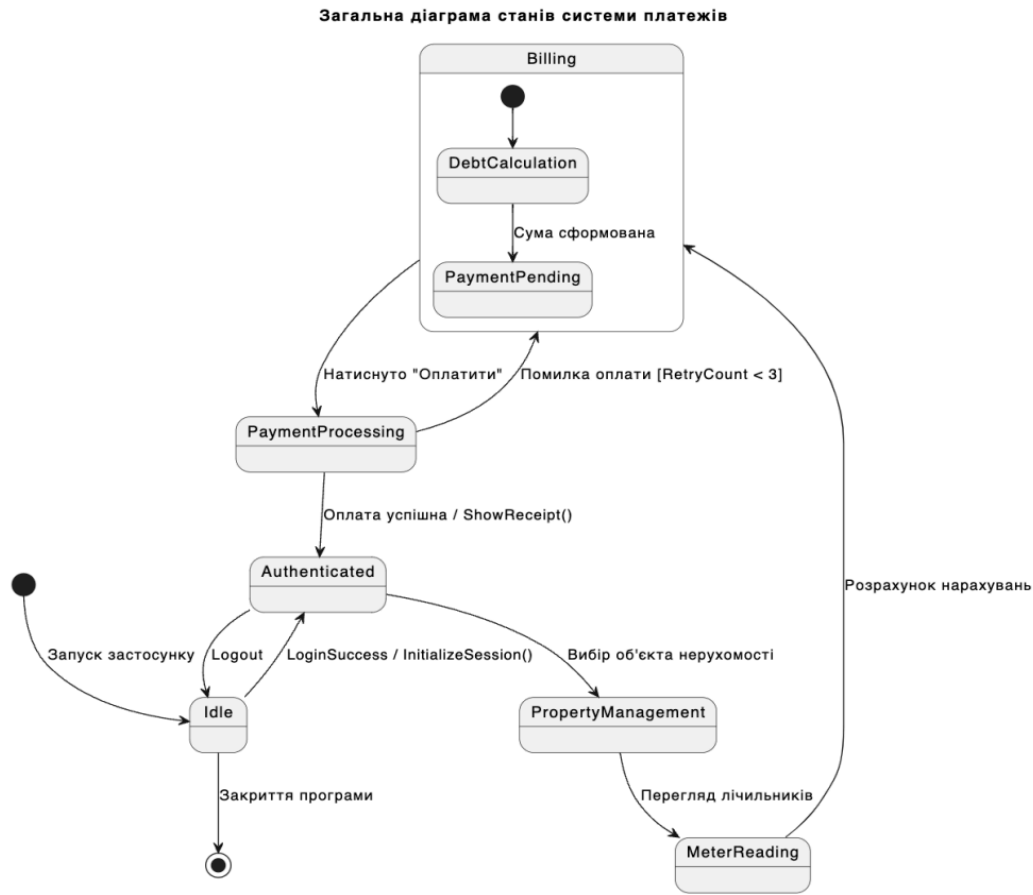


Рисунок 3 – Діаграма станів системи платежів

Діаграма демонструє переходи між різними станами системи залежно від дій користувача та результатів виконання операцій.

Основні стани системи:

1. Idle - початковий стан після запуску застосунку або перед авторизацією користувача.
2. Authenticated - стан після успішної авторизації користувача.
3. PropertyManagement - режим роботи з об'єктами нерухомості-сервіс для аналізу вподобань користувача на основі історії переглядів та взаємодій.
4. MeterReading - введення та перегляд показників лічильників.
5. Billing (вкладений процес):
 - DebtCalculation - розрахунок заборгованості;
 - PaymentPending - очікування підтвердження оплати.
6. PaymentProcessing - обробка платежу.

Висновки

Проведено системний аналіз процесу формування комунальних платежів. Побудовано UML-діаграми (Use Case, Sequence, State Machine), які дозволили чітко розмежувати бізнес-логіку програми на сервіси (MeterService, PaymentService) та забезпечити модульність архітектури.

Розроблено мобільний застосунок з використанням сучасного стеку технологій (Flutter/Dart). Обраний інструментарій дозволив суттєво скоротити час розробки за рахунок єдиної кодової бази для різних мобільних операційних систем.

Впровадження локальної СКБД SQLite забезпечило принципи ACID для транзакцій користувача на рівні пристрою. Це гарантує доступність даних про тарифи та історію платежів незалежно від якості Інтернет-з'єднання.

Створений прототип є масштабованим. Наступним етапом розвитку інформаційної системи є імплементація REST API для автоматичної синхронізації показників із серверами компаній-постачальників та інтеграція платіжних шлюзів (Google Pay / Apple Pay). [2, 3].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Flutter documentation. *Flutter documentation*. URL: <https://docs.flutter.dev> (date of access: 14.03.2026).
2. SQLite Documentation. *SQLite Home Page*. URL: <https://www.sqlite.org/docs.html> (date of access: 14.03.2026).
3. Sqflite | Flutter package. *Dart packages*. URL: <https://pub.dev/packages/sqflite> (date of access: 14.03.2026).
4. Dart programming language. *Dart programming language*. URL: <https://dart.dev> (date of access: 14.03.2026).

Губайдулліна Алсу Фанілівна — студентка групи ІСТ-22б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: alsuhubaidullinaa081819@gmail.com.

Жуков Сергій Олександрович — кандидат технічних наук, доцент кафедри системного аналізу та інформаційних технологій, Вінниця, e-mail: sazhukov@gmail.com.

Hubaidullina Alsu F. — Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: alsuhubaidullinaa081819@gmail.com.

Zhukov Serhii O. — Cand. Sc. (Eng.), Assistant Professor of the Department of Systems Analysis and Information Technologies, Vinnytsia, e-mail: sazhukov@gmail.com.