Специфікація ПЗ

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук

Кафедра програмної інженерії

СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

програмної системи для керування платними парковками



Студент гр. ПЗПІ-21-10\_\_\_\_\_\_\_\_\_\_\_\_\_ Радько М. М.

Харків 2025

**ЗМІСТ**

[1 Вступ 3](#_Toc200362028)

[1.1 Огляд продукту 3](#_Toc200362029)

[1.2 Мета 3](#_Toc200362030)

[1.3 Межі 4](#_Toc200362031)

[1.4 Посилання 5](#_Toc200362032)

[1.5 Означення та абревіатури 6](#_Toc200362033)

[2 Загальний опис 8](#_Toc200362034)

[2.1 Перспективи продукту 8](#_Toc200362035)

[3 Конкретні вимоги 16](#_Toc200362036)

[3.1 Вимоги до зовнішніх інтерфейсів 16](#_Toc200362037)

[3.1.1 Інтерфейс користувача 16](#_Toc200362038)

[3.1.2 Програмний інтерфейс 17](#_Toc200362039)

[3.1.3 Комунікаційний протокол 17](#_Toc200362040)

[3.1.4 Обмеження пам’яті 18](#_Toc200362041)

[3.1.5 Операції 19](#_Toc200362042)

[3.1.6 Функції продукту 20](#_Toc200362043)

[3.1.7 Припущення й залежності 22](#_Toc200362044)

[3.2 Атрибути програмного продукту 23](#_Toc200362045)

[3.2.1 Надійність 23](#_Toc200362046)

[3.2.2 Доступність 24](#_Toc200362047)

[3.2.3 Безпека 24](#_Toc200362048)

[3.2.4 Продуктивність 26](#_Toc200362049)

[3.2.5 Супроводжуваність 26](#_Toc200362050)

[3.2.6 Зручність використання 26](#_Toc200362051)

[3.3 Вимоги бази даних 27](#_Toc200362052)

**1 ВСТУП**

1.1 Огляд продукту

Програмне забезпечення, що розробляється, є комплексною системою для автоматизованого управління паркувальними майданчиками в умовах міської інфраструктури. Продукт спрямований на вирішення актуальних проблем, пов’язаних із дефіцитом паркомісць, неефективним використанням паркувального простору та відсутністю централізованого контролю. Система забезпечує облік, бронювання, моніторинг і оплату послуг паркування з використанням сучасних інформаційних та апаратних технологій.

Програмний продукт передбачає централізовану обробку даних про паркінги, транспортні засоби та користувачів, забезпечуючи безперервну взаємодію між усіма компонентами інфраструктури в реальному часі. Архітектура рішення охоплює серверну частину з бізнес-логікою, адміністративний веб-інтерфейс, мобільний клієнт для користувачів, а також IoT-пристрої для автоматичного обліку вільних місць і контролю доступу.

Система розробляється з використанням сучасного технологічного стеку, зокрема: PostgreSQL як СУБД, Node.js з Express для серверної частини, React для веб-інтерфейсу адміністратора, Kotlin для створення мобільного додатку, а також мікроконтролерів ESP32 з відповідними сенсорами для апаратної частини. Для реалізації функціоналу онлайн-оплати інтегровано підтримку популярних платіжних сервісів — PayPal і LiqPay, що дозволяє користувачам швидко й безпечно здійснювати оплату послуг безпосередньо через мобільний застосунок.

1.2 Мета

Метою створення програмного забезпечення є розробка інтегрованої інформаційної системи, призначеної для ефективного управління паркувальними майданчиками в умовах сучасної міської інфраструктури. Рішення передбачає централізоване зберігання та обробку даних, що стосуються об’єктів паркування, паркомісць, транспортних засобів, бронювань та зареєстрованих користувачів.

Система забезпечує взаємодію усіх складових паркувальної інфраструктури в режимі реального часу, що дозволяє здійснювати координацію процесів, моніторинг поточних станів та оптимізацію операційної діяльності. Програмний продукт включає серверну частину з реалізованою бізнес-логікою, адміністративну веб-панель для управління паркінгом, мобільний застосунок для кінцевих користувачів, а також апаратні засоби на базі IoT-пристроїв для автоматизованого обліку вільних місць та контролю доступу.

1.3 Межі

Програмне забезпечення має визначені межі функціональності та доступу, що залежать від ролей користувачів і специфіки реалізованої архітектури. Система передбачає поділ прав доступу між адміністраторами, операторами паркінгів та звичайними користувачами. Адміністративна частина надає розширені можливості для управління структурою паркінгів, конфігурацією обладнання, моніторингу активності та перегляду статистики, тоді як мобільний застосунок орієнтований на кінцевих користувачів і забезпечує лише функції пошуку, бронювання та оплати паркомісць.

Доступ до окремих функцій системи обмежується автентифікацією користувачів та перевіркою їхніх повноважень. Кожен запит до серверної частини супроводжується авторизаційною перевіркою, що унеможливлює виконання критично важливих операцій неавторизованими особами.

Система не передбачає використання у відсутності стабільного інтернет-з’єднання, оскільки обробка даних та взаємодія між компонентами здійснюється через хмарну інфраструктуру. Також не підтримується інтеграція з паркувальними майданчиками, які не обладнані відповідними IoT-пристроями або не відповідають встановленим технічним вимогам. Підтримка веб-інтерфейсу реалізована для сучасних браузерів, мобільний додаток — виключно для операційної системи Android.

Обробка платежів обмежена підтримуваними платіжними сервісами (PayPal та LiqPay), що також визначає регіональні обмеження щодо використання функцій онлайн-оплати.

1.4 Посилання

Під час розробки програмного забезпечення використовуються зовнішні та внутрішні джерела, що визначають технічні, нормативні та функціональні основи системи. Основу проектування складає технічне завдання, в якому сформульовані вимоги до функціональності, архітектури та призначення системи. В процесі реалізації враховано офіційну документацію до використаних технологій, зокрема для середовищ Node.js, Express, React, PostgreSQL, Kotlin, Android SDK, а також апаратної платформи ESP32 з відповідними сенсорами та модулями.

Інтеграція платіжних засобів реалізована на основі специфікацій та API-документації платіжних сервісів PayPal і LiqPay, що визначають формат запитів, вимоги до безпеки, обробку транзакцій та правила взаємодії з клієнтськими додатками. Крім того, при роботі з персональними даними враховано положення чинного законодавства України, зокрема Закону «Про захист персональних даних», а також міжнародні стандарти якості програмного забезпечення, такі як ISO/IEC 25010:2011, що регламентує вимоги до надійності, безпеки, зручності використання та підтримуваності систем.

Зазначені джерела слугують базою для прийняття технічних рішень, побудови архітектури системи, реалізації функціональних модулів, а також для оцінки відповідності програмного продукту сучасним вимогам і практикам у сфері розробки програмного забезпечення.

1.5 Означення та абревіатури

СИСТЕМА — програмно-апаратний комплекс, що включає серверну частину, веб-інтерфейс адміністратора, мобільний додаток користувача та IoT-пристрої для автоматизованого управління паркуванням.

КОРИСТУВАЧ — фізична особа, що взаємодіє із системою через мобільний додаток або веб-інтерфейс з метою отримання паркувальних послуг (звичайний користувач або адміністратор).

ПАРКОМІСЦЕ — одиниця паркувального простору, інформація про яку зберігається у системі, включає статус (вільне/зайняте), координати та належність до конкретного паркінгу.

ІОТ (IoT) — Internet of Things, інтернет речей; сукупність пристроїв, підключених до мережі, що здійснюють автоматичний облік паркомісць, контроль доступу та передають дані до системи.

API — Application Programming Interface, інтерфейс прикладного програмування; набір визначень і протоколів для взаємодії компонентів системи між собою та з зовнішніми сервісами.

БД (БАЗА ДАНИХ) — централізоване сховище структурованих даних, що використовується для зберігання інформації про користувачів, паркомісця, бронювання та інші сутності системи.

UI — User Interface, інтерфейс користувача; графічне або текстове середовище, через яке користувач взаємодіє із програмною системою.

HTTP — HyperText Transfer Protocol, протокол прикладного рівня для передавання гіпертексту, який використовується для комунікації між клієнтами та сервером.

ESP32 — мікроконтролер із підтримкою Wi-Fi та Bluetooth, що використовується для реалізації апаратної частини системи обліку паркомісць та контролю доступу.

PAYPAL — міжнародна система електронних платежів, яка використовується у складі мобільного додатку для здійснення безпечної оплати паркувальних послуг.

LIQPAY — українська платіжна система, що забезпечує електронну обробку транзакцій та інтегрується з мобільним застосунком як альтернативний спосіб оплати.

**2 ЗАГАЛЬНИЙ ОПИС**

2.1 Перспективи продукту

Розроблена програмна система управління паркувальними майданчиками має високий потенціал для подальшого розвитку та масштабування. З урахуванням сучасних тенденцій урбанізації, зростання кількості транспортних засобів та популяризації концепції «розумного міста», потреба в автоматизованих рішеннях для організації паркувального простору є надзвичайно актуальною. Запропоноване програмне забезпечення може бути адаптоване для використання в різних умовах — від муніципальних паркінгів до закритих стоянок житлових комплексів, торговельних центрів, аеропортів чи бізнес-центрів.

Завдяки модульній архітектурі система легко розширюється новими функціональними можливостями, зокрема інтеграцією з додатковими платіжними сервісами, впровадженням аналітичних інструментів для прогнозування завантаженості, додаванням підтримки відеонагляду або систем розпізнавання номерних знаків. Передбачена підтримка масштабування дозволяє використовувати систему як у межах одного об’єкта, так і на рівні міста чи мережі паркінгів різних операторів.

Інтерфейси та API системи відкривають можливості для інтеграції з іншими сервісами, зокрема навігаційними системами, мобільними додатками для транспорту, або муніципальними інформаційними платформами. Наявність мобільного додатку, веб-інтерфейсу та IoT-компонентів формує комплексне рішення, яке може бути комерціалізоване або використане в рамках соціально значущих ініціатив. У перспективі система може стати основою для створення повноцінної міської платформи паркування, орієнтованої на зручність користувачів, ефективне використання ресурсів та зниження навантаження на дорожню інфраструктуру.

2.2 Функції продукту

Програмна система управління паркувальними майданчиками реалізує широкий спектр функцій, які забезпечують ефективне, зручне та безпечне використання паркувального простору як для кінцевих користувачів, так і для адміністраторів. Однією з ключових функцій є реєстрація та аутентифікація користувачів із різними рівнями доступу. Це дозволяє розмежувати права та можливості між звичайними користувачами, які здійснюють бронювання паркомісць, та адміністраторами, які управляють роботою паркінгу, контролюють стан інфраструктури та проводять аналітику.

Для кінцевих користувачів система надає зручний мобільний додаток з інтуїтивно зрозумілим інтерфейсом. Користувачі можуть резервувати місця в реальному часі, отримувати підтвердження бронювання та переглядати історію своїх поїздок і платежів. Вбудовані повідомлення інформують користувачів про зміни статусу бронювання, наближення терміну оплати або інші важливі події.

Адміністративна панель системи надає розширені можливості управління паркувальними майданчиками. Адміністратори можуть додавати або змінювати інформацію про паркомісця, керувати бронюваннями, здійснювати моніторинг завантаженості майданчиків у режимі реального часу, отримувати звіти про використання ресурсів та фінансові операції. Система підтримує налаштування правил доступу та тарифів, що дозволяє гнучко адаптувати послуги під різні категорії користувачів і типи об’єктів.

Інтеграція з IoT-пристроями є важливою складовою системи. Завдяки використанню датчиків, мікроконтролерів ESP32 та інших апаратних рішень автоматично визначається статус паркомісць — зайняте чи вільне. Також реалізовано контроль доступу транспортних засобів за допомогою систем ідентифікації, що підвищує рівень безпеки та зменшує ризик несанкціонованого проникнення. Дані з IoT-пристроїв передаються у центральну систему в режимі реального часу, що дозволяє підтримувати актуальність інформації та швидко реагувати на зміни.

Щодо фінансової складової, система підтримує інтеграцію з платіжними сервісами PayPal та LiqPay, що забезпечує надійність, безпеку та зручність проведення платежів. Користувачі можуть обирати комфортний для себе спосіб оплати послуг паркування безпосередньо через мобільний додаток. Реалізовано механізми підтвердження платежів, обробки транзакцій, а також ведення історії фінансових операцій із можливістю формування звітності.

Захист персональних даних та інформаційна безпека є пріоритетними у функціоналі системи. Використовуються сучасні методи шифрування, аутентифікації та авторизації, забезпечується безпечне зберігання та передавання даних між клієнтськими додатками, сервером та IoT-пристроями. Система підтримує резервне копіювання інформації та відновлення у разі виникнення збоїв.

Додатково, система реалізує функції моніторингу та аналітики, що дозволяє адміністраторам отримувати детальну статистику щодо використання паркомісць, поведінки користувачів, доходів від послуг та ефективності роботи обладнання. Це сприяє прийняттю обґрунтованих управлінських рішень і подальшому вдосконаленню сервісу.

2.3 Характеристики користувачів

Користувачі програмної системи поділяються на кілька основних категорій, кожна з яких має свої особливості, рівень доступу та функціональні потреби. До кінцевих користувачів належать водії, які користуються паркувальними послугами. Вони мають змогу реєструватися в системі, шукати та бронювати паркомісця, здійснювати оплату через інтегровані платіжні сервіси, а також переглядати історію власних бронювань і платежів. Для зручності користувачів передбачено інтуїтивно зрозумілий мобільний додаток, що дозволяє виконувати всі необхідні операції у режимі реального часу.

Адміністратори паркувальних майданчиків виконують роль управлінців системою. Вони мають розширені права доступу, що дозволяють управляти інформацією про паркомісця, контролювати статус бронювань, моніторити завантаженість паркінгу, переглядати звіти та аналітику, а також налаштовувати параметри системи відповідно до потреб організації. Адміністратори також відповідають за підтримку працездатності IoT-пристроїв, які забезпечують автоматичний облік паркомісць і контроль доступу.

Користувачі з роллю технічного персоналу відповідають за обслуговування та підтримку апаратної частини системи, зокрема IoT-пристроїв, контролерів та мережевого обладнання. Вони мають доступ до спеціалізованих інструментів діагностики та оновлення системного програмного забезпечення.

Усі категорії користувачів взаємодіють із системою через відповідні інтерфейси — мобільний додаток для кінцевих користувачів, веб-інтерфейс для адміністраторів та технічного персоналу. Рівень доступу та функціональні можливості суворо регламентовані з метою забезпечення безпеки та ефективності роботи системи. Кожен користувач повинен проходити процедуру аутентифікації та отримувати відповідні права відповідно до своєї ролі.

2.4 Загальні обмеження

Програмна система Park4Flow має комплекс технічних, організаційних, правових та функціональних обмежень, які необхідно враховувати при її впровадженні, експлуатації та масштабуванні. Водночас існують певні виключення, що визначають сфери, в яких відповідальність постачальника програмного забезпечення обмежена. До технічних обмежень належить залежність системи від стабільного інтернет-з’єднання, оскільки без нього неможливі обробка платежів, синхронізація бронювань, надсилання push-сповіщень та отримання аналітики. У регіонах з нестабільним мобільним покриттям можливі затримки або втрата даних. Обмежена апаратна сумісність означає, що система підтримує роботу тільки з рекомендованими моделями камер розпізнавання номерних знаків, датчиків заповненості та шлагбаумів, використання стороннього обладнання вимагає додаткової адаптації або заміни.

Фінансові та операційні обмеження включають обмежений перелік платіжних провайдерів, зокрема інтеграцію лише з LiqPay та PayPal, що може ускладнити використання системи у юрисдикціях, де ці сервіси обмежені чи заборонені. Регулярні оновлення програмного забезпечення є обов’язковими для підтримки стабільності та безпеки, однак вони можуть призводити до тимчасової недоступності деяких сервісів. Також впровадження і обслуговування системи потребує фінансових затрат на придбання обладнання та абонентську плату, а ефективне використання можливе лише за умови навчання персоналу, відсутність якого знижує якість сервісу.

З правових обмежень слід враховувати суворе дотримання вимог до обробки персональних даних відповідно до GDPR, CCPA та локального законодавства, а також можливу необхідність отримання ліцензій чи погоджень для легального використання системи у певних регіонах.

Функціональні обмеження передбачають, що динамічне ціноутворення можливе лише за наявності відповідно налаштованих алгоритмів, а інтеграція з картографічними сервісами базується на Google Maps з обмеженою підтримкою інших платформ без додаткової розробки. Зі сфери виключень варто зазначити, що система не підтримує роботу офлайн — відсутність інтернет-з’єднання робить неможливим оплату, бронювання та отримання сповіщень. Постачальник не несе відповідальності за неполадки зовнішніх платіжних та картографічних сервісів, несумісність із несертифікованим обладнанням, а також обмеження щодо підтримки альтернативних методів оплати, таких як криптовалюти чи бартер.

Щодо безпеки, хоча система застосовує сучасні протоколи шифрування, вона не гарантує повний захист від усіх кіберзагроз, включаючи цільові атаки або витоки через сторонні пристрої. Відповідальність за помилки користувачів, наприклад, при введенні номерного знака, також несе користувач.

У разі змін законодавства адаптація системи може займати час та потребувати додаткових ресурсів, при цьому постачальник не відповідає за можливі втрати через затримки у відповідності. Крім того, система може тимчасово не працювати під час планових технічних перерв або обслуговування, про що користувачі попереджаються заздалегідь. У форс-мажорних ситуаціях, таких як природні катастрофи, масштабні кібератаки, відключення електропостачання чи банкрутство оператора паркінгу, стабільність роботи системи не гарантується.

2.5 Припущення й залежності

Для успішної розробки, впровадження та подальшої експлуатації програмної системи Park4Flow робляться певні припущення, які забезпечують належне функціонування та ефективність системи. По-перше, передбачається наявність стабільного та високошвидкісного інтернет-з’єднання на території паркінгів, що інтегрують систему. Це є необхідною умовою для безперебійної роботи функцій обробки оплат, синхронізації даних у реальному часі та взаємодії із хмарними сервісами. Крім того, оператори паркінгів повинні мати сучасне технічне обладнання, таке як комп’ютери, POS-термінали, планшети або мобільні пристрої з необхідними характеристиками, які забезпечують комфортну та повноцінну роботу з системою.

Важливою передумовою є також наявність базових навичок у користувачів системи. Очікується, що персонал паркінгів і водії, пройшовши навчання у вигляді відеоінструкцій, документації та інтерактивних тренінгів, володітимуть усіма необхідними знаннями для ефективного використання Park4Flow. Впровадження системи передбачає активну участь адміністрацій паркінгів, які повинні забезпечити доступ до навчальних матеріалів, організацію навчання персоналу, а також надавати зворотний зв’язок для подальшого вдосконалення системи.

Також до важливих припущень належить дотримання чинного законодавства. Система буде розроблена та адаптована відповідно до актуальних нормативно-правових вимог, зокрема в частині обробки персональних даних, захисту інформації та проведення електронних фінансових операцій. Перед запуском проекту передбачається отримання всіх необхідних ліцензій і дозволів, що дозволить використовувати Park4Flow у законодавчому полі конкретних країн чи регіонів.

Фінансування проекту має бути забезпечене повністю на всіх етапах життєвого циклу системи — від розробки до тестування, впровадження та подальшої технічної підтримки. До бюджету проекту також включаються витрати на навчання персоналу, супровід користувачів, оновлення програмного забезпечення та супутню документацію. Очікується, що кінцеві користувачі — оператори паркінгів і водії — будуть зацікавлені у застосуванні системи завдяки її зручному інтерфейсу, широкому функціоналу та можливості оптимізувати операційні витрати.

Крім того, важливою умовою є активна співпраця адміністрацій паркінгів з командою розробників для оперативного вирішення технічних питань, внесення необхідних змін і покращень функціональності системи.

Ефективність роботи Park4Flow залежить також від зовнішніх та внутрішніх факторів. Система повинна безперебійно інтегруватися з різноманітними зовнішніми сервісами, такими як платіжні шлюзи, картографічні API (наприклад, Google Maps), служби сповіщень та аналітичні інструменти. Надійність цих технологічних рішень є критичною, адже стабільність роботи хмарних платформ, банківських процесингових центрів і зовнішніх API безпосередньо впливає на функціонування системи.

Для підтримки актуальності, безпеки та відповідності ринку Park4Flow передбачає регулярне оновлення програмного забезпечення. Важливою складовою є також наявність кваліфікованої технічної підтримки користувачів, що включає в себе чат-боти, базу знань та підтримку у режимі реального часу, що сприяє стабільному та комфортному користуванню системою.

Система має відповідати чинним і майбутнім законодавчим вимогам у сфері обробки даних, кібербезпеки та електронної комерції. Її архітектура повинна бути гнучкою для оперативної адаптації до змін у законодавстві, безпекових стандартах або потребах ринку. Важливою складовою розвитку є збір і аналіз зворотного зв’язку від користувачів, що дозволить цілеспрямовано вдосконалювати інтерфейс, логіку роботи та функціональність системи.

Також Park4Flow повинна мати достатню продуктивність і масштабованість для обслуговування зростаючої кількості користувачів без втрати якості обслуговування. Ефективне просування системи на ринку забезпечується комплексними маркетинговими заходами, спрямованими на залучення нових користувачів. Для підтримки життєздатності продукту необхідна стабільна професійна команда розробників, здатна оперативно вирішувати технічні завдання, розвивати функціонал і забезпечувати відповідність системи сучасним вимогам.

**3 КОНКРЕТНІ ВИМОГИ**

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Користувацький інтерфейс реалізовано з використанням бібліотеки React, яка є однією з найпопулярніших технологій для розробки односторінкових додатків (SPA). Вибір React зумовлений високою швидкістю роботи, гнучкістю компонентного підходу та активною спільнотою розробників.

Інтерфейс спроектовано відповідно до принципів UX/UI-дизайну. Кожна функціональність, така як реєстрація, вхід, бронювання, перегляд паркомісць, історія транзакцій чи налаштування профілю, представлена у вигляді окремого компонента або модуля. Завдяки цьому досягається висока модульність, повторне використання коду та зручність підтримки.

Валідація користувацького вводу реалізована як на клієнтській стороні, так і на сервері для забезпечення гнучкого опису правил перевірки, зокрема перевірки обов’язковості полів, формату електронної пошти, мінімальної довжини пароля тощо. При виявленні помилки валідації користувач отримує зрозуміле повідомлення, яке дозволяє швидко виправити некоректні дані.

Крім того, передбачено обробку помилок при взаємодії з API. Якщо сервер повертає помилку (наприклад, неправильні облікові дані при вході або відсутність доступу до ресурсу), на інтерфейсі виводиться відповідне повідомлення з поясненням проблеми. У випадку втрати підключення до інтернету або недоступності сервера користувач також інформується через інтерфейс.

Уся навігація між сторінками здійснюється без повного перезавантаження завдяки використанню бібліотеки React Router, що забезпечує плавний користувацький досвід. Інтерфейс адаптований під мобільні пристрої за допомогою CSS Grid, що дозволяє комфортно користуватися додатком з будь-якого пристрою — смартфона, планшета чи настільного комп’ютера.

Загалом, інтерфейс користувача орієнтований на простоту, швидкодію та зручність використання, з акцентом на доступність та мінімізацію дій, необхідних для досягнення цілі.

3.1.2 Програмний інтерфейс

Програмний інтерфейс системи реалізовано у вигляді набору RESTful API-ендпоінтів, які надають функціональність для клієнтських додатків. Інтерфейс побудований відповідно до принципів REST, що забезпечує логічну організацію доступу до ресурсів та зручність розширення системи в майбутньому.

Кожен ресурс системи, такий як користувачі, транспортні засоби, паркомісця, підписки, платежі, бронювання тощо, має окремі маршрути, які обробляють відповідні HTTP-запити.

API розроблено з урахуванням принципів модульності та розмежування відповідальності: логіка обробки запитів винесена в окремі контролери, а робота з базою даних реалізована через рівень репозиторіїв. Це дозволяє легко тестувати, масштабувати та підтримувати систему.

На стороні клієнта (веб- або мобільного додатку) здійснюється виклик відповідних ендпоінтів для взаємодії з бекендом. Усі відповіді мають стандартну структуру: код стану HTTP, повідомлення про успішність або помилку, та, за необхідності, тіло з даними.

Особливу увагу приділено обробці помилок: API завжди повертає інформативні повідомлення у випадках неправильного запиту, відсутності прав доступу, помилок валідації або внутрішніх збоїв сервера.

3.1.3 Комунікаційний протокол

Для обміну даними між клієнтськими застосунками (веб-інтерфейсом, мобільним застосунком) і серверною частиною системи використовується комунікаційний протокол HTTPS, використання якого забезпечує шифрування переданих даних за допомогою SSL/TLS, що гарантує конфіденційність, цілісність та аутентичність.

Усі запити до REST API виконуються виключно через HTTPS-з’єднання за визначеним IP-адресом та портом (https://192.168.31.250:5192). Доступ до захищених ресурсів здійснюється тільки після проходження автентифікації.

Авторизація користувача реалізована за допомогою токенів формату JWT, які передаються в заголовку Authorization кожного запиту. Це дозволяє чітко контролювати доступ до функціональності в залежності від ролі користувача (адміністратор, звичайний користувач, тощо).

Запити та відповіді формуються у форматі JSON, що дозволяє легко обробляти дані як на фронтенді, так і в мобільному застосунку.

Таким чином, комунікація в системі є безпечною, надійною та відповідною сучасним стандартам передачі даних в Інтернеті.

3.1.4 Обмеження пам’яті

Система має працювати ефективно в умовах обмежених обсягів оперативної пам’яті, особливо на мобільних пристроях користувачів та серверному хостингу із фіксованими ресурсами.

Для клієнтської частини (веб та мобільного застосунку) обмеження пам’яті враховуються шляхом:

* використання оптимізованих компонентів інтерфейсу користувача (уникається зберігання великих об'єктів у стані);
* очищення зайвих даних з оперативної пам’яті після завершення дій;
* мінімального використання великих зображень та медіафайлів у додатку;
* обмеження кешу локального сховища (до 5 МБ для вебзастосунку).

Для серверної частини система має працювати стабільно на сервері з мінімальною конфігурацією:

* ОЗП: від 1 ГБ — з урахуванням обмеженого числа одночасних підключень;
* процесор: від 1 vCPU — достатньо для обробки запитів REST API без високонавантажених обчислень;
* всі тимчасові дані (у тому числі зображення та файли) зберігаються у файловій системі лише на час обробки, після чого автоматично видаляються;
* застосовується потокова обробка файлів без повного завантаження в пам’ять (наприклад, при завантаженні фото).

Крім того, при проектуванні моделі бази даних враховано обмеження на розмір записів, уникається зберігання великих JSON-структур або файлів безпосередньо в БД.

Таким чином, система залишається стабільною і продуктивною навіть в умовах обмежених апаратних ресурсів.

3.1.5 Операції

Зовнішні інтерфейси системи надають набір операцій, доступних через клієнтські застосунки (веб та мобільний), а також за допомогою програмного інтерфейсу (REST API). Ці операції реалізують ключову функціональність системи керування паркуванням та забезпечують взаємодію користувачів з усіма компонентами.

Основні групи операцій включають:

* Реєстрація та автентифікація користувачів – створення акаунтів, вхід у систему, перевірка прав доступу.
* Управління користувачами – оновлення профілю, блокування, скидання паролю.
* Робота з транспортними засобами – додавання, оновлення, видалення та перегляд власного автопарку.
* Управління тарифами та підписками – перегляд актуальних тарифів, створення підписок через PayPal або LiqPay.
* Бронювання місць для паркування – створення, перегляд, скасування та пропуск бронювань.
* Платежі та поповнення балансу – ініціалізація та підтвердження оплат різними платіжними сервісами.
* Виконання паркувальних дій – старт та зупинка паркування вручну або за допомогою IoT-модулів.
* Робота з політиками (знижки, скасування) – додавання, оновлення або видалення правил та політик.
* Пошук та аналітика – доступ до інформації про паркувальні місця, історію дій, улюблені паркінги, аналіз доступності.

Усі операції доступні через захищений канал зв’язку (HTTPS) та потребують належної автентифікації за допомогою токена доступу (JWT).

Інтерфейс розроблений таким чином, щоб користувач мав змогу інтуїтивно виконувати необхідні дії за кілька кроків. При цьому на рівні API підтримується модульність, логічна структурованість запитів та чітке розділення обов’язків між ендпоінтами.

3.1.6 Функції продукту

Зовнішні інтерфейси програмної системи повинні забезпечувати доступ до всіх основних функцій продукту згідно з їх призначенням. Вимоги до реалізації цих функцій через інтерфейси охоплюють як функціональність для кінцевих користувачів, так і для адміністраторів системи.

Для кінцевих користувачів через мобільний додаток та веб-інтерфейс мають бути доступні такі функції:

* Реєстрація, автентифікація, авторизація користувача (через API із захищеною передачею даних).
* Перегляд доступних паркувальних майданчиків у режимі реального часу.
* Бронювання паркомісця та отримання підтвердження.
* Відображення стану активних бронювань і повідомлень про зміни.
* Проведення оплат за допомогою інтегрованих сервісів PayPal та LiqPay.
* Перегляд історії транзакцій, поїздок, бронювань.
* Отримання push- або системних повідомлень щодо статусу бронювання, нагадувань, завершення часу тощо.
* Зміна налаштувань профілю, видалення акаунту.

Для адміністратора через веб-інтерфейс доступні:

* Управління користувачами, їх ролями та правами доступу.
* Додавання, редагування або видалення паркувальних майданчиків та окремих місць.
* Перегляд аналітики використання паркінгів, завантаженості, фінансової статистики.
* Контроль платежів, знижок, тарифів та правил надання послуг.
* Моніторинг даних з IoT-пристроїв (ESP32, сенсори) щодо стану паркомісць.
* Реагування на нештатні ситуації, перегляд журналів подій.
* Експорт звітів по бронюваннях, доходах, роботі обладнання.

Вимоги до реалізації:

* Усі функції повинні бути доступні через стандартизовані REST API з чітким описом параметрів і відповідей.
* Мобільний застосунок і веб-інтерфейс мають бути реалізовані відповідно до принципів UX-дизайну для забезпечення зручності користувачів.
* Доступ до критичних функцій (фінанси, управління доступом, адміністрування) повинен бути захищений та обмежений роллю користувача.
* Обробка помилок та повідомлення про них мають бути інтегровані як у UI, так і в API (через відповідні статус-коди та повідомлення).

Усі функції повинні працювати стабільно незалежно від платформи (мобільний додаток чи веб), а дані – синхронізуватися через бекенд у режимі реального часу.

3.1.7 Припущення й залежності

Успішна робота зовнішніх інтерфейсів системи Park4Flow (мобільного застосунку, веб-інтерфейсу, REST API та IoT-взаємодії) базується на низці припущень і залежностей, які необхідно враховувати під час розробки, впровадження та експлуатації системи.

Припущення:

* Припускається, що всі користувачі, які взаємодіють із зовнішніми інтерфейсами (водії, оператори, адміністратори), мають стабільне підключення до Інтернету, що забезпечує доступ до функцій у режимі реального часу.
* Очікується, що кінцеві користувачі мають сучасні пристрої (Android/iOS смартфони або комп’ютери з сучасними браузерами), які сумісні з клієнтськими додатками.
* Адміністрація паркінгів забезпечить використання сумісного обладнання для підключення до IoT-пристроїв та веб-інтерфейсу (наприклад, ПК з підтримкою HTTPS і стабільним з’єднанням).
* Користувачі та оператори пройдуть початкове навчання або ознайомлення з інтерфейсами системи (доступ через інтуїтивно зрозумілий UI, інструкції, довідка).
* Ключові функції зовнішніх інтерфейсів (реєстрація, оплата, бронювання, перегляд паркінгів) не потребуватимуть додаткових дій з боку користувача поза межами стандартного функціоналу.

Залежності:

* Зовнішні інтерфейси напряму залежать від стабільної роботи сторонніх API (Google Maps, платіжні шлюзи LiqPay/PayPal, служби аутентифікації), які забезпечують ключову функціональність.
* Безперебійна робота REST API для мобільного додатку, веб-інтерфейсу та IoT-пристроїв залежить від хмарної інфраструктури (сервери, CDN, бази даних).
* Усі зовнішні інтерфейси повинні регулярно оновлюватися згідно з новими вимогами безпеки, змінами в протоколах, оновленнями платформ (Android/iOS/браузери).
* Сумісність мобільного додатку з останніми версіями ОС Android та iOS залежить від дотримання вимог відповідних маркетів (Play Market, App Store).
* Інтерфейси IoT-компонентів (ESP32 тощо) залежать від стабільності передачі даних через Wi-Fi та коректної обробки пакетів REST/WebSocket-запитів.
* Взаємодія між клієнтськими інтерфейсами та бекендом має бути захищена сучасними методами шифрування (HTTPS, JWT, OAuth).
* Надійна технічна підтримка необхідна для вирішення проблем із доступом або помилками в роботі інтерфейсів у режимі реального часу.
* Масштабування кількості запитів до API та інтерфейсів прямо залежить від продуктивності серверної частини системи та кешування запитів.

3.2 Атрибути програмного продукту

Програмний продукт Park4Flow створюється для надійного, зручного та безпечного управління паркувальними майданчиками. Для успішного функціонування та задоволення потреб користувачів система повинна відповідати ряду ключових властивостей, які визначають якість та ефективність продукту.

3.2.1 Надійність

Надійність системи є критично важливою для забезпечення безперервної роботи сервісу, особливо у випадку управління паркувальними майданчиками, де від доступності системи залежить комфорт користувачів і дохід операторів. Park4Flow має бути здатним працювати без збоїв і швидко відновлюватися після можливих помилок.

* Система повинна забезпечувати стабільну роботу 24/7 без збоїв.
* Критичні зовнішні інтерфейси (API, мобільний додаток, веб-інтерфейс) повинні мати постійний доступ до мережі та електроенергії.
* Передбачено механізми обробки помилок і аварійне відновлення (наприклад, повторні запити, резервне копіювання даних).

3.2.2 Доступність

Доступність системи має важливе значення, адже користувачі повинні мати можливість отримувати сервіс у будь-який час, без обмежень за часом або місцем, особливо у випадках, коли паркування є критичною послугою.

* Система повинна забезпечувати високий рівень доступності (не менше 99.5%) і підтримувати роботу в режимі 24/7.
* Передбачена можливість роботи мобільного додатку в офлайн-режимі з подальшою синхронізацією даних.
* Серверна інфраструктура має бути резервована для мінімізації часу простою.

3.2.3 Безпека

Забезпечення безпеки є критично важливою вимогою до програмної системи Park4Flow, оскільки вона обробляє чутливу інформацію користувачів — включно з персональними даними, платіжними реквізитами та історією паркувань. Надійний захист цих даних є основою довіри користувачів до сервісу, а також необхідною умовою відповідності міжнародним стандартам і законодавству про захист персональної інформації.

У системі обробляються кілька категорій конфіденційних даних, зокрема ім’я, прізвище, контактна інформація, дані транспортного засобу, а також фінансова інформація — баланс рахунку та історія платежів. Дані банківських карток не зберігаються на сервері, а всі транзакції здійснюються через сертифіковані платіжні шлюзи з високим рівнем захисту. Також система працює з геолокаційними даними — координатами припаркованих авто, часом паркування та історією бронювань, — що потребує особливої уваги до приватності користувача.

Щоб запобігти витоку або несанкціонованому доступу, у Park4Flow реалізовано низку технічних і організаційних заходів. Усі дані шифруються — як у спокої (з використанням AES-256), так і під час передачі (через TLS 1.3). Щодня створюються резервні копії, які зберігаються у захищеному середовищі з обмеженим доступом. При роботі з користувачами застосовується двофакторна аутентифікація, а також політика доступу на основі ролей, яка дозволяє обмежити доступ до критичної інформації лише уповноваженим співробітникам. У разі підозрілої активності користувач отримує автоматичне сповіщення, що дозволяє швидко реагувати на потенційні інциденти безпеки.

Окрему увагу приділено відповідності нормативним вимогам. Park4Flow відповідає положенням GDPR: користувачі мають повний контроль над своїми персональними даними — вони можуть переглядати, редагувати або видаляти інформацію у власному кабінеті. Також система дотримується принципів прозорості та отримання згоди на обробку даних. З боку інфраструктури впроваджено стандарти інформаційної безпеки ISO/IEC 27001, які регулюють контроль доступу, захист на рівні серверів і управління ризиками.

Користувачі мають змогу в будь-який момент ознайомитися з актуальною політикою конфіденційності та умовами використання. При оновленні цих документів система повідомляє користувача автоматично. Усі запити на видалення або редагування персональних даних обробляються відповідно до чинного законодавства та внутрішньої політики обробки інформації.

Таким чином, система Park4Flow забезпечує комплексний підхід до захисту даних — як технічно, так і нормативно — і створює безпечне середовище для зберігання, обробки й використання персональної інформації користувачів.

3.2.4 Продуктивність

Для користувачів дуже важливо, щоб система реагувала швидко, особливо під час здійснення бронювання чи оплати. Висока продуктивність гарантує комфортний досвід та запобігає втраті клієнтів через затримки або зависання.

* Відгук зовнішніх інтерфейсів (UI/API) не повинен перевищувати 1–2 секунд при нормальному навантаженні.
* API повинно обробляти щонайменше 100 запитів на секунду без втрати стабільності.
* IoT-інтерфейси повинні передавати оновлення про статус паркомісць з інтервалом не більше 5 секунд.

3.2.5 Супроводжуваність

Легка підтримка та оновлення системи забезпечує довготривалу ефективність та швидку адаптацію до нових вимог ринку.

* Кодова база повинна мати зрозумілу структуру, логування, коментарі та відповідати принципам SOLID.
* Передбачено автоматизоване тестування (unit, integration, UI), CI/CD та документацію API (наприклад, Swagger/OpenAPI).
* Адміністративна панель має функціональність для оновлення тарифів, керування паркінгами та генерації звітів без потреби залучати розробників.

3.2.6 Зручність використання

Зручність і простота інтерфейсів є запорукою широкого прийняття системи серед користувачів різного рівня технічної підготовки. Park4Flow орієнтований на максимальне спрощення взаємодії з сервісом.

* Інтерфейси для водіїв (мобільний застосунок) повинні бути інтуїтивно зрозумілими, із мінімальною кількістю кроків до здійснення бронювання чи оплати.
* Для операторів передбачена панель адміністрування з фільтрами, аналітикою та візуалізацією даних.
* Передбачена локалізація UI мінімум на дві мови: українську та англійську.

3.3 Вимоги бази даних

Програмна система Park4Flow потребує надійної, масштабованої та безпечної бази даних для зберігання ключової інформації: користувацьких профілів, транспортних засобів, паркувальних сесій, транзакцій, геолокаційних координат, журналів подій та технічних логів. До бази даних висуваються кілька критично важливих вимог: підтримка транзакційності, забезпечення цілісності даних, швидка обробка запитів, можливість роботи з геоданими, масштабованість, безпечне зберігання конфіденційної інформації та підтримка резервного копіювання.

У контексті цих вимог оптимальним вибором є PostgreSQL — потужна об’єктно-реляційна система управління базами даних, що відповідає стандарту ACID і підтримує складні структури даних та індексацію. По-перше, PostgreSQL забезпечує повну транзакційну підтримку: будь-яка дія в системі, пов’язана з оплатою, бронюванням або зміною даних користувача, виконується у межах транзакції, що гарантує узгодженість навіть у разі помилки або збою.

Крім того, PostgreSQL підтримує зовнішні ключі, унікальні обмеження, перевірки значень (CHECK), що дозволяє реалізувати сувору модель цілісності даних — наприклад, бронювання не може існувати без відповідного автомобіля та користувача. Це особливо важливо для динамічної структури системи, де багато сутностей взаємопов’язані.

Ще однією критичною вимогою є робота з геолокаційними даними. PostgreSQL разом з розширенням PostGIS дозволяє зберігати координати, виконувати просторові запити (наприклад, пошук найближчих паркінгів) і оптимізувати ці операції за допомогою спеціалізованих індексів. Це значно підвищує продуктивність та точність пошуку паркувальних місць на основі місцезнаходження.

Безпека даних — ще один ключовий аспект. PostgreSQL підтримує шифрування з’єднань (SSL/TLS), налаштування доступу за ролями, розмежування прав користувачів та аудит змін. Це дозволяє реалізувати надійний контроль доступу до конфіденційної інформації та відповідати вимогам GDPR та ISO 27001.

У контексті масштабованості та надійності PostgreSQL забезпечує підтримку реплікації, резервного копіювання та кластеризації, що дозволяє системі зростати разом із кількістю користувачів без втрати продуктивності. Також система підтримує індексацію великих таблиць, розбиття (partitioning) та асинхронну реплікацію, що є критичним для збереження високої доступності даних.

Усі ці властивості роблять PostgreSQL ідеальним кандидатом для використання в системі Park4Flow, де важлива надійність, масштабованість, безпека та розширена підтримка географічних і фінансових даних.