Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра автоматизації та інтелектуальних інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«**Система контролю доступу до купе при використанні Bluetooth Low Energy на основі електронного залізничного квитка**»

Виконав: студент 2 курсу, групи 1АКІТ-22м, спеціальність 151– «Автоматизація та комп'ютерно-інтегровані технології».

_____ Вадим КОЦЮБНЯК

Керівник: к.т.н., доцент кафедри АІІТ

_____ Роман МАСЛІЙ

«__4__» ___грудня___ 2023 р.

Опонент: к.т.н., доцент кафедри КН

_____ Ігор АРСЕНЮК

«__8__» ___грудня___ 2023 р.

Допущено до захисту
Завідувач кафедри АІІТ

_____ д.т.н., проф. Олег БІСІКАЛО
«__11__» ___грудня___ 2023 р.

Вінниця ВНТУ – 2023 року

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Рівень вищої освіти другий (магістерський)
Галузь знань 15 – Автоматизація та приладобудування
(шифр і назва)
Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва спеціальності)
Освітня програма Інтелектуальні комп'ютерні системи
(назва освітньо-професійної програми)

ЗАТВЕРДЖУЮ
Завідувач кафедри АІІТ
д.т.н., проф. Олег БІСІКАЛО.
20 09 2023 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Коцюбняк Вадим Андрійович
(прізвище, ім'я, по батькові)

1. Тема роботи: «Система контролю доступу до купе при використанні Bluetooth Low Energy на основі електронного залізничного квитка»
Керівник роботи Маслій Р.В. к.т.н., доцент кафедри АІІТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ВНТУ від «18» 09 2023р. №247
2. Строк подання студентом роботи 05. 12 2023 р.
3. Вихідні дані до роботи:
- Напруга живлення – 5В.
- Робоча частота модулю приймача – 2,4 ГГц
- Дальність прийому сигналу – до 60 метрів .
4. Зміст текстової частини
Вступ. Дослідження предметної області. Обгрунтування методів реалізації. Реалізація системи контролю доступу. Тестування системи контролю доступу та розробленого програмного забезпечення. Економічний розділ. Висновки. Список використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
Макетна плата тестового зразку системи контролю доступу; Схема тестового зразку системи контролю доступу; Блок-схема роботи системи контролю доступу, Блок-схема роботи додатку, Рисунки екранів додатку.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	виконання прийняв
Спеціальна частина	к.т.н., доцент кафедри АІІТ Роман МАСЛІЙ	20.09.2023	06.12.2023
Економічна частина	к.е.н., доцент, Володимир КОЗЛОВСЬКИЙ	20.09.2023	06.12.2023

7. Дата видачі завдання 20 вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1.	Вибір, узгодження та затвердження теми МКР	20.09.2023	30.09.2023	Виконано
2.	Дослідження предметної області	01.10.2023	10.10.2023	Виконано
3.	Обгрунтування методів реалізації та контролю	11.10.2023	13.10.2023	Виконано
4.	Розробка структури та схеми	14.10.2023	25.10.2023	Виконано
5.	Розробка програмного забезпечення	26.10.2023	01.11.2023	Виконано
6.	Тестування та перевірка припущень	02.11.2023	08.11.2023	Виконано
7.	Підтвердження економічної частини	09.11.2023	12.11.2023	Виконано
8.	Оформлення матеріалів до захисту МКР	13.11.2023	03.12.2023	Виконано

Студент _____ Вадим КОЦЮБНЯК
(підпис) (прізвище та ініціали)

Керівник роботи _____ Роман МАСЛІЙ
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

УДК 004.67

Коцюбняк В.А. Система контролю доступу до купе при використанні Bluetooth Low Energy на основі електронного залізничного квитка. Магістерська кваліфікаційна робота зі спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології, освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2023. 109с.

На укр.мові. Бібліогр.: 30 назв; рис.:36; табл.: 15.

У даній магістерській кваліфікаційній роботі було розроблено система для автоматизованого контролю доступу до купе при використанні Bluetooth Low Energy на основі електронного залізничного квитка. В роботі проводиться ґрунтовний огляд існуючих пристроїв і методів розробки. Досліджено процес створення система контролю доступу. Встановлено оптимальні параметри для роботи системи з мобільним додатком.

Ключові слова: система контролю доступу, цифровий ключ, безпека, мобільна технологія, Bluetooth.

ABSTRACT

Kotsiubniak V.A. Access Control System for Train Compartments Using Bluetooth Low Energy Based on Electronic Railway Ticket. Master's Thesis in the field of 151 – Automation and Computer-Integrated Technologies, Educational Program – Intelligent Computer Systems. Vinnytsia: VNTU, 2023. 109 p.

In Ukrainian language. Bibliography: 30 titles; fig.: 36; tabl.: 15.

This master's thesis presents the development of a system for automated access control to train compartments using Bluetooth Low Energy based on an electronic railway ticket. The work provides a comprehensive review of existing devices and development methods. The process of creating an access control system is investigated, and optimal parameters for the system's operation with a mobile application are established.

Keywords: access control system, digital key, security, mobile technology, Bluetooth.

ЗМІСТ

ВСТУП

Актуальність. З ростом популярності мобільних технологій та збільшенням кількості подорожуючих, виникає необхідність у вдосконаленні системи використання квитків на потяги та полегшенні процесу подорожі. У цьому контексті система контролю доступу до купе при використанні Bluetooth Low Energy на основі електронного залізничного квитка, що знаходитиметься у мобільному додатку набуває особливої актуальності. Цей додаток може не лише спростити та прискорити процес використання квитків, але й покращити безпеку та комфорт подорожуючих, а також відповісти на сучасні вимоги з точки зору безконтактності та цифровізації. У даному контексті розглянемо актуальність та ключові переваги такої системи:

- Технологічний прорив: Використання BLE технології в додатках для мобільних пристроїв стає все більш актуальним. За допомогою Bluetooth можна забезпечити безконтактну комунікацію між смартфоном та різними електронними пристроями, включаючи ключі та замки.
- Зручність та безпека: Використання електронного Bluetooth ключа дозволяє подорожуючим легко та безпечно відкривати двері до купе. Це зменшує ризик втрати або крадіжки фізичних ключів.
- Безконтактність в епоху пандемії: Після отриманого досвіду в контексті пандемії COVID-19 безконтактні технології, такі як Bluetooth, набувають особливої важливості, оскільки вони знижують ризик передачі інфекції через фізичний контакт.
- Зручність пасажирів: Користувачі можуть отримати доступ до свого купе через мобільний додаток і електронний ключ, що робить процес зручнішим та швидшим.
- Інноваційні можливості: Розробка додатка з електронним Bluetooth ключем може стати основою для впровадження інших інноваційних функцій, таких як керування кондиціонером або освітленням в купе через смартфон.

- Залізничний транспорт: Залізничний транспорт залишається популярним видом перевезень в багатьох країнах, і розвиток цифрових рішень може поліпшити якість обслуговування пасажирів.

- Підвищення конкурентоспроможності: Залізничні компанії можуть використовувати такі інноваційні рішення для підвищення конкурентоспроможності та залучення нових клієнтів.

З урахуванням цих аспектів, розробка мобільного додатку для придбання квитків на потяг із використанням електронного Bluetooth ключа до купе є актуальною та перспективною темою для магістерської роботи. Такий додаток може внести значний внесок у покращення якості обслуговування пасажирів та стати інноваційним рішенням у галузі транспортних послуг.

Мета даної роботи полягає в розробці та дослідженні ефективної системи контролю доступу, який спрямований на оптимізацію процесу використання квитків для подорожей на потягах. Система має на меті забезпечити безпеку, зручність та ефективність для користувачів, зокрема за допомогою використання електронних Bluetooth ключів для доступу до купе та інших переваг сучасних технологій.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

– аналіз існуючих інструментів для реалізації автоматизованого тестування мобільних додатків та вибір оптимальних;

– аналіз існуючих систем контролю доступу для забезпечення максімальної надійності та вибір оптимального під існуючі потреби;

– розробка підходу до роботи з квитками, який дозволятиме ефективно інтегрувати цифровий ключ ;

– розробка архітектури яка забезпечить найшвидший і надійніший спосіб купівлі цифрових квитків з цифровим ключем;

– розробка програмного забезпечення для роботи з придбаними квитками на потяг з додатковою можливістю цифрового ключа до місця в купе для системи контролю доступу Bluetooth Low Energy;

– розробка схем яка забезпечить найкращі показники для системи контролю доступу до приміщення;

– розробка системи контролю доступу з керуванням на базі розробленої програми;

– дослідження швидкодії розробленого програмного забезпечення та аналіз ефективності автоматизованої системи купівлі квитка з цифровим ключем на основі отриманих результатів дослідження.

Об'єктом є процес запровадження системи для безпечного подорожування на потязі з використанням електронного Bluetooth ключа до купе.

Предметом дослідження – методи та засоби систем контролю доступу та використання Bluetooth ключів.

Методи дослідження. є створення і дослідження функціональності додатку а також системи контрою доступу, які оптимізують та спрощують подорожі на потягах для користувачів пристроїв. Робота спрямована на розробку інноваційного рішення, яке відповідає сучасним стандартам та вимогам цифрового транспорту, а також на поліпшення якості обслуговування пасажирів та підвищення зручності їхніх подорожей.

Новизна:

– Робота має важливий внесок у вирішення актуальних проблем ефективності, зручності та безпеки в громадському транспорті, особливо в умовах зростаючої мобільності та вимог до безконтактних послуг.

Практична цінність даної роботи полягає в тому, що на основі запропонованого підходу розроблено апаратно-програмне забезпечення, яке на відміну від існуючих дає змогу забезпечити пасажирів більш надійним захистом від всіляких злочинних методів завдяки цифровим ключам.

Апробація результатів та публікації. За результатами даної роботи опубліковано доповідь на науко-технічній конференції підрозділів Вінницького національного технічного університету Молодь в науці: дослідження, проблеми, перспективи (МН-2024) (м. Вінниця 2023).[1]

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Стан та тенденції громадської залізнотранспортної сфери

Акціонерне товариство «Українська залізниця»[2] – національний перевізник вантажів та пасажирів. Метою діяльності товариства є задоволення потреб у безпечних та якісних залізничних перевезеннях, забезпечення ефективного функціонування та розвитку залізничного транспорту, створення умов для підвищення конкурентоспроможності галузі тощо.

Свою господарську діяльність АТ «Укрзалізниця» розпочало 1 грудня 2015 року. Товариство є правонаступником усіх прав і обов'язків Державної адміністрації залізничного транспорту України, а також підвідомчих підприємств і закладів, що мали статус окремих юридичних осіб.

Наразі компанія забезпечує 82 % вантажних і майже 50 % пасажирських перевезень, які здійснюються усіма видами транспорту. За обсягами вантажних перевезень українська залізниця займає четверте місце на Євразійському континенті, поступаючись, зокрема, залізницям Китаю та Індії.

В таблиці 1.1 наведено порівняльну характеристику пасажірських перевезень з 2015-2018 рік.

Таблиця 1.1 – Дані з «Звіт по стратегічному дослідженню економічної ефективності та обсягів перевізної роботи по поїздо-дільницях регіональних філій АТ «Укрзалізниця» (на основі даних за 2015 - 2018 роки)»[3]

рік	2016	2017	2018
пасажирів, млн. осіб	162,3	158,1	151,1

1. Інфраструктура. Україна має одну з найбільших залізничних мереж у світі, яка включає в себе тисячі кілометрів колійних шляхів та сотні станцій.

Проте, інфраструктура потребує реконструкції та модернізації для підвищення її ефективності та безпеки.

2. Електрифікація та зелена мобільність. Україна активно працює над електрифікацією своєї залізничної мережі та використанням відновлювальних джерел енергії. Це сприяє зменшенню викидів CO2 та інших шкідливих речовин.

3. Рухомий склад. Парк потягів та вагонів в Україні різноманітний, але часто застарілий. Поліпшення рухомого складу та введення сучасних поїздів є важливим завданням.

4. Безпека та технічний стан. Забезпечення безпеки руху поїздів та пасажирських перевезень є однією з головних пріоритетних задач. Регулярний технічний огляд та обслуговування рухомого складу є важливими аспектами.

5. Тарифи та доступність. Ціни на квитки та послуги громадського транспорту в Україні зазвичай відносно доступні, але ціни можуть коливатися і варіюватися залежно від різних факторів.

1.2 Різновиди поїздів та квитків

Укрзалізниця надає різноманітні види пасажирських перевезень, які включають у себе:

1. Міжміські поїзди. Це поїзди, які з'єднують міста та регіони України. Вони служать для довізу пасажирів між великими містами та населеними пунктами.

2. Міжнародні поїзди. Укрзалізниця також забезпечує міжнародні пасажирські перевезення до країн-сусідок та інших європейських країн. Ці поїзди з'єднують Україну з Польщею, Росією, Угорщиною, та іншими країнами.

3. Регіональні поїзди. Для забезпечення регіональної мобільності Укрзалізниця пропонує регіональні поїзди, які обслуговують короткі

маршрути та перевозять пасажирів між обласними центрами та іншими населеними пунктами.

4. Пасажирські електрички. У багатьох містах України і їх околицях діють електрички, які забезпечують пасажирські перевезення на коротких відстанях.

5. Пасажирські автобуси та автомобільні перевезення. Укрзалізниця також пропонує пасажирські автобусні маршрути та автомобільні перевезення для пасажирів.

6. Пасажирські послуги в метрополітенах. У деяких великих містах України є метрополітени, які забезпечують міські пасажирські перевезення в метрополітенах.

7. Спеціальні поїзди. Іноді Укрзалізниця організовує спеціальні пасажирські поїзди для великих подій, таких як концерти, спортивні змагання та інші масові заходи.

8. Літні поїзди. У літній період Укрзалізниця може запускати додаткові поїзди для відпочинку на морських або інших курортах.

Укрзалізниця намагається забезпечити різноманітні пасажирські перевезення, щоб задовольнити потреби пасажирів у мобільності та зручності у всіх регіонах України.

Укрзалізниця пропонує різні класи та види вагонів для пасажирів, які шукають різний рівень комфорту та обслуговування. Ось деякі з найпоширеніших класів та типів вагонів:

1. Купе. Вагони класу "Купе" зазвичай мають окремі купе для пасажирів. Кожне купе може вміщувати від 4 до 6 осіб та має ліжка для сидіння та спання. Зазвичай це комфортний та популярний клас для середньої дальності та нічних поїздок.

2. Плацкарт. Плацкартні вагони розраховані на більше пасажирів. Вони мають велику відкриту зону з ліжками для сидіння та спання, розділену перегородками. Це економічний вибір для подорожей на середні та довгі відстані.

3. Спальний вагон (Люкс). У спальних вагонах, також відомих як вагони "Люкс", є окремі купе з власними ванними кімнатами та ліжками для сидіння та спання. Це найвищий рівень комфорту, і вони часто використовуються для розкішних або делегаційних поїздок.

4. Економ-клас. Економ-клас, як правило, пропонує найнижчий рівень вартості квитків. Вагони цього класу можуть мати відкриту зону для сидіння та обмежену кількість послуг. Вони підходять для тих, хто шукає економічний варіант подорожі.

5. Приватні купе. Деякі поїзди можуть мати приватні купе, які можуть бути орендовані для індивідуального використання. Це може бути зручно для груп або сімей, які хочуть мати окремий простір.

6. VIP-вагони. Деякі довгі маршрути пропонують VIP-вагони, які мають розкішне обладнання та послуги для вищого рівня комфорту та приватності.

Вибір класу та типу вагона залежить від вашої потреби, бюджету та особистих вподобань. Але найбільш затребуваними є:

1. Економ-клас. Цей клас популярний серед тих, хто шукає бюджетний варіант подорожі.

2. Купе. Цей клас є популярним вибором для багатьох пасажирів, особливо для тих, хто подорожує середніми відстанями. Вони надають комфорт та приватність, оскільки кожне купе має власні місця для сидіння та спання.

1.3 Аналіз існуючих механічних замків дистанційного керування

Для проведення аналізу було обрано 3 популярних варіанти механічних замків, які володіють функцією дистанційного керування. Для забезпечення безпеки і контролю доступу за допомогою мобільних додатків, які

інтегруються з механічними замками, надаючи користувачам зручність та ефективність у керуванні доступом до приміщень або об'єктів:

1. August Smart Lock;
2. Schlage Sense Smart Deadbolt;
3. Yale Assure Lock SL;

August Smart Lock – це механічний замок[4], який можна встановити на існуючий замок вашої двері. Він оснащений модулем Bluetooth, який дозволяє вам відкривати та закривати двері за допомогою мобільного додатка та має наступні функції:

1. Відкривання та закривання дверей за допомогою смартфона через Bluetooth.
2. Можливість надсилати електронні ключі іншим користувачам.
3. Моніторинг стану дверей та журнал входжень.

Переваги August Smart Lock:

- Легкість встановлення: August Smart Lock може бути легко встановлено на існуючий механічний замок без значних змін у дверях чи фіксації.
- Електронні ключі: Можливість надсилати електронні ключі іншим користувачам для тимчасового чи постійного доступу.
- Журнал входжень: Функція моніторингу стану дверей та ведення журналу входжень.
- Співпраця з іншими смарт-пристроями: Сумісність з різними платформами смарт-дому, такими як Apple HomeKit, Google Assistant та Amazon Alexa.

Недоліки August Smart Lock:

- Потреба в батареях: Вимагає живлення від батарей, тому важливо слідкувати за їх станом та замінювати при необхідності.
- Вартість: Ціна може бути вищою порівняно з іншими керованими замками.
- Залежність від Bluetooth: Зовнішній доступ може бути обмежений

дальністю Bluetooth, що може призвести до проблем при віддаленому керуванні.

- Питання приватності: Використання електронних ключів може породжувати питання щодо приватності та безпеки даних.

Schlage Sense Smart Deadbolt – це високоякісний розумний замок[5] з підтримкою Apple HomeKit з коробки, має сенсорну клавіатуру, і при натискані на цифру, активується підсвічування. Він може бути інтегрований з іншими платформами смарт-дому. Також код для замка можна задати декількома способами:

- Перебуваючи в зоні дії Bluetooth.
- Віддалено за допомогою Wi-Fi мосту.
- Віддалено за допомогою Apple TV або Ipad у вашій домашній мережі.

Переваги Schlage Sense Smart Deadbolt:

- Сумісність з платформами смарт-дому: Вбудована сумісність з платформами, такими як Apple HomeKit, розширює можливості використання
- Бездротовий зв'язок: Використання Bluetooth для бездротового зв'язку, що робить установку та налаштування простіше.
- Аудіосистема: Наявність аудіосистеми дозволяє спілкуватися з особами за дверима безпосередньо через замок.
- Система введення коду: Можливість використовувати кодову систему для доступу.

Недоліки Schlage Sense Smart Deadbolt:

- Вимоги до енергопостачання: Вимагає джерела живлення, що може викликати проблеми при відключенні електроенергії.
- Висока ціна: Може бути вартість вищою, ніж у звичайних механічних замків.
- Обмежена сумісність: Може бути менше сумісний із платформами порівняно з іншими моделями.
- Можливість збою аудіосистеми: Аудіосистема може мати збої або не працювати стабільно.

Assure Lock SL – найтонший розумний замок[6] на ринку з сенсорною клавіатурою для входу без ключа. Замок поставляється з комплектом Connected by August та в своъй комплектації має такі функції:

1. Надавання спільного доступу до блокування та розблокування дверей
2. Можливість відслідковування хто входить і виходить з будь-якого місця за допомогою додатку Yale Access App - додатковий хаб не потрібен.
3. Автоматичне відчинення дверей при повернені додому з телефоном у кишені.
4. Можливість автоматичного блокування дверей за заданою періодичністю для підвищення безпеки.

Переваги Assure Lock SL:

- Система введення коду та сумісність: Можливість використовувати коди для доступу, що може бути зручно. Також інтегрується з різними платформами смарт-дому, що забезпечує розширені можливості.
- Віддалений доступ: Забезпечення віддаленого керування через мобільний додаток.
- Система надсилання повідомлень: Можливість отримання повідомлень про події, пов'язані зі замком.
- Сумісність із смарт-платформами: Інтеграція з Amazon Alexa, Google Assistant, Apple HomeKit дозволяє легше керувати замком.
- Система введення коду: Можливість використовувати коди для доступу, що може бути зручно та безпечно.

Недоліки Assure Lock SL:

- Можливість збою в роботі Bluetooth: Залежність від Bluetooth може призвести до проблем в роботі в разі його відмови.
- Необхідність регулярного оновлення ПЗ: Оновлення програмного забезпечення може вимагати від користувача деяких зусиль та викликати тимчасові незручності.
- Потреба в енергозабезпеченні: Не здатний працювати без підключення до енергомережі.

Після аналізу переваг та недоліків додатків для механічних замків з дистанційним керуванням можна зробити висновок, що, незважаючи на значний технологічний прогрес у цьому напрямі, ідеального рішення поки що не існує. Є певні аспекти, які варто вдосконалити для досягнення максимальної зручності та безпеки користувачів.

1.4 Висновки

В даному розділі підводяться підсумки по вивченню теоретичного матеріалу. Розкрита сфера залізничного транспорту на території України, його стан та тенденціїї. Описані основні моменти роботи залізничного транспорту. Також були проаналізовані аналоги додатків, що використовуються сьогодні це складний процес, та важливо пам’ятати про вище наведені пункти. Вони здатні пришвидшити розробку, покращити якість і надійність.

2 ОБҐРУНТУВАННЯ МЕТОДІВ РЕАЛІЗАЦІЇ

2.1 Обґрунтування вибору платформи реалізації

З вище наведених аналогів можна зробити висновок, що типології систем, що розробляються, мають спільні недоліки, такі як локальне зберігання, передача даних в режиміреальногочасута швидкість обробки інформації.

Длятого, щоб вирішити вище зазначені недоліки,для розробки системи булообраноформат сучасного нативного додатку на платформі iOS. НативнідодаткидляiOS дозволяють створити систему, яка постійно підтримується та оновлюється, без необхідності покладатися на сторонні компанії, що розробляють SDK та фреймворки. З нативним додатком для iOS ви отримуєте систему, яка працює ефективно і швидко, зі спеціально розробленими нативними інструментами від Apple, які постійно підтримуються і оновлюються, без необхідності покладатися на сторонні компанії, які розробляють SDK і фреймворки. Цепов'язаноз тим,що інструменти розроблені спеціально для платформи iOS і відповідають усім специфікаціям, правилам, недолікам, перевагам або версіям системи [7], [8]. Інтеграція нейронних мереж в систему також забезпечує швидкість обробки великих обсягів даних. Для зберігання великих обсягів даних використовується локальне сховище, передача даних на сервер здійснюється за допомогою REST API та зручногой інтуїтивно зрозумілого інтерфейсу користувача завдяки використанню нової технології SWIFTUI [9], [10].

2.2 Обґрунтування вибору операційної системи

На сучасному ринку смартфонів домінують дві платформи. Одна з них - це платформа iOS від Apple, операційна система, на якій працює популярна

серія смартфонів Apple iPhone. Інша - Android від Google. Операційна система Android використовується не лише пристроями Google, але й багатьма іншими виробниками для розробки власних смартфонів та інших смарт-пристроїв.

Хоча між цими двома платформами є певна схожість, коли йдеться про розробку додатків, для розробки iOS та Android використовуються різні комплекти розробки програмного забезпечення (SDK) та набори інструментів для розробки. У той час як Apple використовує iOS виключно для власних пристроїв, Google пропонує Android іншим компаніям за умови дотримання певних вимог, таких як включення певних додатків Google на пристроях, які вона пропонує [8]. Порівняльні характеристики iOS та Android [11], [12]:

1)Продуктивність. Завдяки високооптимізованим апаратним і програмним процесам iPhone старого покоління, як правило, перевершують нові флагмани Android у бенчмарках. Загалом, Android, як правило, кращий з точки зору апаратних специфікацій, але iPhone, як правило, кращий з точки зору продуктивності.

2) Більшість сучасних телефонів, будь то Android або iOS, бюджетні або флагманські, використовують IPS-дисплеї. Сюди входять всі iPhone до iPhone8 і iPhoneXR; iPhone від моделі X до моделі 14 мають OLED-дисплеї; для телефонів Android більшість виробників, крім Samsung, використовують IPS замість OLED-дисплеїв.

Як IPS LCD, так і OLED мають свої переваги та недоліки. З технічної точки зору, OLED-дисплеї мають ряд переваг:

1. 1. Висока енергоефективність; при використанні OLED-дисплея кожен піксель підсвічується індивідуально, тому немає необхідності в активному підсвічуванні всього екрану при включенні дисплея

2. OLED-дисплеї мають кращі кути огляду; висококласні IPS-дисплеї, такі як ті, що встановлюються в iPhone, майже співставні з OLED-дисплеями в цьому відношенні, але різниця все одно помітна.

3. OLED-дисплеї легше відбивають світло, ніж IPS. Це особливо важливо для людей, які проводять багато часу на вулиці і не люблять відблисків.

3) Додатки Android, безумовно, має перевагу над iOS, коли мова йде про кількість додатків, доступних в App Store. Ця різниця в основному пов'язана з відкритим вихідним кодом Android і м'якішою політикою Google Play. Крім того, Android-пристрої мають доступ до сторонніх магазинів. Також, на відміну від iOS, пристрої Android дозволяють користувачам завантажувати додатки вручну, прикріплюючи "apk-файл".

Хоча Android має кількісну перевагу над iOS, суворий контроль Apple над додатками гарантує, що всі програми працюють належним чином і є повністю безпечними [13]. [14].

4) Оновлення. Як згадувалося раніше, пристрої Apple відомі своєю довговічністю та майже повною відсутністю погіршення продуктивності. Основною причиною такого довголіття є те, що кожен пристрій iOS отримує безперервні оновлення операційної системи протягом приблизно п'яти років після її першого випуску.

Виходячи з вище зазначених характеристик, система iOS була обрана для вирішення завдань оптимізації продуктивності та процесів, а також високої якості додатків.

2.3 Обґрунтування вибору середовища розробки мобільних додатків

Було вирішено розробляти додаток на платформі iOS, що працює на пристроях iPhone, iPad та iPod Touch. Компанія Apple надає інструменти та ресурси для створення iOS-додатків та аксесуарів для цих пристроїв, а саме нативне середовище розробки Xcode.

Xcode - це інтегроване середовище розробки (IDE), створене компанією Apple для розробки програмного забезпечення для macOS, iOS, watchOS і

tvOS. Це єдиний офіційно підтримуваний інструмент для створення та публікації додатків в магазині додатків Apple, призначений для використання всіма, від початківців до досвідчених розробників [15]. Xcode включає в себе всі інструменти, необхідні для створення додатків, в одному програмному пакеті:

- Текстовий редактор.
- Компілятор.
- Система складання.

За допомогою Xcode можна писати, компілювати, налагоджувати та відправляти додатки до магазину додатків Apple.

Як редактор коду Xcode підтримує величезну кількість мов програмування - C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit і Swift. Він використовує моделі програмування Cocoa, Carbon і Java.

Xcode був обраний через вищезгадані можливості для розробки програмного забезпечення, вирішення завдань, тестування, аналізу та моніторингу [16], [17].

2.4 Обґрунтування вибору архітектури системи

Патерни проектування мали значний вплив на розробку програмного забезпечення. Подібно до веб-додатків, впровадження мобільних додатків створило певні перевірені патерни та стандарти для подолання викликів та обмежень у розробці мобільних додатків. Більшість мобільних додатків розробляються з використанням низькоякісного коду і не базуються на патернах архітектурного дизайну. Розробка мобільних додатків з використанням правильних патернів проектування може ефективно пов'язати інтерфейс користувача з моделлю даних і бізнес-логікою. Це впливає на те, як виглядає вихідний код [18], [19].

Основні патерни дизайну:

1) Apple MVC. MVC - це перший підхід до визначення та реалізації розробки програмного забезпечення відповідно до власних обов'язків. Спочатку розроблений для настільних комп'ютерів, але широко використовуваний основними мовами програмування як архітектура для веб-додатків, MVC включає наступні три компоненти для кожного об'єкта:

- Модель - відповідає за дані або рівень доступу до даних.
- Вид - відповідає за графічне відображення даних.
- Контролер - діють як сполучна ланка між поданням і моделлю. Контролери не діють як посередники між представленнями та моделями. Контролер змінює модель на основі активності користувача в поданні і використовує подання для оновлення змін Приклад того, як працює MVC, показано на рисунку 2.1[20].

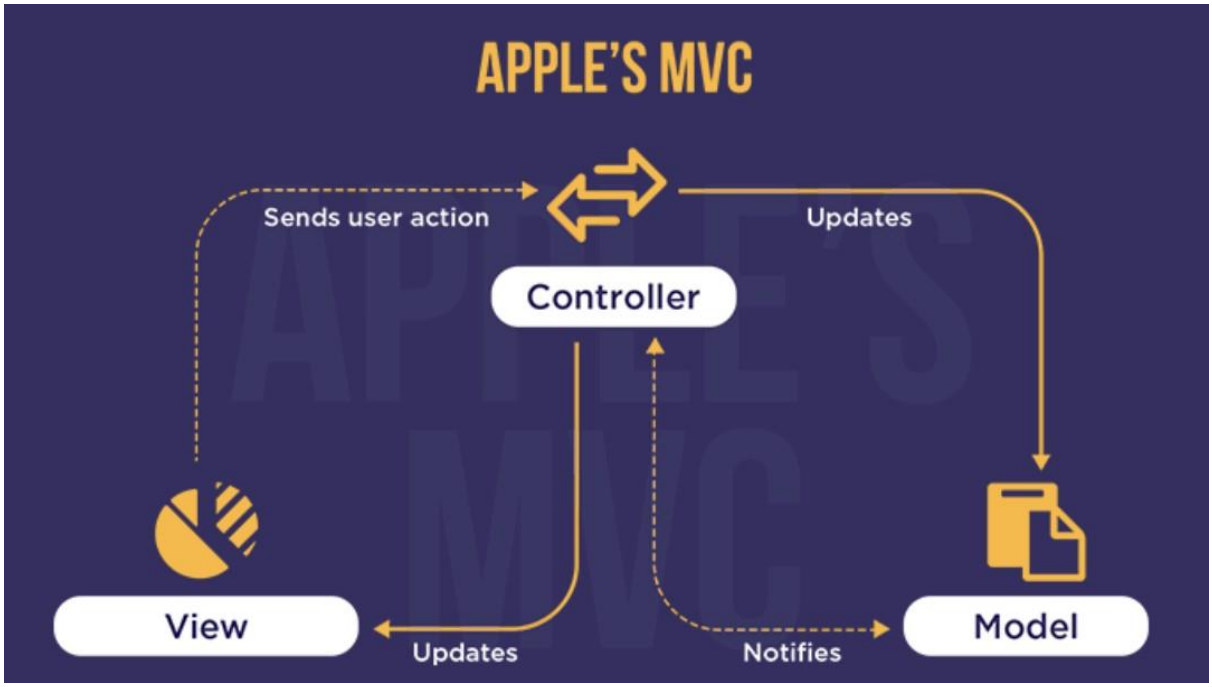

Рисунок 2.1 - Шаблон MVC

Переваги над класичним патерном проектування MVC.

У цьому патерні проектування немає прямого зв'язку між моделлю та представленням. Крім того, контролер містить логіку обробки представлення. Такий розподіл обов'язків робить його більш придатним для розробки додатків.

Недоліки патерну проектування Apple MVC.

Оскільки контролер глибоко залучений в життєвий цикл представлення, важко визначити їх поділ.

2) Шаблон MVVM містить три компоненти: Model, View і ViewModel. Тут ViewModel діє як агент: він ініціює зміни в моделі, а також оновлює себе відповідно до оновленої моделі. Вона також передбачає зв'язування даних і дій користувача, подібно до моделі MVP Supervising Controller, але не між View і Model, а між View і ViewModel [21]. Це дозволяє View оновлювати себе відповідно до посилання на ViewModel, як показано на діаграмі нижче 2.2.

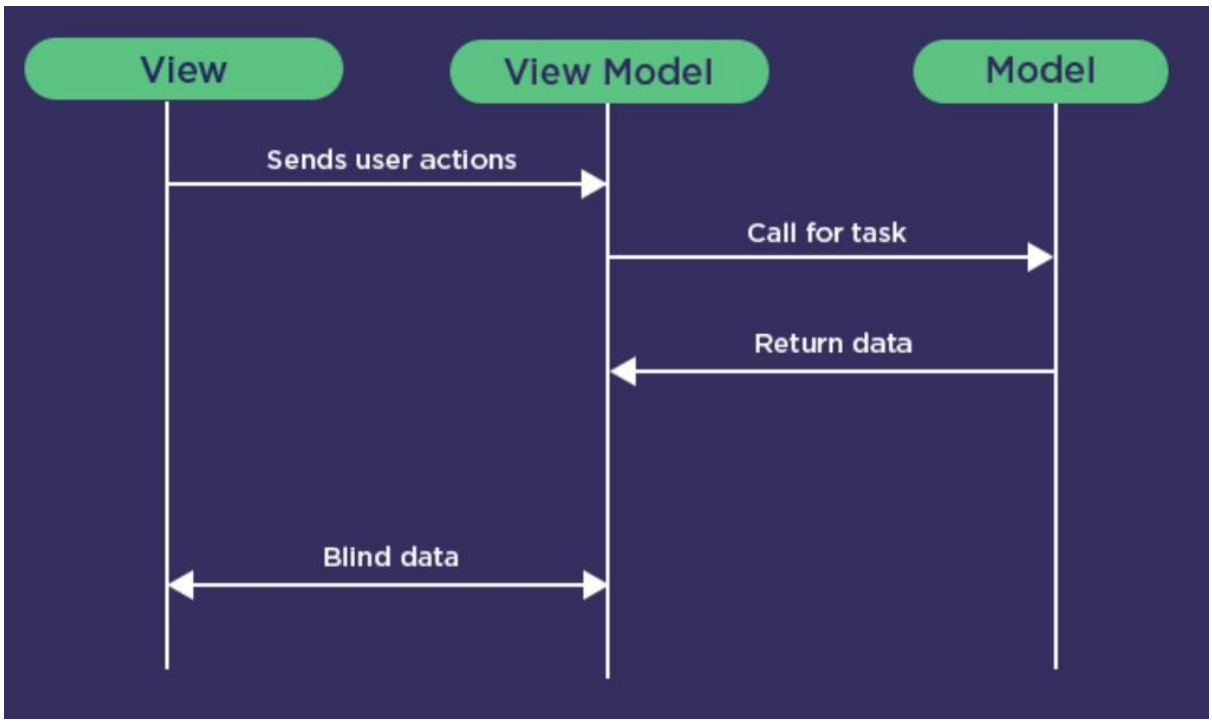

Рисунок 2.2 - Шаблон MVVM.

Переваги порівняно з патерном проектування MVC.

Односторонній зв'язок між View та ViewModel зменшує кількість рядків коду, необхідних для синхронізації View та ViewModel. Кращі можливості тестування та розширення.

3) VIPER. Шаблон містить ідею розподілу обов'язків за п'ятьма рівнями:

1. View— клас, який показує користувачеві інтерфейс програми, а також отримує відповідь.
2. Interactor — містить бізнес-логіку програми.
3. Presenter — включає бізнес-логіку, пов'язану з користувацьким інтерфейсом.
4. Entity — містить прості об'єкти даних.
5. Router – Обробляє обмін між модулями Viper. Приклад роботи та зв’язків між модулями у VIPER зображено на рисунку 2.3 [22].

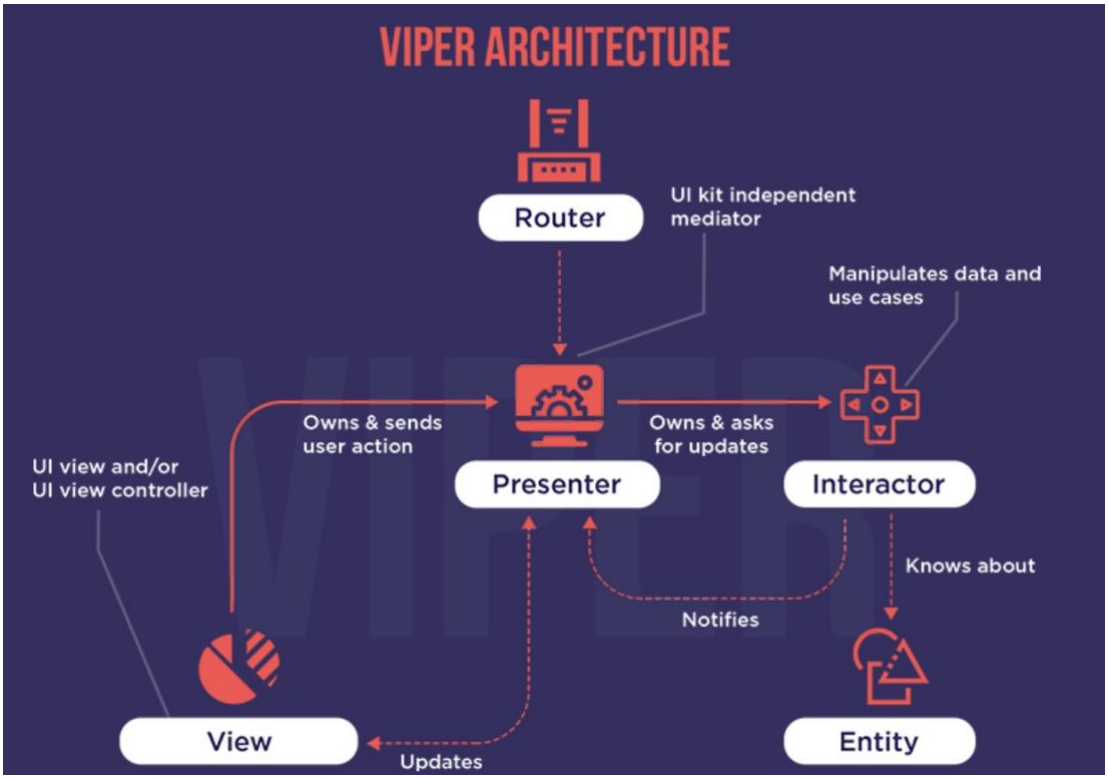

Рисунок 2.3 - Шаблон VIPER.

Переваги над моделлю MV(x).

Ідея роз'єднання вирішує проблеми компіляції та навігації попередньої архітектури. Слабкий зв'язок між мережевою архітектурою та кожним модулем зменшує навантаження на модифікацію та виправлення помилок. Модульний підхід забезпечує якісне середовище для модульного тестування.

2.4.1 Обґрунтування вибору архітектури

Враховуючи вищезазначені особливості та потреби, з якими стикається архітектурний шаблон, було обрано шаблон проектування MVVM, який повністю вирішує проблеми, окреслені в дослідженні, без зайвих залежностей між модулями, надлишкового коду або абстракцій.

2.5 Обґрунтування вибору мови програмування

Для реалізацій функцій додатку було запропоновано дві мови програмування, такі як Swift та Objective C. В таблиці 2.1 наведено порівняльну характеристику та головні відмінності Swift та Objective C.

Таблиця 2.1 - Порівняння характеристик Swift та Obective-C

Властивості	Swift	Objective-C
Дизайн	Розроблений для розробки операційних систем Apple.	Розроблено як об'єктно-орієнтований разом із функцією обміну повідомленнями Smalltalk.

Продовження таблиці 2.1

Спадкування	Не допускає множинного успадкування	Не допускає множинного успадкування
Ліцензія	Це проект під ліцензією Apache із відкритим вихідним кодом	Перебуває під ліцензією GPL (General Public License)
Тип	Статичний і типізований	Динамічна типізація
Логічні оператори	Використовує істинні та хибні значення	C++ використовує YES, NO і BOOL
Шаблони та бібліотеки	Підтримує кілька бібліотек разом з Objective C	В Objective C відсутні бібліотеки шаблонів

Також більш детальна інформація про пункти з таблиці 2.1, наведено нижче.

- Спадкування.

Swift переглянув загальні правила успадкування. Так, більше не потрібно ставити коми в кінці рядків або брати умовні оператори в круглі дужки в операторах if/else. Іншою важливою зміною є те, що виклики методів більше не розміщуються один всередині одного: У Swift методи та функції викликаються за допомогою стандартного списку параметрів, відокремлених комами в круглих дужках. В результаті ми отримали чисту та виразну мову зі спрощеним синтаксисом та граматикою.

- Swift легше підтримувати ніж Objective-C.

Мова C повинна підтримувати два файли коду для збільшення часу налагодження та ефективності розробки додатків; ця вимога не може розвиватися Objective-C без розвитку C. Вона також була перенесена в Objective-C. Swift усуває вимогу щодо двох файлів; компілятори Xcode та LLVM можуть розуміти залежності та автоматично вносити інкрементні зміни. Як результат, ітеративний процес розділення вмісту (заголовних файлів) і тіла (файлів додатків) залишився в минулому: Swift об'єднує заголовні файли Objective-C (.h) та файли додатків (.m) в один файл коду (.swift).

- Безпечність.

Особливістю мови Objective-C є те, як вона працює з вказівниками, особливо з нульовими вказівниками: В Objective-C, якщо ви викликаєте метод з нульовою (неініціалізованою) змінною-покажчиком, нічого не відбудеться. Вирази та рядки коду стають марними, і хоча здається, що вирази не аварійно завершуються, насправді вони містять багато помилок. Необов'язкові типи дозволяють nil мати необов'язкове значення у Swift-коді, а це означає, що помилки компілятора генеруються при написанні поганого коду. Це створює короткий цикл зворотного зв'язку, що дозволяє писати код з більшою впевненістю. Проблеми можна усунути, коли код вже написаний, що значно скорочує час і гроші, витрачені на виправлення помилок.

- Управління пам'яттю.

Swift є уніфікованою мовою, на відміну від Objective-C. Підтримка автоматичного підрахунку посилань (ARC) є повною у процедурному та об'єктно-орієнтованому коді; в Objective-C ARC підтримується у Cocoa API та об'єктно-орієнтованому коді, але недоступна у C-коді та API, таких як Core Graphics. Він недоступний у коді C та API, таких як Core Graphics. Це означає, що розробник бере на себе управління пам'яттю при використанні Core Graphics API або інших застарілих API, доступних в iOS; витоки пам'яті, які можуть статися в Objective-C, не можуть статися в Swift. Це відбувається тому, що ARC виконує все управління пам'яттю під час компіляції, а оскільки ARC

у Swift працює як з процедурним, так і з об'єктно-орієнтованим кодом, витрати, витрачені на управління пам'яттю, можуть бути зосереджені на основній логіці програми та нових функціях [23], [24].

Враховуючи вище перелічені порівняльні характеристики мов програмування, було обрано мову Swift.

2.6 Обґрунтування вибору технологій для цифрового ключа

Існує декілька технологій для передавання цифрового ключа Near Field Communication, Bluetooth Low Energy та Bluetooth. Нижче представлено головні переваги і недоліки.

NFC (Near Field Communication) - це технологія безконтактного зв'язку[25], яка дозволяє обмінювати даними між пристроями на невеликій відстані (зазвичай до 4 см) за допомогою радіочастотних сигналів. NFC використовується в різних сферах, включаючи мобільні платежі, обмін контактами, доступ до будівель, транспортні квитки та інші застосування.

Переваги NFC:

1. Простота використання:.NFC дуже простий у використанні. Для обміну даними двома пристроями потрібно просто прикласти їх один до одного.
2. Безпека. NFC вважається відносно безпечною технологією, оскільки вона вимагає фізичного наближення пристроїв один до одного. Це робить підміну даних більш складною.
3. Широке застосування. NFC використовується в різних галузях, включаючи мобільні платежі, копіювання інформації зі стікерів та транспортні квитки.
4. Швидкість передачі даних. NFC може передавати дані дуже швидко, що робить його ідеальним для миттєвого обміну інформацією.

5. Ефективність енергоспоживання. NFC споживає менше енергії, ніж інші безконтактні технології, такі як Bluetooth.

Недоліки NFC:

1. Обмежена дальність. Однією з найбільших обмежень NFC є дуже маленька дальність передачі даних, що обмежує використання цієї технології в більш великих об'єктах або витворах, де потрібно більше зв'язку.

2. Сумісність пристроїв. Для передачі даних за допомогою NFC обидва пристрої повинні підтримувати цю технологію. Це може бути обмеженням в деяких випадках.

3. Міжнародні стандарти. Існують різні стандарти NFC, і це може призвести до несумісності між пристроями.

4. Безпека даних. Хоча NFC вважається відносно безпечною, існує ризик підміни даних або недостатньої захисту інформації, яка передається через NFC.

Bluetooth Low Energy (BLE) - це бездротова технологія[26], призначена для передачі даних на короткі відстані з низьким споживанням енергії. Вона знайшла широке застосування в IoT-пристроях, сенсорах, медичних пристроях та інших областях.

Переваги BLE:

1. Енергоефективність. BLE споживає значно менше енергії, ніж традиційний Bluetooth, що робить його ідеальним для батарейкових та енергоефективних пристроїв.

2. Дальність зв'язку. BLE має достатньо прийнятну дальність зв'язку, яка може сягати до 30 метрів або більше в ідеальних умовах.

3. Швидкість передачі даних. Вища швидкість передачі даних порівняно з деякими іншими технологіями, такими як NFC, що дозволяє обробляти більше інформації.

4. Широке застосування. BLE використовується в різних галузях, включаючи смартфони, медицинські пристрої, фітнес-трекери, домашні апарати та IoT-пристрої.

5. Сумісність з багатьма пристроями. BLE підтримується багатьма сучасними смартфонами та планшетами, що дозволяє легко створювати додатки, які взаємодіють з багатьма пристроями.

Недоліки BLE:

1. Не підходить для великих об'єктів. Як і в інших технологіях, дальність BLE може бути обмеженою, що робить його менш практичним для великих приміщень або на відкритому повітрі.

2. Не всі пристрої підтримують BLE. Щоб взаємодіяти з BLE, пристрої повинні підтримувати цю технологію. Не всі старі пристрої мають цю можливість.

3. Значні обмеження щодо передачі обсягу даних. Для передачі великих обсягів інформації, таких як мультимедійні файли, BLE може бути неефективним.

4. Специфічність програмування. Розробка додатків для BLE може бути складнішою порівняно з деякими іншими технологіями, і вимагає певного рівня знань в області бездротових комунікацій.

Bluetooth - це технологія бездротового зв'язку[27], яка дозволяє обмінювати даними на відстані до 100 метрів між пристроями. Вона використовується в широкому спектрі пристроїв, включаючи гарнітури, колонки, клавіатури, миші, смартфони і планшети.

Переваги Bluetooth:

1. Універсальність. Bluetooth є широко підтримуваною технологією, і майже всі сучасні смартфони та комп'ютери мають вбудовану підтримку Bluetooth.

2. Дальність зв'язку. Bluetooth дозволяє обмінювати даними на відстані до 100 метрів, що робить його ідеальним для більших просторів.

3. Швидкість передачі даних. Вища швидкість передачі даних порівняно з деякими іншими бездротовими технологіями, такими як NFC і BLE.

4. Широкий спектр застосувань. Bluetooth використовується в різних галузях, включаючи аудіо- та відео-пристрої, медичні пристрої, автомобільну електроніку і багато інших.

5. Зв'язок між багатьма пристроями. Bluetooth дозволяє підключати один пристрій до кількох інших пристроїв одночасно, що дозволяє зберігати взаємодію між різними пристроями.

Недоліки Bluetooth:

1. Споживання енергії. Традиційний Bluetooth споживає більше енергії порівняно з технологією Bluetooth Low Energy (BLE), що може призводити до скорочення часу автономної роботи пристроїв.

2. Сумісність версій. Існують різні версії Bluetooth, і не всі вони сумісні між собою. Це може створювати проблеми з підключенням.

3. Обмежена швидкість передачі даних для деяких застосувань. Для передачі великих обсягів інформації, таких як відео високої якості, можуть знадобитися інші технології.

4. Значний обсяг радіочастотного спектра. Bluetooth використовує значний обсяг радіочастотного спектра, і це може спричиняти перенасичення в переповнених місцях.

2.6.1 Обґрунтування вибору технологій

На підставі порівняння технологій NFC, Bluetooth і Bluetooth Low Energy (BLE), можна зробити висновок, чому саме BLE може бути кращим вибором для вашого додатку, який спрямований на розробку системи для придбання квитків на потяг з електронним Bluetooth ключем до купе:

1. Енергоефективність: Оскільки ваша система, ймовірно, працюватиме на батареях або інших джерелах живлення, енергоефективність є критичною. BLE відомий своєю низькою споживаною енергією, що дозволить забезпечити довший час автономної роботи пристроїв, таких як електронні ключі.

2. Дальність зв'язку: Хоча NFC має обмежену дальність зв'язку (до 4 см)

і Bluetooth має більшу дальність (до 100 метрів), BLE надає золоту середину, забезпечуючи дальність зв'язку до 60 метрів. Це може бути ідеальним для системи, яка потребує обміну даними на невеликій відстані між мобільним пристроєм та електронним ключем.

3. Швидкість передачі даних: BLE має достатньо високу швидкість передачі даних для вашого застосунку, оскільки ви, ймовірно, не будете передавати великі обсяги даних, але все ж потребуєте надійну та швидку комунікацію між пристроями.

4. Широкий спектр застосувань: BLE добре підходить для різних застосувань, включаючи IoT-пристрої, сенсори, а також додатки для мобільних пристроїв, яким необхідно взаємодіяти з різними типами обладнання.

5. Сумісність зі смартфонами: Більшість сучасних смартфонів мають підтримку BLE, що робить його ідеальним вибором для додатків, які взаємодіють з мобільними пристроями.

Отже, з огляду на ваші потреби у високій енергоефективності, досить середньому діапазону дальності зв'язку та невеликі обсяги передачі даних, BLE може бути кращим вибором для розробки системи для придбання квитків на потяг з електронним Bluetooth ключем до купе.

2.7 Обґрунтування вибору локального сховища

Існує два основні способи зберігання даних у застосунку iOS: CoreData, SQLite і Realm. Порівняльні характеристики CoreData, SQLite та Realm представлено в таблиці 2.1.

Таблиця 2.1 - Характеристики CoreData, SQLite та Realm.

CoreData	SQLite	Realm
Core Data - це основою для управління об'єктним графом. Об'єктний граф - це набір взаємопов'язаних об'єктів.	SQLite - Це дуже продуктивний і легкий рушій баз даних, що робить його придатним для мобільних пристроїв та інших вбудованих систем.	Realm - це кросплатформенна мобільна база даних, яка працює швидше та ефективніше, ніж SQLite і Core Data
Основні дані зберігають вміст об'єкта в детальному вигляді	SQLite орієнтований в основному на традиційне зберігання вмісту таблиць	Realm зберігає об'єкти в пам'яті у найбільш зручному для відображення форматі, використовує технологію стовпчастого зберігання для швидкого доступу.
CoreData не є базою даних	SQLite - це база даних відносин	Realm не є базою даних, але може використовувати граф об'єктів для опису зв'язків між об'єктами.

- Переваги використання Realm над CoreData. Немає потреби в складних схемах. Немає необхідності визначати схему перед зберіганням даних. Дані

автоматично зберігаються як об'єкти, тому немає необхідності окремо керувати рядками та стовпцями структурованих даних у додатку. Це спрощує управління даними порівняно з CoreData.Realm може зберігати практично будь-який тип об'єктів, від простих типів, таких як рядки, числа і дати, до більш складних типів, таких як масиви, списки і вкладені об'єкти. Це дозволяє легко моделювати зв'язки між різними сутностями без необхідності створювати явну схему.

- Переваги використання CoreData над Realm За замовчуванням, Apple Foundation Data Framework дозволяє розробникам просто налаштувати базу даних і негайно почати програмувати додаток. Це чудовий варіант, якщо додаток не надто складний, а база даних не надто велика. Однак, якщо база даних більша (наприклад, у випадку SQLite), це стає більш складним, оскільки SQLite не зберігає дані так, як це було задумано компанією Apple. Крім того, через обмеження CoreData, виконання складних запитів та управління синхронізацією бази даних на різних пристроях (наприклад, iCloud) стає ще більш громіздким.

Оцінивши запропоновані варіанти локальних сховищ, було обрано CoreData для великих обсягів даних та фреймворк UserDefaults для менших обсягів даних. Обидва фреймворки є нативними для платформи iOS, розроблені компанією Apple, мають широкий спектр функцій, оскільки вони є нативними, мають високу швидкість виконання коду, постійно оновлюються та підтримуються.

2.8 Обґрунтування вибору мікроконтролера

Існує велика кількість різновидів плат розглянемо три найбільш популярних Adafruit Feather, Arduino, ARM Cortex-M. Порівняльні характеристики Adafruit Feather, Arduino, ARM Cortex-M представлено в таблиці 2.2. [28], [29], [30]

Таблиця 2.2 - Характеристики CoreData, SQLite та Realm.

Adafruit Feather	Arduino	ARM Cortex-M
Adafruit Feather - ці плати, як правило, мають компактний розмір та вбудовану підтримку для різноманітних додаткових модулів та датчиків.	Arduino - це відкрита платформа для створення простих та складних електронних проектів.	ARM Cortex-M - це сімейство спроектоване спеціально для використання в системах з обмеженими ресурсами, таких як вбудовані системи, IoT-пристрої, мобільні пристрої та інші.
Ці плати призначені для легкості використання та можуть бути інтегровані в різноманітні проекти для збільшення їхньої функціональності.	Вона включає в себе не тільки апаратну частину (плати з мікроконтролерами), але і середовище програмування (Arduino IDE), що дозволяє розробникам легко програмувати мікроконтролери за допомогою мови програмування C/C++.	Контролери з архітектурою ARM Cortex-M забезпечують високу продуктивність при низькому енергоспоживанні.
Adafruit Feather Board може включати в себе мікроконтролер, модуль зв'язку (наприклад, Bluetooth або Wi-Fi), роз'єми для додаткових модулів, а також інші функції.	Основою Arduino є мікроконтролери, які можуть керувати різними компонентами, такими як світлодіоди, датчики, мотори та інші пристрої.	Ці мікроконтролери використовуються в широкому спектрі застосувань, включаючи автомобільну промисловість, аудіо/відео пристрої, пристрої для зв'язку та багато інших.

Arduino мікроконтролери вирізняються простотою використання та широким спектром готових бібліотек, спрощуючи розробку для початківців. Їх легко інтегрувати з різними сенсорами та актуаторами, а простий інтерфейс Arduino IDE сприяє швидкому старту. У контексті дипломної роботи, де важливо швидко реалізувати ідеї та використовувати готові рішення, Arduino може бути ефективним вибором порівняно з більш складними ARM Cortex-M мікроконтролерами, які зазвичай використовуються в більш високотехнологічних та обчислювально потужних системах.

Arduino мікроконтролери відзначаються високою доступністю і легкістю використання, завдяки широкому вибору готових модулів та зручному середовищу розробки. Це робить їх привабливими через прискорення процесу прототипування. У порівнянні з мікроконтролерами Adafruit Feather, Arduino володіє більшою екосистемою та простотою інтеграції, що особливо важливо в контексті роботи, де потрібно швидко та ефективно розробляти та тестувати пристрої.

2.9 Висновок

В даному розділі було досліджено, проаналізовано та обґрунтовано вибір платформи для реалізації програмного забезпечення, операційної системи, середовища розробки, мови програмування, архітектурних підходів, технології передачи даних, тип та фреймворк для зберігання даних в локальному сховищі та обгрунтування вибору мікроконтролера.

3 РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ

3.1 Основні компоненти системи контролю доступу

Для підключення до системи контролю доступу було обрано модуль Bluetooth 4.0 BLE AT-09 наведений на рисунку 3.1.

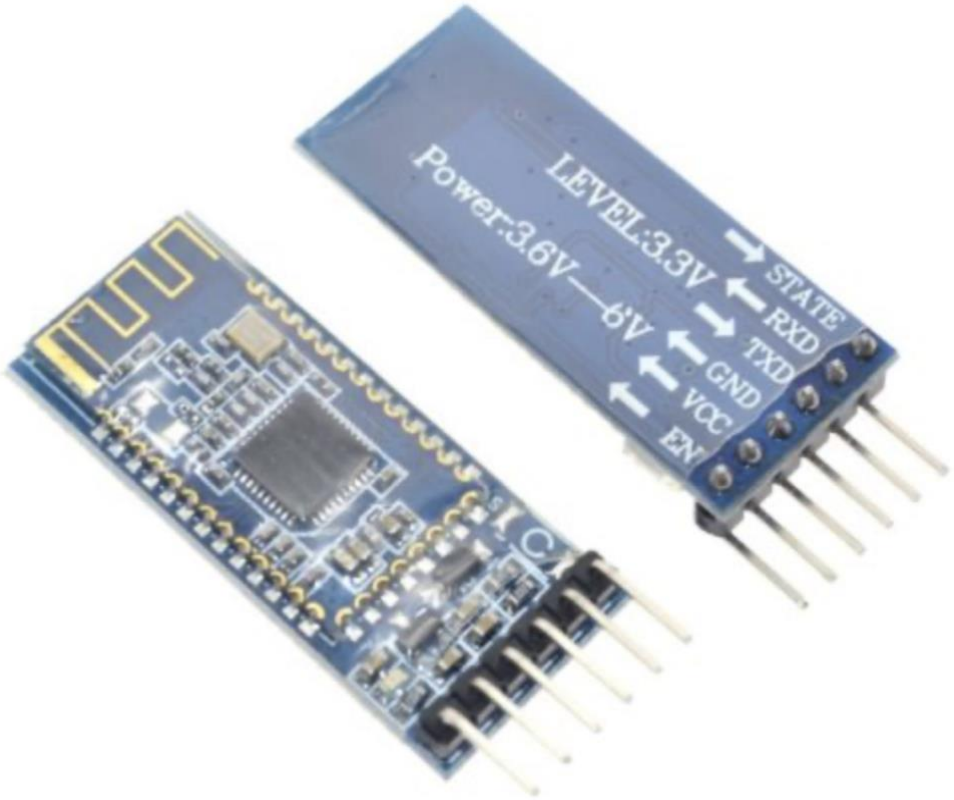

Рисунок 3.1 – Модуль Bluetooth 4.0 BLE AT-09

Характеристика модуля представлена в таблиці 3.1.

Таблиця 3.1 - Характеристики модуль Bluetooth 4.0 BLE AT-09.

Робоча частота	2,4 ГГц
Підтримувані профілі	GAP, GATT, L2CAP, SMP

Продовження таблиці 3.1

Споживання в режимі очікування	від 90мкА до 400мкА
Покриття	до 60 м
Антена	вбудована на друкованій платі
Напруга живлення	від 3,6 В до 5 В
Максимальна напруга живлення	6В
Швидкість передачі	1 Мбіт / с
Потужність передавача	0 дБм
Чутливість приймача	-94 дБм
Flash-пам'ять	128 кБ
Температурний діапазон	-40 ... + 85 ° С
Розміри	43 мм х 15 мм
Вага	4 грами

Модуль Bluetooth 4.0 BLE AT-09 відзначається високою енергоефективністю та компактним дизайном, що робить його ідеальним вибором для вбудовування в електронні пристрої з обмеженим живленням. Завдяки простоті інтеграції через командний інтерфейс AT, модуль легко інтегрується в різноманітні проекти. Його сумісність з Bluetooth 4.0 та доступність за вигідною ціною роблять AT-09 привабливим вибором для

розробників з обмеженим бюджетом, працюючи у сфері розробки електронних систем контролю доступу та інших інтегрованих рішень.

Мікросхема для системи було обрано Arduino UNO, яка предствлена на рисунку 3.2.

Рисунок 3.2 – Arduino UNO

Характеристика мікросхеми представлена в таблиці 3.2.

Таблиця 3.2 - Характеристики мікросхеми Arduino UNO

Робоча напруга контролера	- Вхід USB: 5В - Вхід VCC: 5В - Вхід Vin: 7,5В-12В
Цифрові входи/виходи	14
Аналогові входи	6
Флеш-пам'ять програм	32Кб
Оперативна пам'ять	2Кб

Продовження таблиці 3.2

Тактова частота	16 МГц
Розмір	68 х 53 х 15 мм

Arduino UNO, єдний з найпопулярніших мікроконтролерів у світі, визначається своєю простотою використання та високою гнучкістю для розробки пристроїв. Його відкритий інтерфейс забезпечує величезний спектр доступних бібліотек і модулів, що сприяє ефективній розробці пристроїв для широкого спектру застосувань. Зручність в програмуванні за допомогою Arduino IDE робить його ідеальним вибором для як досвідчених розробників, так і початківців, а його доступність та стабільність виробника гарантують надійність у використанні в проектах з автоматизації, робототехніки та інших напрямках інтегрованих систем.

Останнім основним компонентом для системи є двигун для демонстрування функціоналу замикання дверей, було обрано серводвигун Tower Pro MG996R, який предствлений на рисунку 3.3.

Рисунок 3.3 – Серводвигун Tower Pro MG996R

Характеристика серводвигуна представлена в таблиці 3.3.

Таблиця 3.3 - Характеристики серводвигун Tower Pro MG996R

Матеріал редуктора	пластик (нейлон)
Тип серводвигуна	Цифровий
Напруга живлення	4.8В
Кут повороту	360 °
Швидкість повороту	0.17сек/60° при 4.8В
Зусилля на вагу	9,4 кг/см при 4.8В

Продовження таблиці 3.3

Розміри	41 x 20 x 43 мм
Вага	55 г

Серводвигуни, що використовуються в інтегрованих системах та робототехніці, відзначаються високою точністю позиціонування та динамікою руху. Їхнє основне перевагою є здатність точно контролювати положення валів або виконувати заздалегідь визначені рухи, що робить їх незамінними для застосувань, де важлива точність та швидкість. Додатково, серводвигуни забезпечують гладке регулювання швидкості та моменту сили, що розширює їхні можливості в широкому спектрі завдань від промислових виробництв до автоматизованих систем управління.

3.2 Проектування системи

Проектування макетних плат перед їхньою реалізацією є важливим етапом в розробці електроніки і має кілька ключових переваг:

1) Верифікація схеми:

Проектування макетної плати дозволяє перевірити правильність схеми та електричних з'єднань. Це допомагає виявити помилки або проблеми в схемі до того, як ви створите реальний пристрій.

2) Оптимізація розміщення компонентів:

Проектування дає можливість оптимізувати розміщення компонентів на макетній платі. Ефективне розміщення може покращити електричні характеристики, зменшити взаємовплив між компонентами та взагалі поліпшити виявлені характеристики пристрою.

3) Відладка та налагодження:

Перед тим, як виробляти велику кількість фізичних пристроїв, проектування макетної плати дозволяє вам відлажувати та налагоджувати електронний пристрій у віртуальному середовищі. Ви можете виявити проблеми з електричними сигналами, взаємодію між компонентами та інші потенційні проблеми.

4) Масштабованість та виробничі аспекти:

Проектування PCB враховує аспекти виробництва, такі як оптимізація розташування компонентів для полегшення автоматизованого монтажу, вибір правильних матеріалів для шарів плати, та інші аспекти, що впливають на виробництво.

5) Мінімізація помилок виробництва:

Перед реалізацією проектування макетної плати дозволяє виявити потенційні проблеми, що можуть виникнути під час виробництва, такі як занадто вузькі проміжки між треками, неправильні розміри компонентів тощо.

6) Дослідження та тестування:

Засоби проектування дозволяють проводити різноманітні дослідження та тестування, такі як електричні та теплові аналізи, щоб забезпечити надійність та ефективність вашого пристрою.

Всі ці переваги сприяють створенню більш точного, ефективного та виробничого електронного пристрою. На рисунку 3.4 схематично зображено метод підключення.

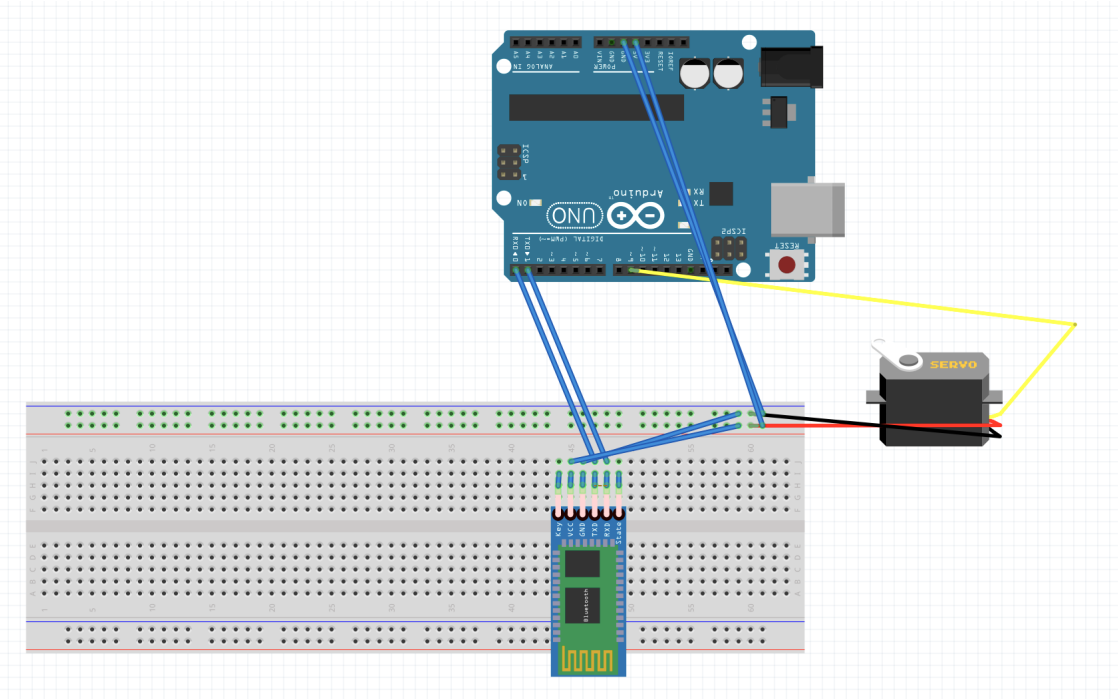

Рисунок 3.4 – Спроектований макет

Для проектування макету системи було використано програму Fritzing. Fritzing -це програмне забезпечення з відкритим вихідним кодом, призначене для розробки схем і друкованих плат для Arduino. Нижче на рисунку схематично зображено метод підключення.

3.3 Реалізація системи

Першим етапом є підключення системи згідно макету з пункту 3.2. Нижче на рисунку відтворено систему пристроя.

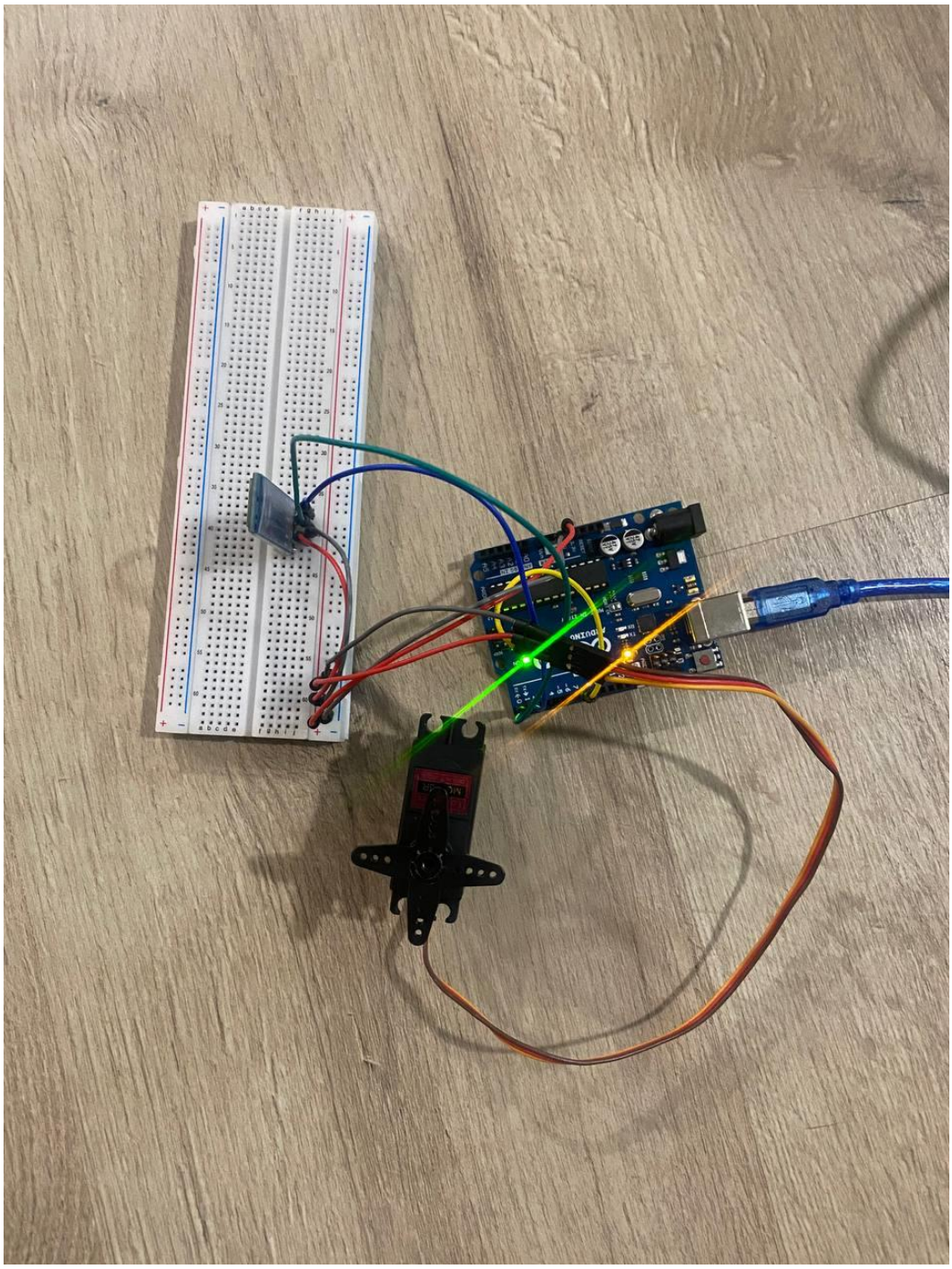

Рисунок 3.5 – Реалізація макета

Для енергопідтримки використовується кабель з роз'ємами типу Am и Bm. На рисунку 3.5 наведено приклад такого кабелю.

Наступним єтапом буде програмування та передачі даних на комп’ютер за допомогою того ж кабелю.

Рисунок 3.5 – Роз'єми типу Am и Bm

Для програмування використовується Arduino IDE середа для розробки на базі Arduino.

Arduino IDE (Integrated Development Environment) - це середовище розробки для програмування мікроконтролерів Arduino. Воно забезпечує зручний інтерфейс для написання, компіляції та завантаження програмного коду на мікроконтролери Arduino.

Основні характеристики Arduino IDE включають:

- Мова програмування: Arduino IDE використовує мову програмування C/C++. Проте, для спрощення роботи з Arduino, бібліотеки та функції Arduino API роблять код більш доступним для початківців.
- Спрощений інтерфейс: Arduino IDE має простий та легкий у використанні інтерфейс, що робить його ідеальним для початківців. У ньому є засоби для написання коду, перегляду результатів компіляції та виводу інформації з мікроконтролера.
- Бібліотеки та приклади: Arduino IDE поставляється з багатьма вбудованими бібліотеками, які спрощують роботу з різними пристроями та модулями. Також у ньому є приклади коду, які можна використовувати як основу для ваших власних проектів.
- Можливість завантаження коду на мікроконтролер: Arduino IDE

забезпечує засоби для компіляції коду та завантаження його на плату Arduino через USB-порт. Процес завантаження коду на мікроконтролер називається "завантаженням прошивки".

- Серійна монітор: В Arduino IDE є інтегрований серійний монітор, який дозволяє вам виводити дані з мікроконтролера та взаємодіяти з ним через COM-порт.
- Підтримка різних моделей Arduino: Arduino IDE підтримує різні моделі та версії плат Arduino, включаючи Arduino Uno, Arduino Nano, Arduino Mega, та інші.
- Розширення та плагіни: Є можливість встановлення розширень та плагінів, які дозволяють розширити функціональність середовища розробки.

На рисунку 3.6 відтворено код для запрограмування системи.

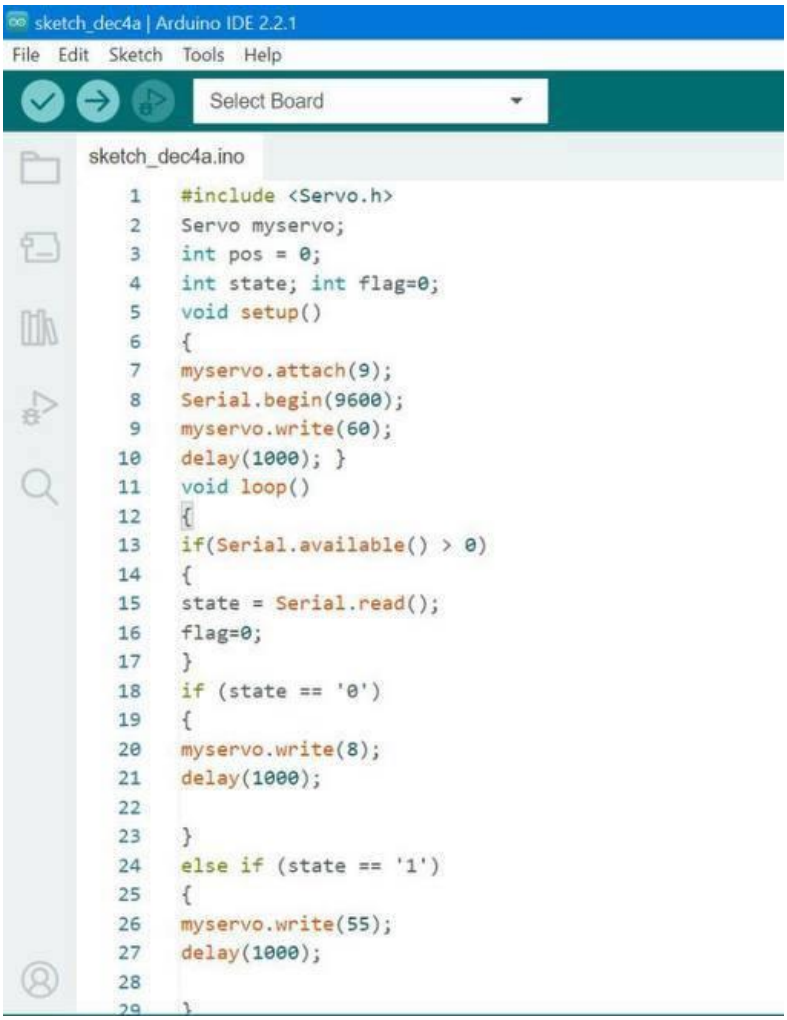

Рисунок 3.6 – Реалізація коду в Arduino IDE

3.4Реалізація додатку для керування системою

Виходячи із поставленої мети роботи, додаток, що розробляється повинна забезпечити:

- Локальне безпечне збереження інформації.
- Робота із сервером.
- Зручний інтерфейс.

3.4.1 Проектування та архітектура

Як описано в розділі 2, додатки повинні плануватися відповідно до моделі MVVM+ Clean Architecture. Для цього всі файли проекту структуруються і групуються відповідно до призначення. Базова структура проекту показана на рисунку 3.6.

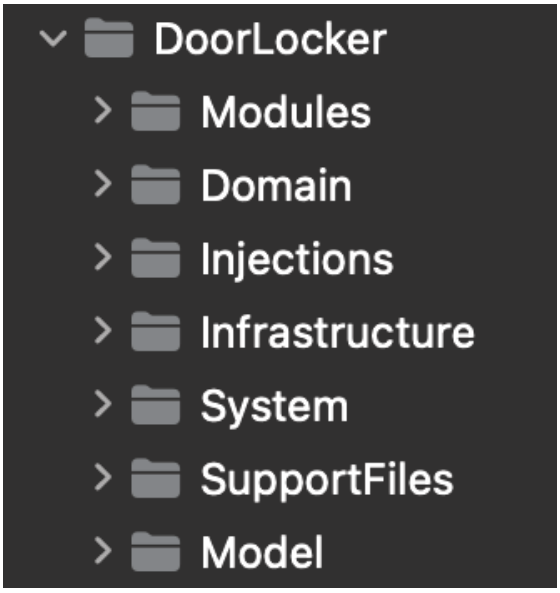

Рисунок 3.6 - Основна структура проекту.

3.4.2 Допоміжні налаштування

SupportFiles містить файли для початкового налаштування проекту. Файли завантажуються в різні середовища проекту за допомогою зовнішніх сервісів Firebase та внутрішньої системи Appe Store Connect. Структура архітектурного блоку SupportFiles показана на рисунку 3.7..

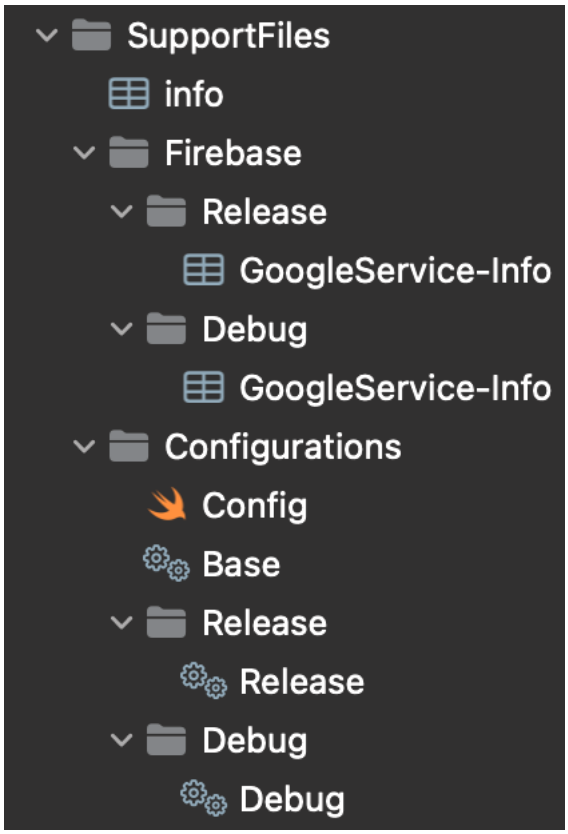

Рисунок 3.7 - Структура SupportFiles.

Проект має кілька робочих середовищ, і конфігурація програми залежить від них.

Середовища робота:

- Debug.
- Release.

Конфігурації додатків можуть сильно відрізнятися в різних середовищах. Наприклад, кінцеві точки серверів, підключення до деяких ресурсів, сервісів тощо.

Розробка користувацького середовища налагодження. Це середовище включає конфігурації ресурсів, системи та технологій і дозволяє налаштовувати експерименти, зовнішні сервіси, бібліотеки та поведінку сервера, не впливаючи на версію програми, що працює на платформі дистрибутива. Середовище випуску - це середовище, у якому програму встановлено на дистрибутивній платформі. Не рекомендується вносити прямі зміни до цього середовища. Нові функції, виправлення помилок у коді та зміни дизайну повинні бути налагоджені та протестовані перед тим, як їх буде інтегровано у середовище випуску. Помилки в середовищі випуску часто є фатальними, оскільки вони роблять додаток непридатним для використання. Щоб внести зміни до середовища, створіть відповідні файли налагодження та випуску, які містять конфігурацію Приклади параметрів конфігурації для середовища випуску показано на Рисунку 3.8.

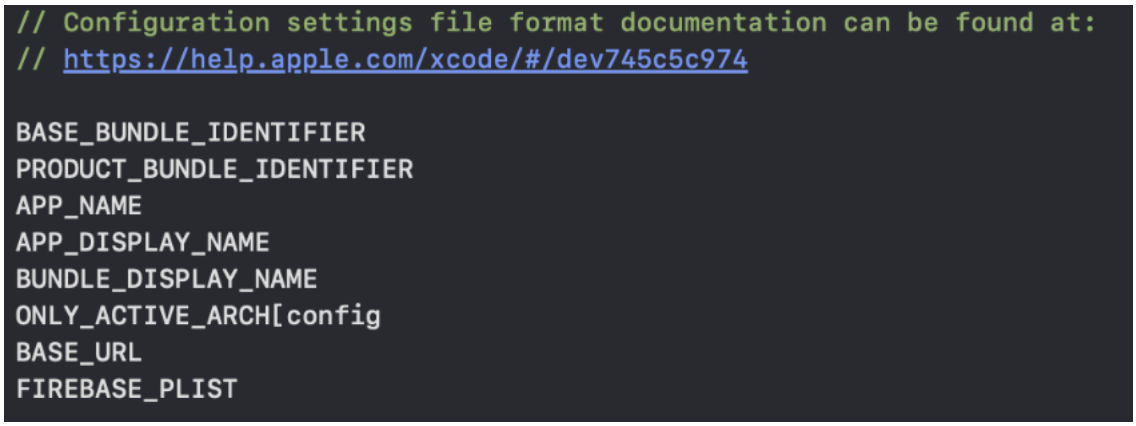

Рисунок 3.8 - Параметри конфігурацій Release.

Після аналізу проблем стабільної поведінки застосунку було використано сервіс Firebase; для налаштування сервісу Firebase було створено файл GoogleService-Info, який містив додаткові налаштування для коректної поведінки застосунку та взаємодії з сервісом. Потім його було розділено на

Debug-файл і Relase-файл кожен з яких містить свій власний GoogleService-Info.

На рисуноку 3.9 показано параметри конфігурації для GoogleService-Info(Release).

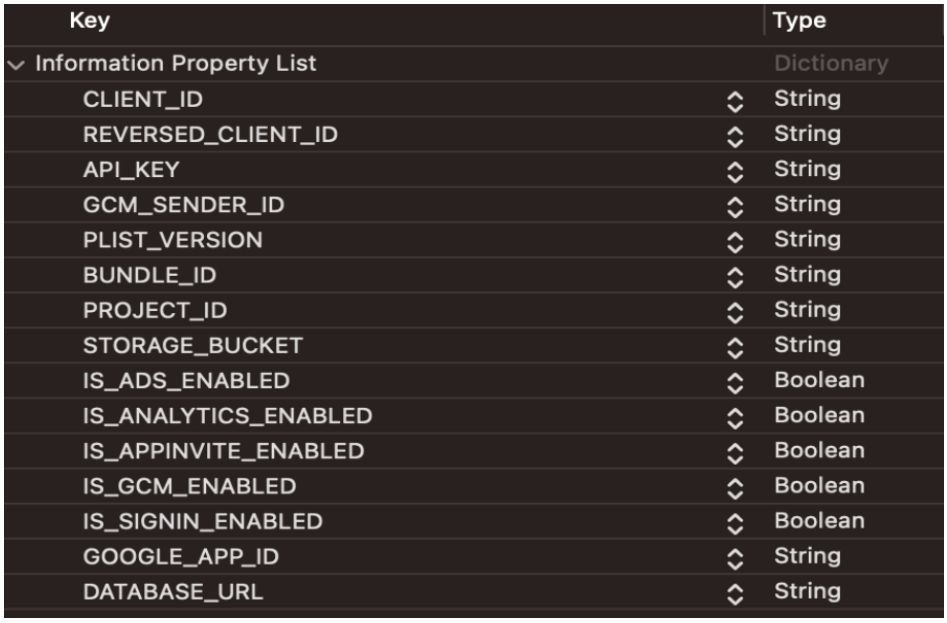

Рисунок 3.9 - Параметри конфігурацій GoogleService-Info(Release).

3.4.2 Інфраструктура

Інфраструктура містить файли, що відповідають за базову конфігурацію користувачів та логіку сповіщень. Приклад структури показано нижче на рисунку 3.10.

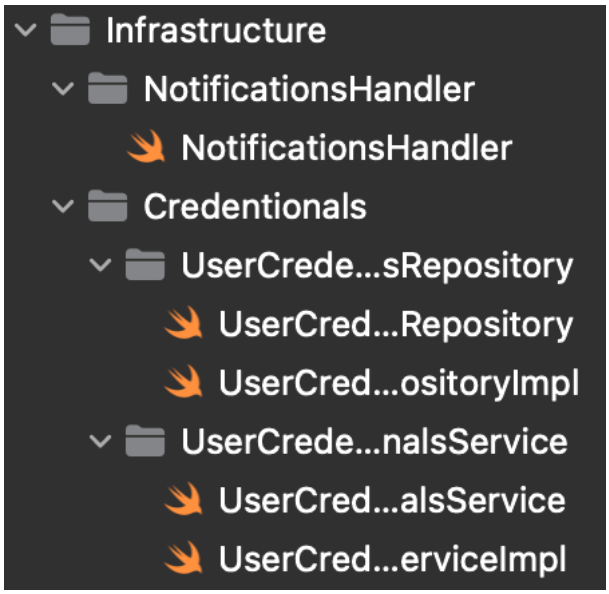

Рисунок 3.10 - Структура Infrastructure.

Папка NotificationsHandler містить логіку обробки сповіщень при зміні даних.

Credentionals - це сервіс для зручного маніпулювання даними користувачів. Основне завдання - зберігати імена користувачів та електронні адреси в зашифрованому вигляді за допомогою фреймворку Keychain і отримувати або видаляти їх у потрібний момент.

3.4.3 Ін'єкції

Ін'єкції - це набір файлів, які ініціалізують усі сутності, класи та їх залежності у проекті за допомогою Container. Цей метод ініціалізації допомагає вирішити проблему залежностей між сутностями. Існує три основні рівні ініціалізації в додатку: Services, Storages, Helpers. Приклад структури Injections показано на рисунку 3.11.

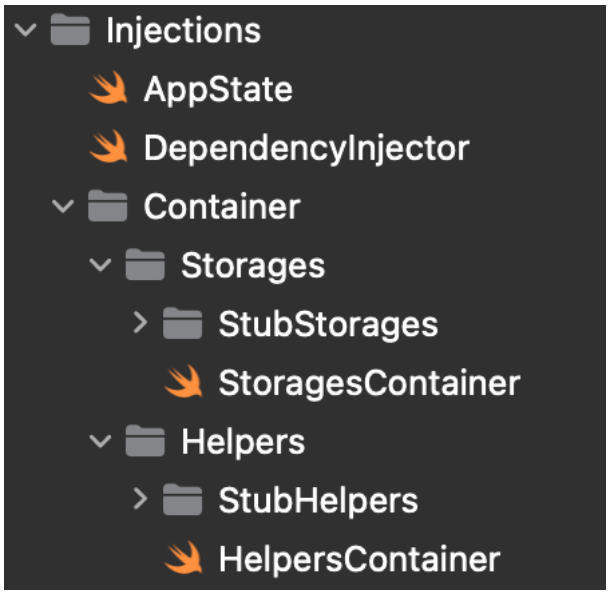

Рисунок 3.11 - Структура Injections.

AppState - це файл, який керує станом програми та реагує на його зміни..

DependencyInjector - Файл, що відповідає за залежності між сутностями та їхніми структурами.

Container містить два шари сутностей, відповідальних за різні аспекти програми.

- Storages - це сутність, яка отримує, зберігає, видаляє та оновлює інформацію в локальних базах даних та інших локальних сховищах.

- Helpers - це сутність, яка може встановлювати зв'язки між модулями та іншими модулями, сервісами.

3.4.4 Система

Система є основним модулем додатків, який запускає програму та запускає всі ресурси, служби та бібліотеки. Приклад структури системи показаний на рисунку 3.12.

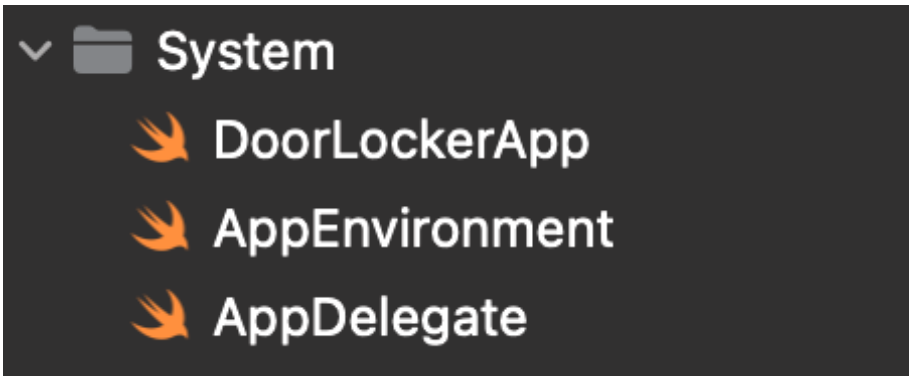

Рисунок 3.12 - Структура System.

DoorLockerApp - це основний файл проекту, який відповідає за запуск програми. Тобто точка входу і підключення програми і системи. Цей файл налаштовує зовнішні служби, вказує ключі API та запускає DIContainer.

AppEnvironment - це файл, в якому налаштовані підключення модулів і ініціалізовані всі рівні і їх залежності.

AppDelegate - це файл системної утиліти, що містить методи обробки стану програми або сповіщень. Він був створений для підтримки старих версій iOS.

3.4.5 Сутності

Domain - це модуль, який відповідає за об'єкти бази даних та об'єкти в додатку. Показано приклад структури Domain на рисунку 3.13.

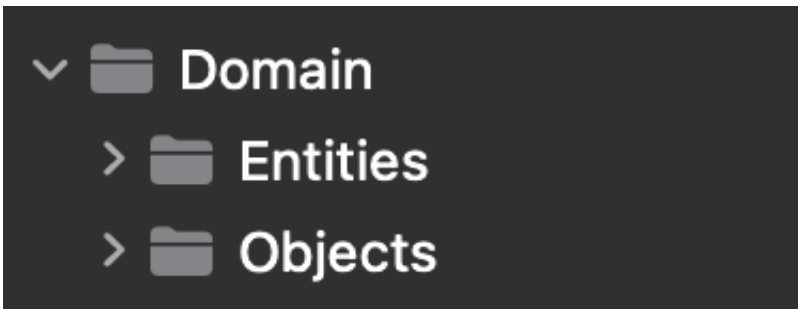

Рисунок 3.13 - Структура Domain.

3.4.6 Головні модулі

Modules являє собою збірку основних прикладних модулів, рівнів і з'єднань. Приклад структури модуля показаний на малюнку 3.14.

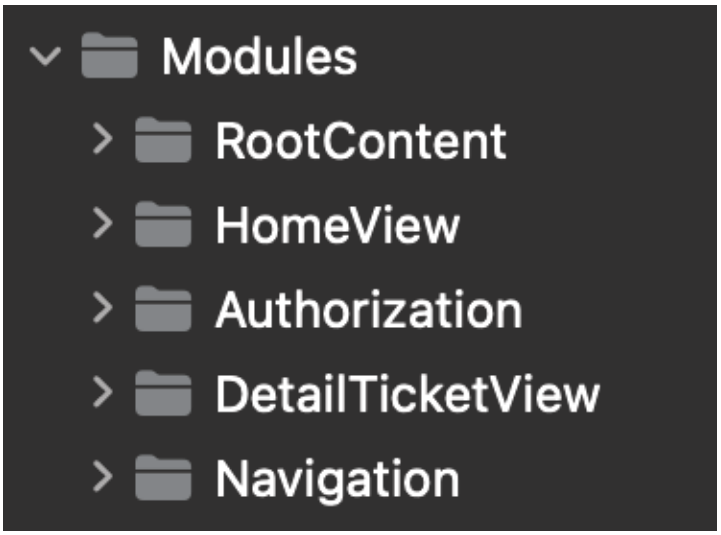

Рисунок 3.14 - Структура Modules.

Navigation містить файли, що відповідають за навігацію між деками. Приклад структури навігації показаний на рисунку 3.15.

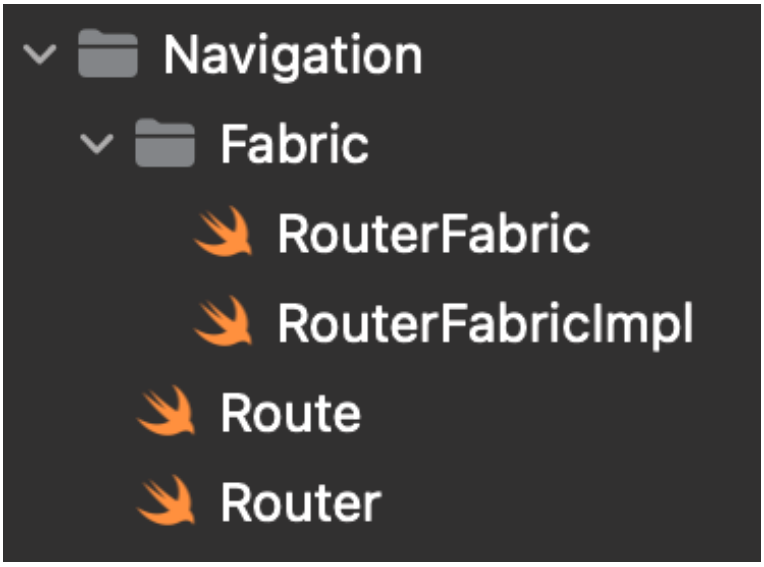

Рисунок 3.15 - Структура Navigation.

- Fabric являє собою конструктор залежностей для модуля.
- Route являє собою маршрути модулів.
- Router містить логіку маршрутів між модулями.

Основні модулі додатку:

- RootContent. Модуль, в кому вирішується, з якого екрану буде функціонувати додаток.

- Authorization. Модуль для авторизації в додатку з обліковим записом користувача.

- HomeView. Модуль для роботи із квитками користувача.

-DetailTicketView. Модуль для роботи із придбаним квитком користувача.

Всі модулі поділяються на три шари. Repository, Service, Presentation.

- Repository - містить логіку роботи модуля з даними. Мета цього рівня- надати модулю інформацію, яка повинна бути отримана з сервера, з локальної бази даних або іншого локального сховища.

- Service - містить модульну бізнес-логіку і передає дані зі Repository в Presentation.

- Presentation - містить параметри проектування та логіку для обробки даних, які повинні бути представлені користувачеві у бажаному форматі.

Приклад одного із модулів наведено на рисунку 3.16.

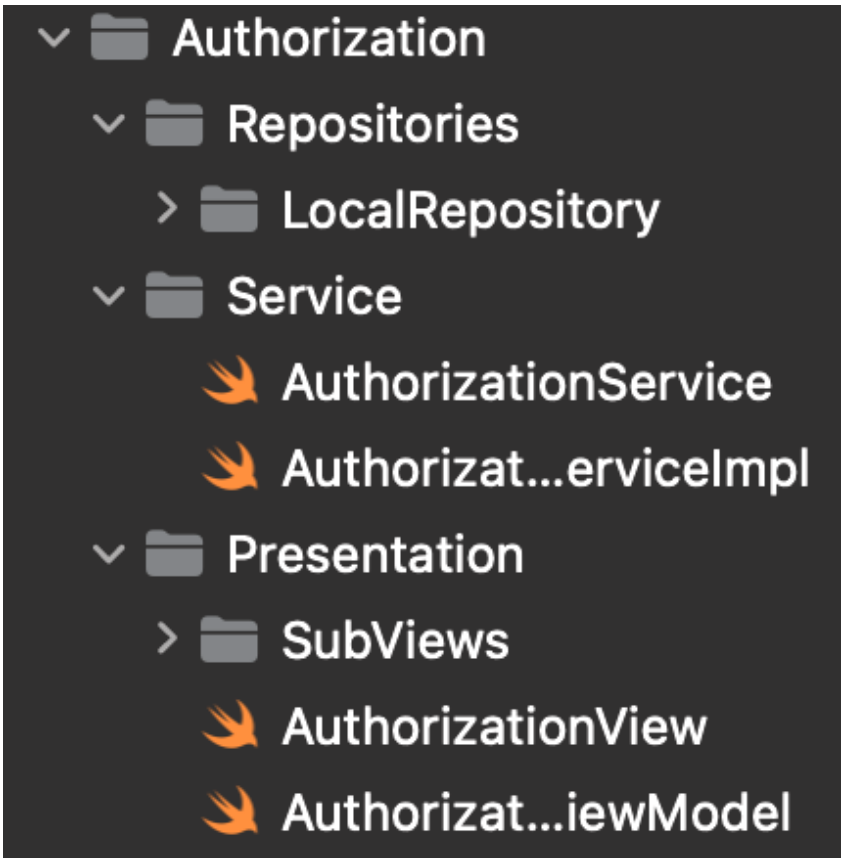

Рисунок 3.16 - Приклад модулю авторизації та його шарів.

3.4.7 Поширені модулі

Common - це модуль, що містить розширення для типів даних, класів та елементів дизайну. Щоб уникнути дублювання коду, і за потреби використовувати кілька елементів, методів знову і знову. Приклад типової конструкції показаний на рисунку 3.17.

Рисунок 3.17 - Структура Common.

3.4.8 Розробка програмного коду для роботи з Bluetooth передачею

Для передачі данних з телефону до системи використовується такий фреймворк, як CoreBluetooth.

CoreBluetooth - це фреймворк для роботи з технологією Bluetooth на платформі iOS та macOS. Використовуючи CoreBluetooth, можна створювати програми, які взаємодіють з Bluetooth-пристроями, такими як датчики, різноманітні модулі, дрони, медичні пристрої, інші мобільні пристрої.

Основні кроки для використання CoreBluetooth в SwiftUI:

- Створення BluetoothManager. Створіть клас або об'єкт, який буде взаємодіяти з CoreBluetooth. Цей об'єкт повинен реалізовувати CBCentralManagerDelegate для керування центральним пристроєм Bluetooth.
- Визначення розгортання UI. Визначте SwiftUI інтерфейс, який буде взаємодіяти з вашим BluetoothManager. Це може бути, наприклад, введення та виведення стану Bluetooth, кнопки для початку/зупинки сканування, відправлення даних тощо.
- Запуск та взаємодія з Bluetooth. Використовуйте методи вашого CBCentralManager для запуску/зупинки сканування, з'єднання з пристроєм та відправлення/прийому даних.

CBCentralManager містить наступні інтерфейси:

- func centralManagerDidUpdateState(_ central: CBCentralManager)

Перевірка чи вімкнений Bluetooth.

- manager.scanForPeripheralsWithServices(withServices:[CBUUID],options: nil)

Пошук девайсів з вімкненим Bluetooth навколо, де withServices: відповідає за можливість обрання окремих девайсів на які ви спримовані також якщо не вказувати нічого то воно буде шукати всі девайси.

- потрir.stopScan()

Зупіняє пошук по Bluetooth для зменшення енергозатратності ця функція.

- centralManager.connect(peripheral: peripheral, options: nil)

Підключення до потрібного нам девайсу, де peripheral: це назва девайсу до якого відбувається підключення.

- peripheral.writeValue(data:data,for:peripheral.services!,type:.withResponse)

Відправка параметрів до під’єднаного пристрою, де data: дані які відправляються, for: кому відправляються та type: яким способом відбувається відправка

3.4.9 Створення структур даних

Питання про правильну структуру даних стало актуальним для запобігання плутанини і заплутування сутностей і їх властивостей. Для вирішення цієї проблеми були створені наступні сутності:

- User. Структура, яка містить дані про користувача.

- TicketList. Структура, яка відповідає за всю кількість білетів які доступні на разі користувачу

- TicketDetails. Структура, яка містить детальну інформацію щодо білету.

- TicketBluetooth. Структура, яка відповідає за поверхневу інформацію що до Bluetooth з’єдняння.

Оскільки ці сутності використовуються практично всіма модулями, правильно додані функції впливають на обсяг коду та запобігають дублюванню їх.

3.4.10 Розробка програмного коду для дизайну

Щоб налаштувати дизайн, у коді використовується структура з відповідною назвою з подання. Подання - це тип, який представляє частину інтерфейсу програми та надає модифікатор, який буде використовуватися для налаштування подання.

Всі структури повинні містити змінну body типу some View. Для відображення дизайну на полотні створюється структура типу Постачальника попереднього перегляду.

У SwiftUI всі екрани мають дуже схожу структуру, що зменшує кількість коду та великі файли. Основна структура конструкції показана на рисунку 3.18.

```
import Foundation
import SwiftUI

struct BaseView: View {

    var body: some View {
        NavigationView {
            GeometryReader { _ in
                EmptyView()
            } //: GeometryReader
        } //: NavigationView
    } //: Body
}

#if DEBUG
struct Base_Previews: PreviewProvider {

    static var previews: some View {
        return BaseView()
    }

}
#endif
```

Рисунок 3.18 – Початковий вид налаштувань екрану

Ви можете використовувати наведені вище основні налаштування для створення нового екрану в додатку.

На всіх екранах використовуються наступні функції:

- Body, що представляє користувальницький дочірній контейнер на екрані тіла.
- NavigationView. Представлення стека відображення, що представляє видимі шляхи в ієрархії навігації.
- GeometryReader, який визначає геометричні розміри як функцію власного розміру та координатного простору.

- PreviewProvider. Попередній перегляд усіх відкритих провайдерів у редакторі вихідного коду. Xcode створює попередній перегляд, використовуючи поточне місце призначення завантаження як підказку для пристрою для перегляду. Наприклад, якщо ви виберете ціль для iOS, яка буде працювати в симуляторі iPhone12Pro Max, Xcode відобразить наступний попередній перегляд.

3.4.11 Створення інтерфейсу

Після аналізу завдання створення простого та зручного інтерфейсу користувача в другому розділі була інтегрований фреймворк SwiftUI, і весь дизайн був розроблений за допомогою механізму коду та Canvas для відображення результатів. На рисунку 3.19 показаний початковий екран.

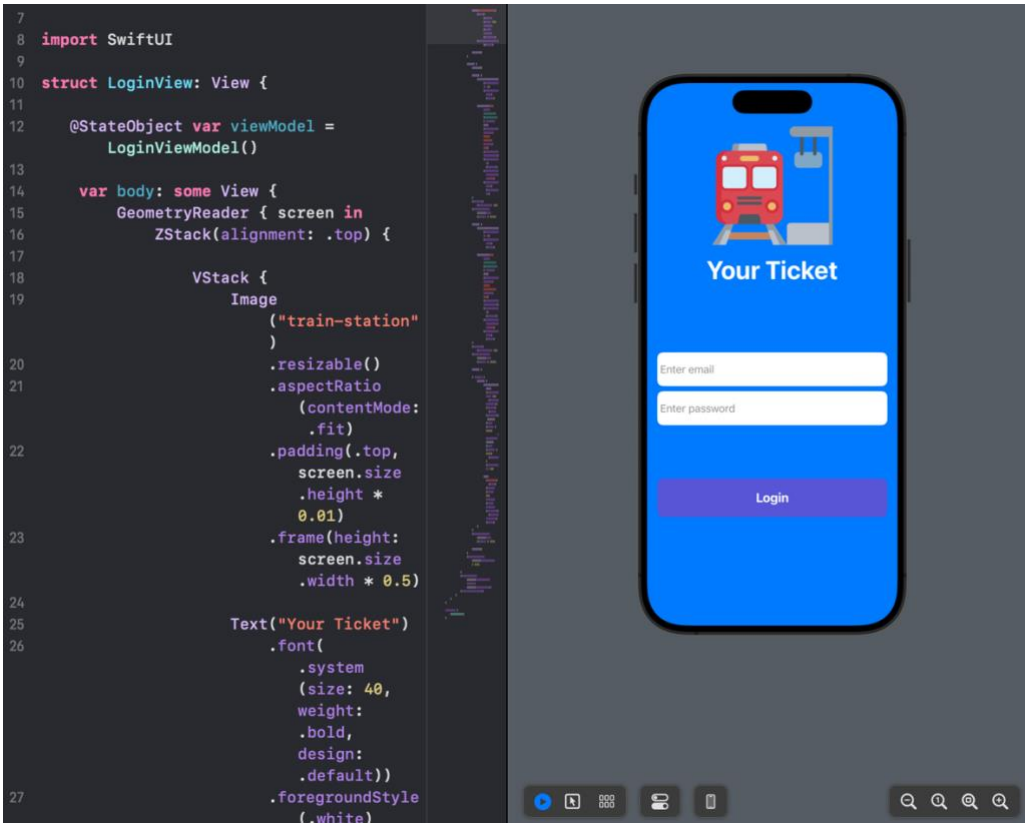

Рисунок 3.19 – Дизайн екрану авторизації

Оновлений механізм Canvas в Xcode дозволяє швидко відображати ваш дизайн практично на всіх доступних моделях iPhone. На рисунку 3.20 показаний дизайн різних моделей телефонів.

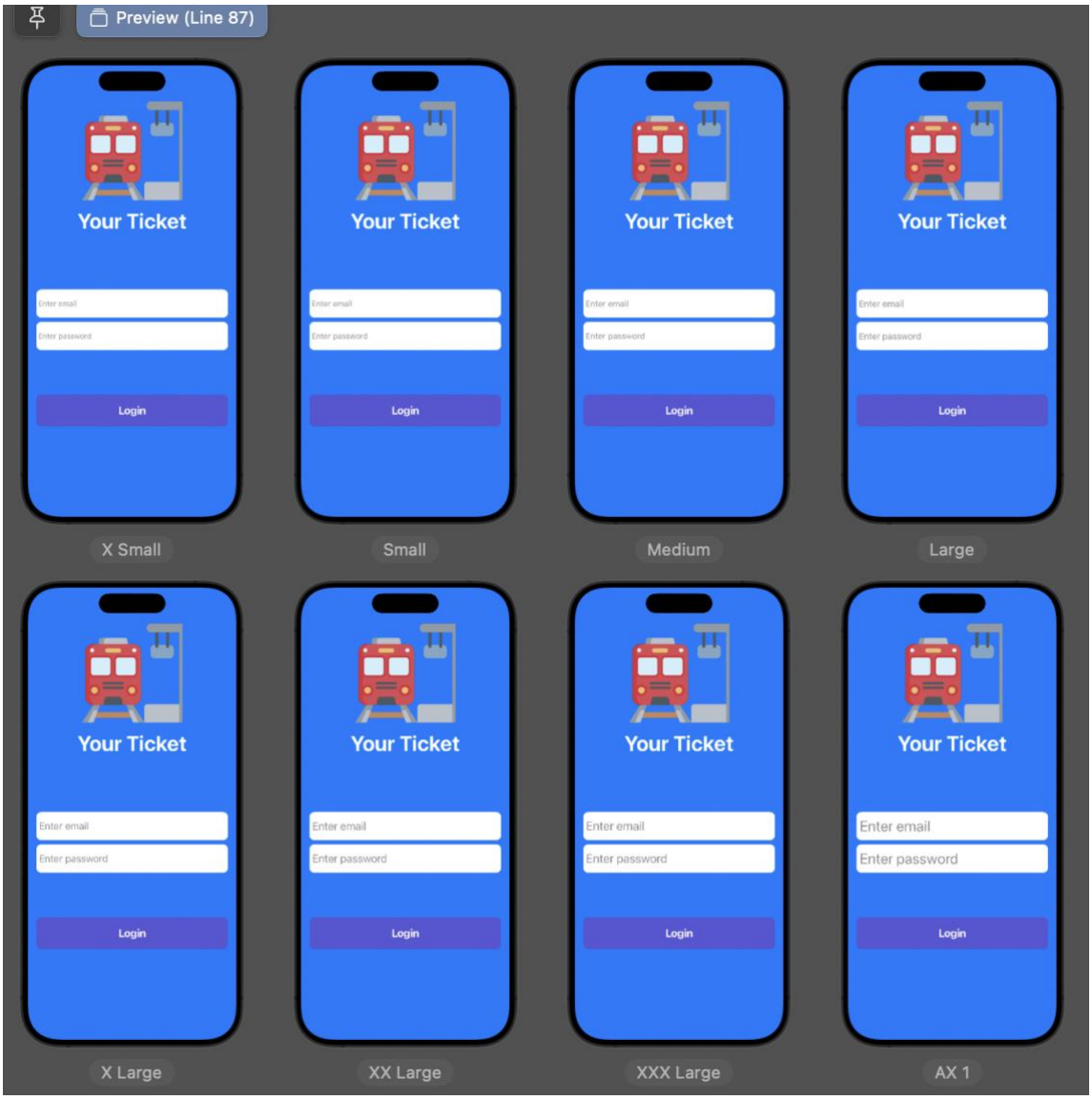

Рисунок 3.20 - Перевірка інтерфейсу на різних моделях.

3.5 Висновок

У цьому розділі розроблено архітектуру та модулі додатків, створено систему контролю доступу. Пояснюються функції, з'єднання і призначення основного, додаткових модулів і під’єднано всі компоненти системи контролю доступу. Для того, щоб система працював відповідно до їх технічних вимог було обрано потрібні пристрої: мікроконтролер, BLE модуль. Для того, щоб додатко працював відповідно до їх технічних вимог, застосовуються необхідні методи та додаткові зовнішні сервіси: робота з мережею, дизайн, Bluetooth, локальне сховище для обробки даних.

4 ТЕСТУВАННЯ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ ТА РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вибір способу тестування програми

Для тестування розробленої програми розглядалось два шляхи тестування:

• Автоматизоване тестування

• Ручне тестування

Автоматизоване тестування – це підхід, який передбачає використання інструмента для пошуку помилок і оцінки продуктивності продукта. Для початку обирається набір інструментів, узгоджується план тестування, виконує план і збирає отримані дані після тестування.

Переваги автоматизованого тестування :

- Запуск тестів в будь який час. Такий вид тестування підходить, коли необхідно проводити тести 24 години на добу

- Повторне використання. Є можливість повторити тестування, як тільки будуть внесені якісь зміни.

- Підвищена швидкість виправлення помилок на ранніх етапах розробки. Наприклад помилки, які пов’язані з втратою пам’яті простіше знайти після після кількох одночасних тестів.

Недоліки автоматизованого тестування :

- Витрати на обслуговування. Тести складно оновлювати і потребують багато часу.

-Інструментальне обмеження для тестування.

- Тестування дрібного функціоналу зменшує ефективність.

Ручне тестування - це базовий підхід до тестування, який виконується лише людиною. Передбачається, що під час ручного тестування фахівець проведе тест незалежно від початку до кінця без використання автоматизованих інструментів.

Переваги ручного тестування:

- Точність. Ручне тестування дозволяє фахівцям більш точно зрозуміти, яким буде кінцевий результат функції для користувача.

- Він більше підходить для складних процедур тестування. Наприклад, протестуйте гру. Щоб протестувати такий продукт, користувачеві необхідно натиснути, зрушити або нахилити пристрій.

- Це дає можливість краще зрозуміти проблему та обрати рішення на концептуальному рівні.

4.2 Вимоги до розробленої системи та програмного забезпечення

Головні етапи тестування системи контролю доступу містять наступні пункти:

- Швидкість прийняття сигналу Bluetooth
- Швидкість реагування рушійної системи на сигнал
- Перевірка на перегрівання системних компонентів при роботі

Головні етапи тестування програмного забезпечення містять наступні пункти:

1) Адаптація дизайну на різних моделях iPhone.

- Відповідність графічних елементів на різних девайсах.
- Перевірка активних елементів на дії користувача.
- Швидкість відгуку графічного інтерфейсу.

2) Ресурси системи.

- Контроль над втратами пам’яті.
- Перевірка на вплив програми на батарею пристрою.
- Навантаження на CPU.

3) Контроль підтримки версій iOS.

- Перевірка роботи програми на версії iOS 16 і новіших

4) Реакція програми на роботу із зовнішніми сервісами.

- Робота з Bluetooth.

4.3 Розробка тест кейсів для програмного забезпечення

Програмне забезпечення було протестовано відповідно до вимог продукту. Щоб забезпечити якість програми, потрібно створити тестовий випадок для тестування основних функцій програми.

Усі тестування та моніторинг ресурсів та графіки проводилися в середовищі Xcode за допомогою стандартних інструментів профілювання(Profile). Приклад вибору інструменту з Xcode показаний на рисунку 4.1.

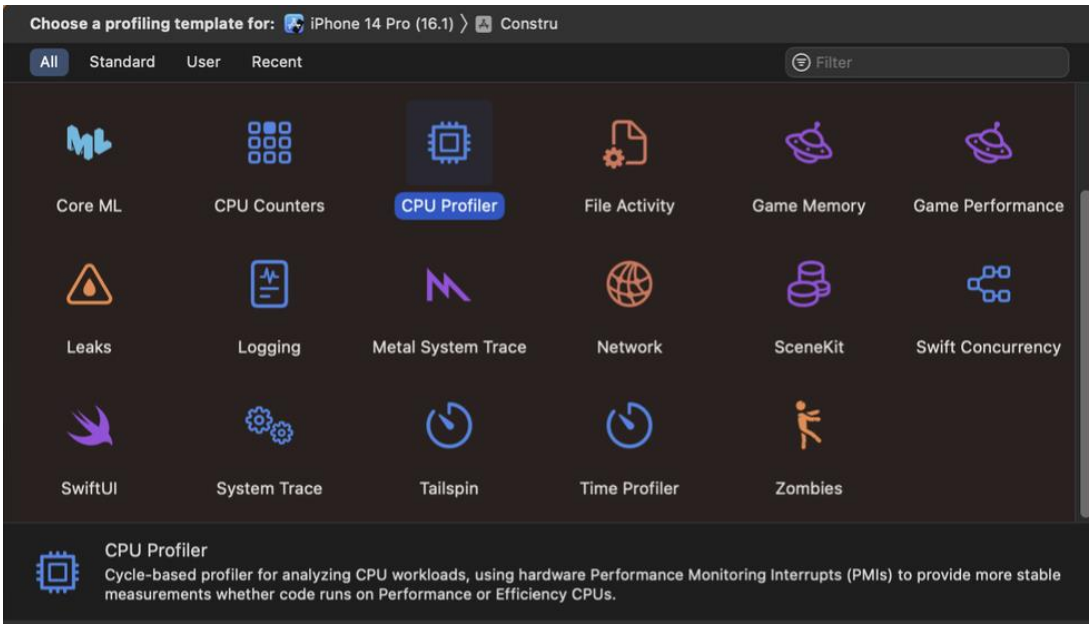

Рисунок 4.1 - Інструменти для тестування Profile.

Основні тест кейси для перевірки роботи програми представлені у таблиці 4.1.

Таблиці 4.1 - Сценарії для перевірки додатку

Назва тесту	Крок	Очікування
Перевірка роботи програми на версіях iOS не нижче 16.	Встановити програму на пристрій із iOS 16 та вище, запустити додаток та перейти на єкран з білетом.	Додаток працює, після скачування плану можна його відкрити. Додаток працює стабільно, збоїв немає.
Перевірка додатку на аварійні завершення роботи.	Відкрити програму, підтягнути білети зайти на білет і активувати ключ після чого згорнути додаток	Додаток працює стабільно без збоїв.
Перевірка графічного інтерфейсу на різних моделях девайсу.	Відкрити програму на iPhone SE та iPhone 13 Pro max.	Дизайн відображається згідно розроблених макетів.
Перевірка підключення Bluetooth.	Запустити додаток з вімкненим Bluetooth, перейти на екран білету, натиснути кнопку відчинити двері, в інформуючому вікні погодитись підключити Bluetooth, Bluetooth вімкнено.	Додаток працює стабільно під час підключення Bluetooth з самого додатку.
Перевірка навантаження на CPU	Запустити програму, підгрузити 10 білетів, підключти Bluetooth, натиснути декілька раз на кнопку відчинити двері.	Додаток працює стабільно без збоїв.

- Перевіряючи, що навантаження на пам'ять пристрою відповідає зазначеним вимогам. Виявив об'єкти в пам'яті, які не були видалені з пам'яті і

займали невелику частину надлишкової пам’яті, декілька мегабайт. Ця проблема була вирішена за допомогою ARC механізму та перетворення сильних посилань у слабкі посилання.

- Перевіряючи, що завантаження CPU відповідає зазначеним вимогам. Використання процесора досягало 3-15%, коли відбувається натискання на кнопку активації Bluetooth ключа . Приклади використовуваних ресурсів показані на рисунках 4.2, 4.3.

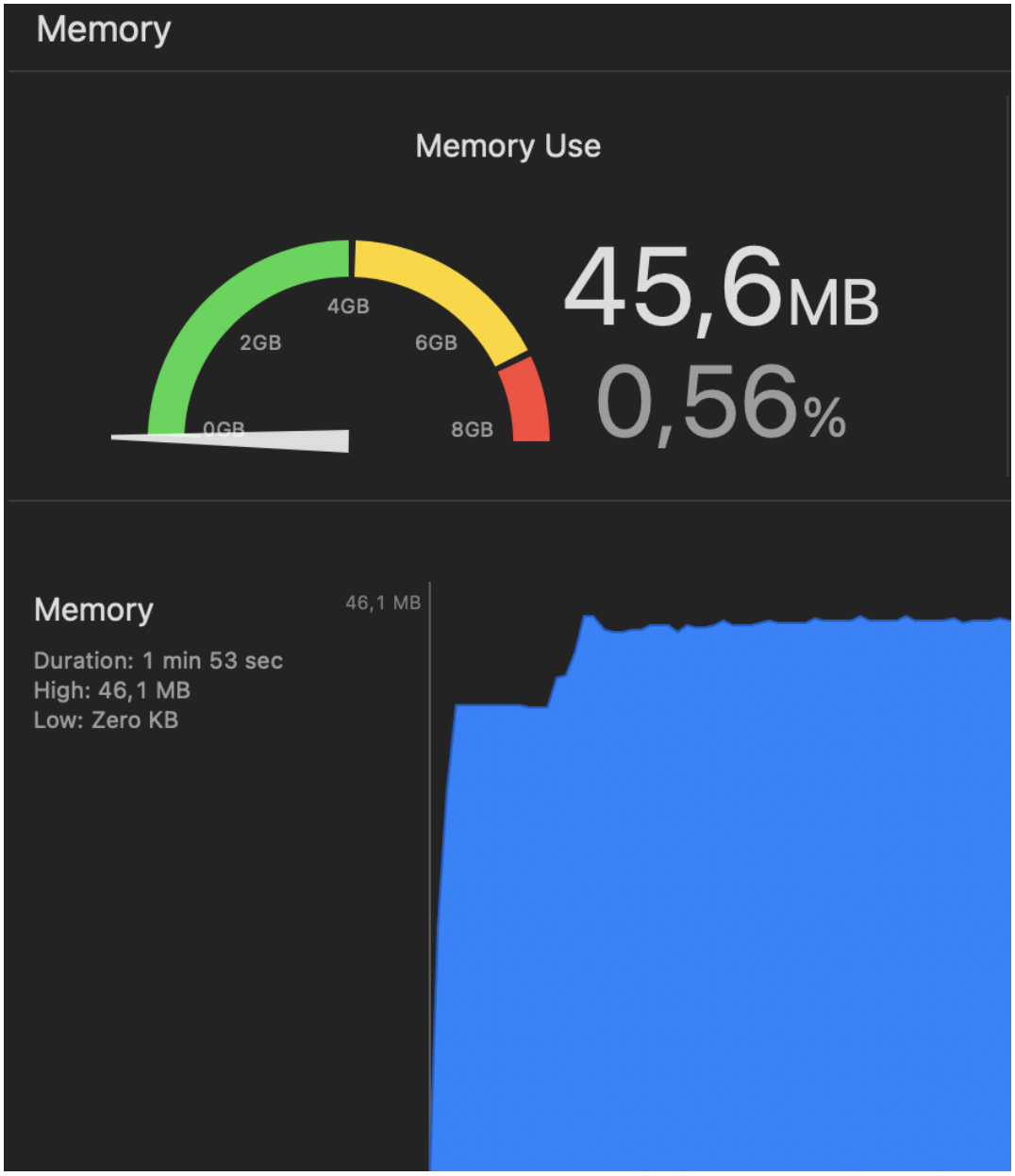

Рисунок 4.2 - Навантаження на пам’ять девайсу.

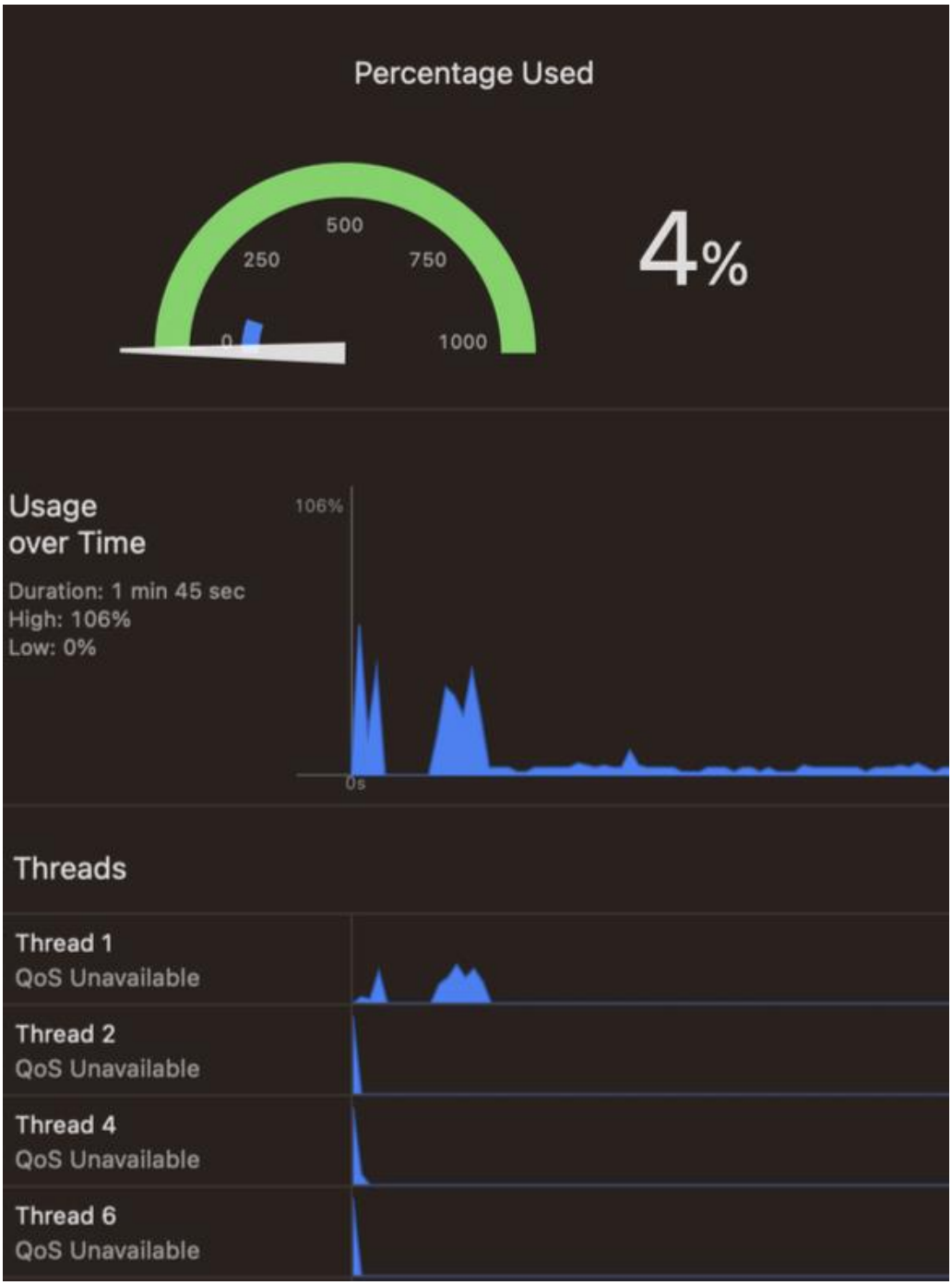

Рисунок 4.3 - Навантаження на CPU.

4.4 Розробка тест кейсів для системи контролю доступу

Таблиці 4.2 - Сценарії для перевірки системи контролю доступу

Назва тесту	Крок	Очікування
Швидкість появи девайсу у покушовій системі Bluetooth	Після вімкнення до живлення системи, телефон одразу знаходить потрібний Bluetooth приймач	Телефон одразу знайшов потріний Bluetooth приймач
Швидкість реагування рушійного механізму після подачі сигналу	Система вімкнена до живлення, телефон знайшов потрібний Bluetooth приймач, телефон подає сигнал на роботу рушійного механізму	Система після подачі сигналу одразу реагує та за декілька секунд рушійний механізм завершує виконання поставленої задачі
Перевірка на перегрівання системних компонентів при роботі	Система вімкнена до живлення, та експлатується в продовж декількох години кожних 5 хвилин віддаючи рушійному механізму сигнал на виконання постевленої задачі	Система відпрацьовує без збоїв, компоненти системи не мають ознак перегрівання

- Перевірка, що швидкість реагування рушійного механізму після подачі сигналу відповідає поставленим вимогам. Після 3 зроблених подач

сигналів середній показник завершення виконання поставленої задачі 2 секунди.

5 ЕКОНОМІЧНИЙ РОЗДІЛ

5.1 Технологічний аудит розробленої системи доступу до купе на основі електронного залізничного квитка

Загальновідомо, що зі зростання популярності мобільних технологій та збільшенням кількості подорожуючих, виникає необхідність у вдосконаленні системи придбання квитків на потяги та полегшенні процесу подорожі. У цьому контексті система контролю доступу до купе при використанні Bluetooth Low Energy на основі електронного залізничного квитка, що знаходитиметься у мобільному додатку набуває особливої актуальності. Цей додаток може не лише спростити та прискорити процес купівлі квитків, але й покращити безпеку та комфорт подорожуючих, а також відповісти на сучасні вимоги з точки зору безконтактності та цифровізації.

Тому перед виконаною нами магістерською кваліфікаційною роботою було поставлено мету: дослідити та розробити ефективну програмно-апаратну систему, спрямовану на оптимізацію процесу придбання та використання квитків для подорожей на потягах.

Для цього нами було: проаналізовано існуючі інструменти для реалізації автоматизованого тестування мобільних додатків; розроблено підхід до автоматизованої купівлі квитків, який дозволятиме ефективно інтегрувати цифровий ключ; розроблено архітектуру, яка забезпечить найшвидший і надійніший спосіб купівлі цифрових квитків з цифровим ключем; розроблено програмне забезпечення для автоматизації придбання квитків на потяг з додатковою можливістю цифрового ключа до місця в купе за допомогою системи Bluetooth; розроблено систему для контролю доступу до купе на базі розробленої програми тощо.

В результаті було розроблено апаратно-програмне забезпечення, яке дає змогу забезпечити пасажирів надійним захистом від можливих злочинних методів доступу до купе завдяки використанню різних цифрових ключів.

Для встановлення комерційного потенціалу розробленої нами програмно-апаратної системи доступу до купе було запрошено 3-х експертів: д.т.н., професора Паламарчука Є.А., к.т.н., доцентів Кулика Я.А. та Овчинникова К.В.

Встановлення комерційного потенціалу розробленої нами системи доступу до купе було здійснено за критеріями, наведеними в таблиці 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання технічного рівня та комерційного потенціалу будь-якої розробки і їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 5.1

Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

Продовження таблиці 5.1

10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Запрошені експерти оцінили розроблену нами програмно-апаратну систему доступу до купе так: (див. таблицю 5.2):

Таблиця 5.2 – Результати технологічного аудиту розробленої нами програмно-апаратної системи доступу до купе (за шкалою оцінювання 0-1-2-3-4)

Критерії	Прізвище, ініціали експертів		
	Паламарчук Є.А.	Кулик Я.А.	Овчинников К.В.
	Бали, що їх виставили експерти:		
1	4	3	3
2	3	3	4
3	4	3	4
4	3	3	4
5	3	3	4
6	3	4	4
7	3	3	3
8	3	3	4
9	4	3	3
10	3	4	4
11	4	4	3
12	3	3	4
Сума балів	$СБ_1 = 40$	$СБ_2 = 39$	$СБ_3 = 44$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{3} СБ_i}{3} = \frac{40+39+44}{3} = \frac{123}{3} = 41,00$		

Встановлення комерційного потенціалу розробленої нами програмно-апаратної системи доступу до купе на основі електронного залізничного квитка було зроблене на основі рекомендацій, наведених в таблиці 5.3. [31]

Таблиця 5.3 – Рівні комерційного потенціалу будь-якої наукової розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 41,0 балів, то це свідчить, що розроблена нами програмно-апаратна

система доступу до купе на основі електронного залізничного квитка має рівень комерційного потенціалу, який вважається «високим».

Це пояснюється тим, що наша розробка розв’язує проблеми, пов'язані зі стандартними методами покупки та використання квитків у потягах, і націлена на створення такого інноваційного рішення, яке покращить якість обслуговування пасажирів відповідно до сучасних вимог транспорту та безконтактних технологій, а також забезпечить клієнтам більш безпечні подорожі.

5.2 Розрахунок витрат на розроблення програмно-апаратної системи доступу до купе на основі електронного залізничного квитка

При виконанні нашої роботи були зроблені певні витрати.

Зокрема:

А). Основна заробітна плата $З_о$ розробників, яка визначається за формулою:

$$З_о = \frac{М}{Т_р} \cdot t \text{ грн}, \qquad (5.1)$$

де М – місячний посадовий оклад розробника, грн; приймемо, що

М = (6700…23000) грн/місяць;

$Т_р$ – число робочих днів в місяці; приймемо $Т_р$ = 25 днів;

t – число днів роботи розробників.

Зроблені розрахунки зведемо до таблиці 5.4:

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник магістерської роботи	21600	684	20 годин	2280
2. Магістрант-студент-виконавець	2000 (беремо 6700)	268	88	23584
3. Консультант з економічної частини	19700	788	1,5 години	197 (при 6-годинному робочому дні)
Загалом				$З_о$ = 26 061 грн

Б). Додаткова заробітна плата $З_д$ розробників розраховується як (10…12)% від величини їх основної заробітної плати, тобто:

$$З_д = \alpha \cdot З_о = (0{,}1...0{,}12) \cdot З_о . \quad (5.2)$$

Приймемо, що $\alpha = 0{,}11$. Тоді для нашого випадку отримаємо:

$$З_д = 0{,}119 \times 26061 = 3101{,}26 \approx 3102 \text{ грн.}$$

В). Нарахування на заробітну плату $НЗП_{зп}$ розробників (дослідників) розраховуються за формулою:

$$НЗП_{зп} = (З_о + З_д) \cdot \frac{\beta}{100}, \quad (5.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. $\beta = 22\%$. Тоді:

$$НЗН_{зп} = (26061 + 3102) \times 0{,}22 = 6415{,}86 \approx 6416 \text{ грн.}$$

Г). Амортизація основних засобів А, які використовувались під час виконання цієї роботи:

$$А = \frac{Ц \cdot Н_а}{100} \cdot \frac{Т}{12} \text{ грн}, \quad (5.4)$$

де Ц – загальна балансова вартість основних засобів, грн;

$Н_а$ – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $Н_а = (2,5...25)\%$;

Т – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 5.5.

Таблиця 5.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютерна техніка, обладнання тощо	38000	25	3,15 (при 80% використанні	1995
2. Приміщення університету, кафедри	20000	2,35	3,15 при 80% використанні	98,7≈ 99
Всього				А = 2094 грн

Д). Витрати на матеріали М розраховуються за формулою:

$$М = \sum_{1}^{n} Н_i \cdot Ц_i \cdot К_i - \sum_{1}^{n} В_i \cdot Ц_в \text{ грн.}, \quad (5.5)$$

де $Н_i$ – витрати матеріалу *i*-го найменування, кг; $Ц_i$ – вартість матеріалу *i*-го най-менування; $К_i$ – коефіцієнт транспортних витрат, $К_i = (1,1…1,15)$; $В_i$ – маса відходів матеріалу *i*-го найменування; $Ц_в$ – ціна відходів матеріалу *i*-го найменування; n – кількість видів матеріалів.

Е). Витрати на комплектуючі К розраховуються за формулою:

$$К = \sum_{1}^{n} Н_i \cdot Ц_i \cdot К_i \text{ грн.}, \quad (5.6)$$

де $Н_i$ – кількість комплектуючих *i*-го виду, шт.; $Ц_i$ – ціна комплектуючих *i*-го виду; $К_i$ – коефіцієнт транспортних витрат, $К_i = (1,1…1,15)$; n – кількість видів комплектуючих.

Під час виконання роботи загальні витрати на матеріали та комплектуючі склали приблизно 800 грн.

Ж). Витрати на силову електроенергію $В_е$ розраховуються за формулою:

$$В_е = \frac{В \cdot П \cdot Ф \cdot К_п}{К_д},$$

(5.7)

де В – вартість 1 кВт-год. електроенергії, в 2023 р. В ≈ 4,5 грн/кВт;

П – установлена потужність обладнання, кВт; П = 0,88 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

Приймемо, що Ф = 298 годин;

$К_п$ – коефіцієнт використання потужності; $К_п < 1 = 0,81$.

$К_д$ – коефіцієнт корисної дії, $К_д = 0,77$.

Тоді витрати на силову електроенергію будуть дорівнювати:

$$В_е = \frac{В \cdot П \cdot Ф \cdot К_п}{К_д} = \frac{4,5 \cdot 0,88 \cdot 298 \cdot 0,81}{0,77} = 1241,38 \approx 1242 \text{ грн.}$$

И). Інші витрати $В_{інш}$ можна прийняти як (50…300)% від основної заробітної плати розробників, тобто:

$$В_{інш} = (0,5....3) \times З_о. \quad (5.8)$$

Для нашого випадку отримаємо:

$$В_{інш} = 0,85 \times 26061 = 22151,85 \approx 22152 \text{ грн.}$$

К). Сума всіх попередніх статей витрат складає витрати на виконання роботи безпосередньо розробником-магістрантом – В.

$$В = 26061 + 3102 + 6416 + 2094 + 800 + 1242 + 22152 = 61867 \text{ грн.}$$

Л). Загальні витрати на розроблення програмно-апаратної системи доступу до купе на основі електронного залізничного квитка $В_{заг}$ становлять:

$$В_{заг} = \frac{В}{\beta}, \quad (5.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання цієї роботи. Можна прийняти, що, $\beta \approx 0{,}90$, оскільки робота практично завершена.

Тоді: $B_{заг} = \frac{61867}{0{,}90} = 68741{,}11$ грн або приблизно 69 тисяч грн.

Тобто прогнозовані загальні витрати на розробку програмно-апаратної системи доступу до купе на основі електронного залізничного квитка становлять приблизно 69 тисяч грн.

5.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки

Економічний ефект від впровадження та можливої комерціалізації розробленої нами програмно-апаратної системи доступу до купе на основі електронного залізничного квитка виявляється у суттєвому покращені якості обслуговування пасажирів відповідно до сучасних вимог транспорту та безконтактних технологій, а також забезпечує пасажирам більш безпечні умови подорожі.

Тому нашу розробку можна реалізовувати на ринку дещо дорожче, ніж аналогічні за функціями розробки.

У 2022 році подібна за функціями програмно-апаратна система (але зі значно гіршими характеристиками) коштувала приблизно 2,5 тисяч грн. Тоді нашу розробку можна буде реалізовувати на ринку в середньому приблизно за 3 тисячі грн або на 500 грн дорожче.

Аналіз ринку також показав, що потенційна кількість користувачів аналогічних систем становить приблизно 500 клієнтів на рік. Разом з тим, кількість таких клієнтів буде постійно зростати, тобто можна очікувати зростання попиту на нашу розробку принаймні протягом 3-х років після її впровадження.

Тобто, якщо наша розробка буде впроваджена з 1 січня 2024 року, то її результати будуть виявлятися протягом 2024-го, 2025-го та 2026-го років.

Прогноз зростання попиту на нашу розробку складає по роках:

а) 2024 р. – приблизно +100 шт. до базового року;

б) 2025 р. – +200 шт. до базового року;

в) 2026 р. – +300 шт. до базового року (оскільки можуть з'явитися ще кращі розробки, зроблені студентами-магістрантами ВНТУ).

Можливе збільшення чистого прибутку $\Delta П_i$, що його може отримати потенційний інвестор від виведення нашої розробки на ринок, становитиме:

$$\Delta П_i = \sum_{1}^{n} (\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\upsilon}{100}), \qquad (5.10)$$

де $\Delta Ц_o$ – покращення основного якісного показника від впровадження результатів нашої розробки у цьому році. Для нашого випадку це є збільшення ціни реалізації нашої розробки $\Delta Ц_o$ = 3,0 – 2,5 = 0,5 тисяч грн;

N – основний кількісний показник, який визначає обсяг діяльності у році до впровадження результатів розробки; N = 8 шт.;

ΔN – покращення основного кількісного показника від впровадження результатів розробки.

Таке покращення становитиме по роках, відповідно: у 2024 році – + 100 шт., у 2025 році + 200 шт., та у 2026 році + 300 шт.;

$Ц_o$ – основний якісний показник (тобто ціна), який визначає обсяг діяльності у році після впровадження результатів розробки, грн; $Ц_o$= 3,0 тисячі грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; для нашого випадку n = 3;

λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0{,}8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати ρ = (0,2...0,5); візьмемо ρ = 0,5;

υ – ставка податку на прибуток. У 2023-25 роках υ = 18%.

Тоді можливе зростання чистого прибутку $\Delta П_1$ для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2024 р.) становитиме:

$$\Delta П_1 = [0,5 \cdot 100 + 3 \cdot 100] \cdot 0,8333 \cdot 0,5 \cdot (1 - \frac{18}{100}) \approx 120 \text{ тис. грн.}$$

Можливе зростання чистого прибутку $\Delta П_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2025) року складе:

$$\Delta П_2 = [0,5 \cdot 100 + 3,0 \cdot 200] \cdot 0,8333 \cdot 0,5 \cdot (1 - \frac{18}{100}) \approx 222 \text{ тис. грн.}$$

Можливе зростання чистого прибутку $\Delta П_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2026) року складе:

$$\Delta П_3 = [0,5 \cdot 100 + 3,0 \cdot 300] \cdot 0,8333 \cdot 0,5 \cdot (1 - \frac{18}{100}) \approx 325 \text{ тис. грн.}$$

Приведена вартість зростання всіх чистих прибутків від можливого впровадження нашої розробки становитиме:

$$ПП = \sum_{1}^{т} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (5.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

т – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку т = 3 роки;

τ – ставка дисконтування. Приймемо τ = 0,10 (10%);

t – період часу від моменту початку розроблення програмно-апаратної системи доступу до купе на основі електронного залізничного квитка до моменту отримання можливих чистих прибутків потенційним інвестором.

Тоді приведена вартість зростання всіх можливих чистих прибутків ПП, що їх може отримати потенційний інвестор від комерціалізації нашої розробки, складе:

$$ПП = \frac{120}{(1+0,1)^2} + \frac{222}{(1+0,1)^3} + \frac{325}{(1+0,1)^4} \approx 99 + 167 + 222 = 488 \text{ тисяч грн.}$$

Теперішня вартість інвестицій PV, що повинні бути вкладені у реалізацію нашої розробки: PV = (1,0…5) × $В_{заг}$.

Для нашого випадку PV = (1,0…5) × 69 = 1,2 × 69 = 82,8 ≈ 83 тисяч грн.

Абсолютний ефект від можливих вкладених в реалізацію нашої розробки інвестицій $Е_{абс}$.

$$Е_{абс} = ПП - PV, \qquad (5.12)$$

де ПП – приведена вартість збільшення всіх чистих прибутків для потенційного інвестора від можливої комерціалізації нашої розробки, грн;

PV – теперішня вартість інвестицій PV = 83 тисяч грн.

Абсолютний ефект від можливого впровадження нашої розробки складе:

$$Е_{абс} = 488 - 83 = 405 \text{ тисяч грн.}$$

Оскільки $Е_{абс} > 0$, то комерціалізація нашої розробки може бути доцільною.

Далі розрахуємо внутрішню дохідність $Е_в$ вкладених інвестицій:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1, \qquad (5.13)$$

де $Е_{абс}$ – абсолютний ефект вкладених інвестицій; $Е_{абс}$ = 405 тис. грн;

PV –теперішня вартість початкових інвестицій PV = 83 тис. грн;

$Т_ж$ – життєвий цикл розробки, роки.

$Т_ж$ = 4 років (2023-й, 2024-й, 2025-й, 2026-й роки)

Для нашого випадку отримаємо:

$$Е_в = \sqrt[4]{1 + \frac{405}{83}} - 1 = \sqrt[4]{1 + 4,8795} - 1 = \sqrt[4]{5,8795} - 1 = 1,557 - 1 = 0,557 = 55,7\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційному інвестору не вигідно буде займатися комерціалізацією нашої розробки.

Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau_{мін} = d + f, \qquad (5.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022-2023 роках в Україні d = (0,10...0,12);

f – показник, що характеризує ризикованість вкладень;

f = (0,1...0,50). Приймемо f = 0,30.

Для нашого випадку отримаємо:

$$\tau_{мін} = 0,12 + 0,30 = 0,42 \text{ або } \tau_{мін} = 42\%.$$

Оскільки величина $E_в = 55,7\% > \tau_{мін} = 42\%$, то потенційний інвестор у принципі може бути зацікавлений у фінансуванні та комерціалізації розробленої нами програмно-апаратної системи доступу до купе на основі електронного залізничного квитка.

Далі розраховуємо термін окупності коштів, вкладених у можливу комерціалізацію розробленої нами програмно-апаратної системи доступу до купе на основі електронного залізничного квитка.

Термін окупності $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_в}. \qquad (5.15)$$

Для нашого випадку термін окупності $T_{ок}$ коштів становитиме:

$$T_{ок} = \frac{1}{0,557} = 1,8 \text{ років} < 3 \text{ років},$$

що свідчить про потенційну доцільність комерціалізації розробленої нами програмно-апаратної системи доступу до купе на основі електронного залізничного квитка.

Далі проведено моделювання залежності величини внутрішньої дохідності вкладених потенційних інвестицій від рівня інфляції в країні.

Якщо рівень інфляції в країні зросте до 20%, то:

$$ПП = \frac{120}{(1+0,2)^2} + \frac{222}{(1+0,2)^3} + \frac{325}{(1+0,2)^4} \approx 83 + 128 + 157 = 368 \text{ тисяч грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$Е_{абс} = 368 – 83 = 285 \text{ тисяч грн.}$$

Внутрішня дохідність $Е_в$ вкладених інвестицій становитиме:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1,$$

де $Е_{абс}$ – абсолютний ефект вкладених інвестицій; $Е_{абс} = 285$ тисяч грн;

PV –теперішня вартість початкових інвестицій PV = 83 тисяч грн.

Для нашого випадку отримаємо:

$$Е_в = \sqrt[4]{1 + \frac{285}{83}} - 1 = \sqrt[4]{1 + 3,4337} - 1 = \sqrt[4]{4,4337} - 1 = 1,475 - 1 = 0,475 = 47,5\%.$$

Оскільки величина $Е_в = 47,5\% > \tau_{мін} = 42\%$, то потенційний інвестор у принципі може бути зацікавлений у фінансуванні та комерціалізації розробленої нами програмно-апаратної системи доступу до купе на основі електронного залізничного квитка.

Якщо рівень інфляції в країні зросте до 320%, то:

$$ПП = \frac{120}{(1+0,3)^2} + \frac{222}{(1+0,3)^3} + \frac{325}{(1+0,3)^4} \approx 71 + 101 + 114 = 286 \text{ тисяч грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$Е_{абс} = 286 – 83 = 203 \text{ тисяч грн.}$$

Внутрішня дохідність $Е_в$ вкладених інвестицій становитиме:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1,$$

де $Е_{абс}$ – абсолютний ефект вкладених інвестицій; $Е_{абс} = 203$ тисяч грн;

PV –теперішня вартість початкових інвестицій PV = 83 тисяч грн.

Для нашого випадку отримаємо:

$$Е_в = \sqrt[4]{1 + \frac{203}{83}} - 1 = \sqrt[4]{1 + 2,4458} - 1 = \sqrt[4]{3,4458} - 1 = 1,362 - 1 = 0,362 = 36,2\%.$$

Оскільки величина $E_в$ = 36,2% < $\tau_{мін}$ = 42%, то потенційний інвестор може бути НЕ зацікавлений у фінансуванні та комерціалізації нашої розробки.

Зроблені розрахунки у вигляді графіків наведено на рис. 5.1.

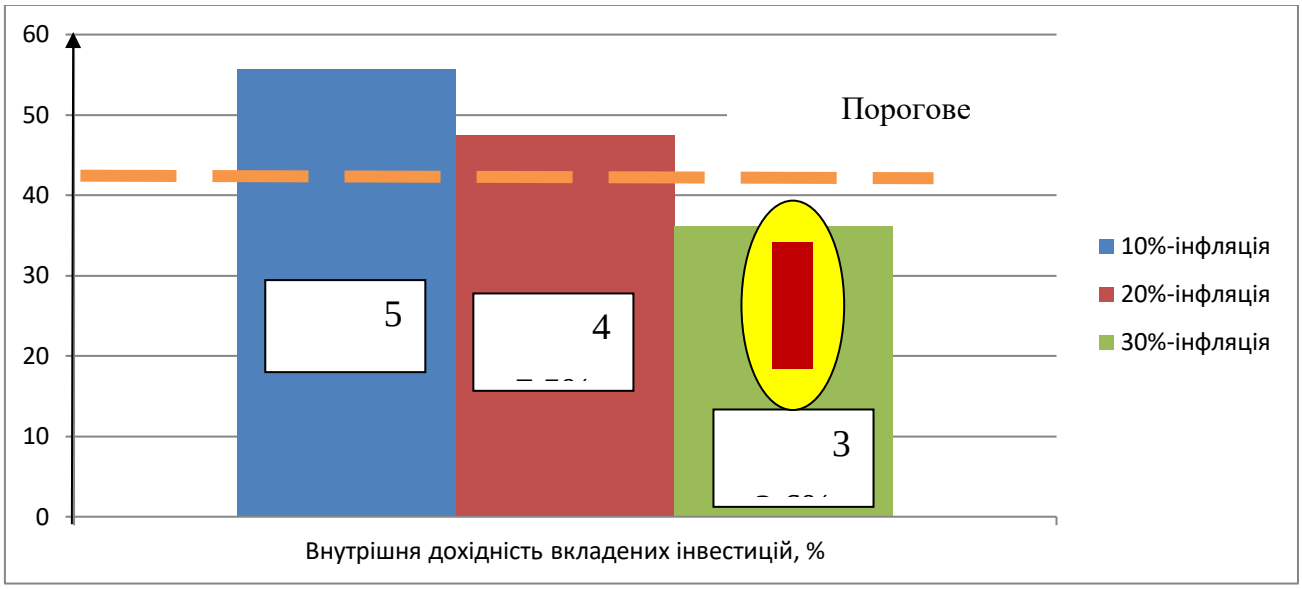

Рисунок 5.1 – Моделювання залежності величини внутрішньої дохідності потенційних інвестицій, вкладених у комерціалізацію розробленої нами про-

грамно-апаратної системи доступу до купе на основі електронного залізнично

го квитка від рівня інфляції в країні (10%. 20% і 30%)

Аналіз діаграм на рис 5.1 показує, що при рівні інфляції в 10% величина внутрішньої дохідності інвестицій становить $E_в$ = 55,7%, що більше порогового значення $\tau_{мін}$ = 42% і тому комерціалізація нашої розробки може бути доцільною. При рівні інфляції в 20% величина внутрішньої дохідності інвестицій, вкладених в комерціалізацію нашої розробки, становить $E_в$ = 47,5%, що також більше порогового значення $\tau_{мін}$ = 42%, і тому комерціалізація нашої розробки потенційним інвестором може бути доцільною. При рівні інфляції в 30% величина внутрішньої дохідності інвестицій, вкладених в комерціалізацію нашої розробки, становить $E_в$ = 36,2%, що менше порогового значення $\tau_{мін}$ = 42%, і тому комерціалізація нашої розробки потенційним інвестором може бути проблематичною.

Остаточне рішення з цього питання потребує проведення додаткових розрахунків (можливо – зниження рівня ризикованості вкладень тощо).

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю:

Таблиця 5.6 – Результати виконаної економічної частини магістерської кваліфікаційної роботи

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку	Не більше 70 тис. грн	69 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження розробки, тисяч грн	Не менше 400 тисяч грн	405 тисяч грн (при 10%-інфляції)	Виконано
3. Внутрішня дохідність інвестицій, %	не менше 42%	55,7% (при 10%-інфляції)	Досягнуто
4. Термін окупності інвестицій, роки	до 3-ти років	1,8 років	Виконано

Таким чином, основні техніко-економічні показники розробленої нами програмно-апаратної системи доступу до купе на основі електронного залізничного квитка, визначені у технічному завданні, виконані.

ВИСНОВКИ

На сьогоднішній день швидко розвиваються цифрові технології, все більше механічних пристроїв можна побачити в цифровому вигляді. Тому люди все більше стають зацікавленими в тому, щоб зробити своє життя зручнішим. Була поставлена задача створити пристрій для відкриття купе за допомогою Bluetooth ключа, який мав реагувати тільки на людину яка має білет саме в її купейний вагон. Пристрій повинен бути мало енергозатратним, також пристрій повинен мати порівняно малу вартість та зручність використання.

В результаті досліджень, проведених в рамках проекту, була проаналізована література, отримані нові практичні і теоретичні навички, використані знання з багатьох предметів.

В першому розділі підводяться підсумки по вивченню теоретичного матеріалу. Розкрита сфера залізничного транспорту на території України, його стан та тенденціїї. Описані основні моменти роботи залізничного транспорту. Також були проаналізовані аналоги додатків, що використовуються сьогодні.

В другому розділі було досліджено, проаналізовано та обґрунтовано вибір платформи для реалізації програмного забезпечення, операційної системи, середовища розробки, мови програмування, архітектурних підходів, технології передачи даних, тип та фреймворк для зберігання даних в локальному сховищі та мікроконтролер для роботи над системою.

У третьому розділі було вдосконалино архітектуру та модулі додатка. Розроблені і обгрунтовані функції, з'єднання і призначення основного і додаткових модулів. Для того, щоб проект працював відповідно до його технічних вимог, застосовуються необхідні методи та додаткові зовнішні сервіси: робота з мережею, дизайн, Bluetooth, локальне сховище для обробки даних.

У четвертому розділі було проведено тестування створеного додатку та системи контролю доступу. Під час якої було визначено, що мінімальний об’єм пам’яті який потрібно мати на телефоні 47 МВ. Після додатку була протестована системи контролю доступу для керування системою з смартфону він повинен знаходитись на відстані від самої системи не більше 60 м. Для використання системи потрібна напруга не меньше 5В. Тестування пройшло задовільно основні кейси система і смартфон виконали задовільно.

Після виконання економічної частини а саме обрахунків було отримано такі результати. Абсолютний ефект від впровадження розробки буде коштувати 405 тис гривень (при 10%-інфляції), а внутрішня дохідність інвестицій 55,7% (при 10%-інфляції), якщо термін окупності інвестицій 1,8 років.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Коцюбняк В.А., Маслій Р.В. Огляд мобільних технологій для побудови системи контролю доступу з використанням цифрового ключа – URL: https://conferences.vntu.edu.ua/index.php/mn/mn2024/author/submission/19388
2. Ukrzaliznytsia – URL: https://www.uz.gov.ua/about/general_information/
3. Звіт по стратегічному дослідженню економічної ефективності та обсягів перевізної роботи по поїздо-дільницях регіональних філій АТ «Укрзалізниця» (на основі даних за 2015 - 2018 роки) – URL: https://www.uz.gov.ua/about/activity/zvit/
4. August Wi-Fi Smart Lock review: Still our favorite – URL: https://www.cnet.com/home/security/august-wi-fi-smart-lock-review/
5. Schlage Sense Smart Deadbolt Review – URL: https://www.allsmartlocks.com/schlage-sense-smart-deadbolt/
6. Yale Assure Lock SL Key Free Touchscreen Deadbolt review: Yale's Assure Lock SL slims down the smart lock – URL: https://www.cnet.com/reviews/yale-assure-lock-sl-key-free-touchscreen-deadbolt-review/
7. Grenadiuk A. Mobile App Marketing And Monetization / A.Grenadiuk.— Semantic Valley LLC, 2014. 151 c.
8. Berney P. Mobile Marketing: Lessons from Global Brand Leaders on How to Make a Success of the Mobile Channel / P.Berney. — Kogan Page, 2019. 224 c.
9. Jagannath S., Debasish D., Lov K., Aneesh K. Application Dev // 18th ICDCIT, 2022. 165c.
10. Tamai S. Technological and business Fundamentals for mobile app dev // Springer Nature Switzerland AG, 2022. 181 c.

11. Офіційний сайт Apple App Store – URL: https://www.apple.com/ua/app-store/
12. Griffiths D. Head First Android Development: A Brain-Friendly Guide 1st Edition / D. Griffiths. — O'Reilly Media, 2015. 734 с.
13. Murphy M. The Busy Coder's Guide to Advanced Android Development M. Murphy. — CommonsWare, LLC, 2011. 630 с.
14. InfoSystem – URL: https://www.hyperlinkinfosystem.com/blog/the-relevance-of-mobile-appdevelopment-to-the-modern-world
15. Quickly Creating, Designing and Utilizing Mobile Apps for Your Business, 2nd Edition / J.McCalister. — CreateSpace Independent Publishing Platform, 2014 — 170 с.
16. Greene R. App Marketing: Top Mobile App Monetization and Promotion Strategies / R.Greene. — Tru Nobilis Pubilishing, 2017. 104 с.
17. Tamai S. Technological and business Fundamentals for mobile app dev // Springer Nature Switzerland AG, 2022. 181 с.
18. Iversen J. Learning Mobile App Development: A Hands-on Guide to Building Apps with iOS and Android, 1st Edition / J.Iversen, M.Eierman. — Addison-Wesley Professional, 2013. 464 с.
19. Goce Armenski, Marjan Gusev. Architecture of Modern e-learning Systems // The 6th International Conference for Informatics and Information Technology, 2008. 38-42– URL:: http://ciit.finki.ukim.mk/data/papers/6CiiT/6CiiT-09.pdf
20. Yiyi S., Practical Application Development // Apress / 2019 – 293 с.
21. Sentry – URL: https://sentry.io/for/mobile/?utm_source=google&utm_medium/
22. Robert C. Martin. Clean Architecture // Prentice Hall, 2017. 432 с.
23. Chris Adamson. iOS 10 SDK Development: Creating iPhone and iPad Apps with Swift / Chris Adamson, Janie Clayton, 2017. 264 с.
24. "Swift, Objectively" – URL: https://www.drdobbs.com/architecture-and-design/swift-objectively/240168424 (Дата звернення: 26.10.2023)

25. Near Field Communication (NFC) Definition – URL: https://www.investopedia.com/terms/n/near-field-communication-nfc.asp
26. Everyone is using Bluetooth Low Energy – should you? – URL: https://www.avsystem.com/blog/linkyfi/bluetooth-low-energy-ble/
27. Understanding Bluetooth Technology – URL: https://www.cisa.gov/news-events/news/understanding-bluetooth-technology
28. Introducing Adafruit Feather – URL: https://learn.adafruit.com/adafruit-feather/feather-specification
29. What is an Arduino:Types, Projects and Applications ? – URL: https://www.raypcb.com/arduino/
30. Arm Cortex – URL: https://www.sciencedirect.com/topics/engineering/arm-cortex
31. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. / Укладачі В.О. Козловський, О.Й. Лесько, В.В.Кавецький. Вінниця, ВНТУ, 2021. 42 с.

ДОДАТКИ

Додаток А

(обов’язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

З А Т В Е Р Д Ж У Ю

Завідувач кафедри АІІТ
д.т.н., професор Олег БІСІКАЛО

_____
(підпис)

«____» _____ 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

СИСТЕМА КОНТРОЛЮ ДОСТУПУ ДО КУПЕ ПРИ ВИКОРИСТАННІ BLUETOOTH LOW ENERGY НА ОСНОВІ ЕЛЕКТРОННОГО ЗАЛІЗНИЧНОГО КВИТКА

08-31.МКР.009.02.000 ТЗ

Керівник к.т.н. доцент кафедри АІІТ

_____ Роман МАСЛІЙ

« _____» _____ 2023 р.

Розробив студент гр. 1АКІТ-22м

_____ Вадим КОЦЮБНЯК

«_____» _____ 2023 р.

Вінниця ВНТУ 2023

1 Назва та галузь застосування.

Система контролю доступу до купе при використанні Bluetooth Low Energy на основі електронного залізничного квитка. Електронна промисловість, автоматизація та приладобудування.

2 Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____ 2023р., та індивідуальне завдання на МКР, затверджене протоколом №__ засідання кафедри АІІТ від «__» _____ 2023р.

Термін виконання робіт:

3 Мета та призначення для проведення робіт.

Метою магістерської кваліфікаційної роботи є оптимізувати та спростити подорожі на потягах для користувачів за допомогою системи контрою доступу.

4 Джерела розробки.

4.1 Lee V. Mobile Applications: Architecture, Design, and Development / V.Lee. — Prentice Hall, 2004 — 368 p.

4.2 Iversen J. Learning Mobile App Development: A Hands-on Guide to Building Apps with iOS and Android, 1st Edition / J.Iversen, M.Eierman. — Addison-Wesley Professional, 2013 — 464 p.

4.3 Arduino Tutorials [Електронний ресурс] – Режим доступу: https://www.arduino.cc/en/Tutorial/HomePage

4.4 Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальностей 126 – «Інформаційні системи та технології», 151 – «Автоматизація та комп'ютерно-інтегровані технології» / Уклад. Р. Н. Квєтний, О. М. Бевз, О. В. Бісікало, Маслі Р.В. – Вінниця: ВНТУ, 2020. – 34 с.

5 Показники позначення.

5.1 технічні дані для системи контролю доступу:

a) Максимальна дальність прийому сигналу – до 60 метрів;
b) Робоча частота модулю приймача – 2,4 ГГц;
c) Напруга живлення – USB: 5В; VCC: 5В; Vin: 7,5В-12В;
d) Споживання в режимі очікування – від 90мкА до 400мкА5.2;

5.2 технічні дані для додатку:

a) Підтримка версії ОС – 16.0+;
b) Розмір додатку – 47 MB;
c) Фреймворк для локального сховища - CoreData;
d) Фреймворк для Bluetooth з’єднання – CoreBluetooth;

6 Економічні показники.

До економічних показників входять:
-витрати на розробку 69 тисяч гривень
-мінімальна дохідність 55,7% при 10% інфляції
-термін окупності не більше 2-х років

7 Стадії розробки.

а) Дослідження предметної області	20.09 – 27.09
б) Обгрунтування методів реалізації та контролю	27.09 – 03.10
в) Розробка структури та схеми	03.10 – 10.10
г) Розробка прогармного забезпечення	10.10 – 20.11
д) Тестування та перевірка припущень	20.11 – 24.11
е) Підтвердженння економічної доцільності	25.11 – 01.12
є) Оформлення матеріалів до захисту МКР	02.12 – 10.12

8 Порядок контролю та приймання.

Рубіжний контроль провести до «01» Листопад 2023 р.
Попередній захист МКР провести «21» » Листопад 2023 р.
Захист МКР провести до «18» » Грудень 2023 р.

Розробив студент групи 1АКІТ-22м _____Вадим КОЦЮБНЯК

Додаток Б

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

СИСТЕМА КОНТРОЛЮ ДОСТУПУ ДО КУПЕ ПРИ ВИКОРИСТАННІ BLUETOOTH LOW ENERGY НА ОСНОВІ ЕЛЕКТРОННОГО ЗАЛІЗНИЧНОГО КВИТКА

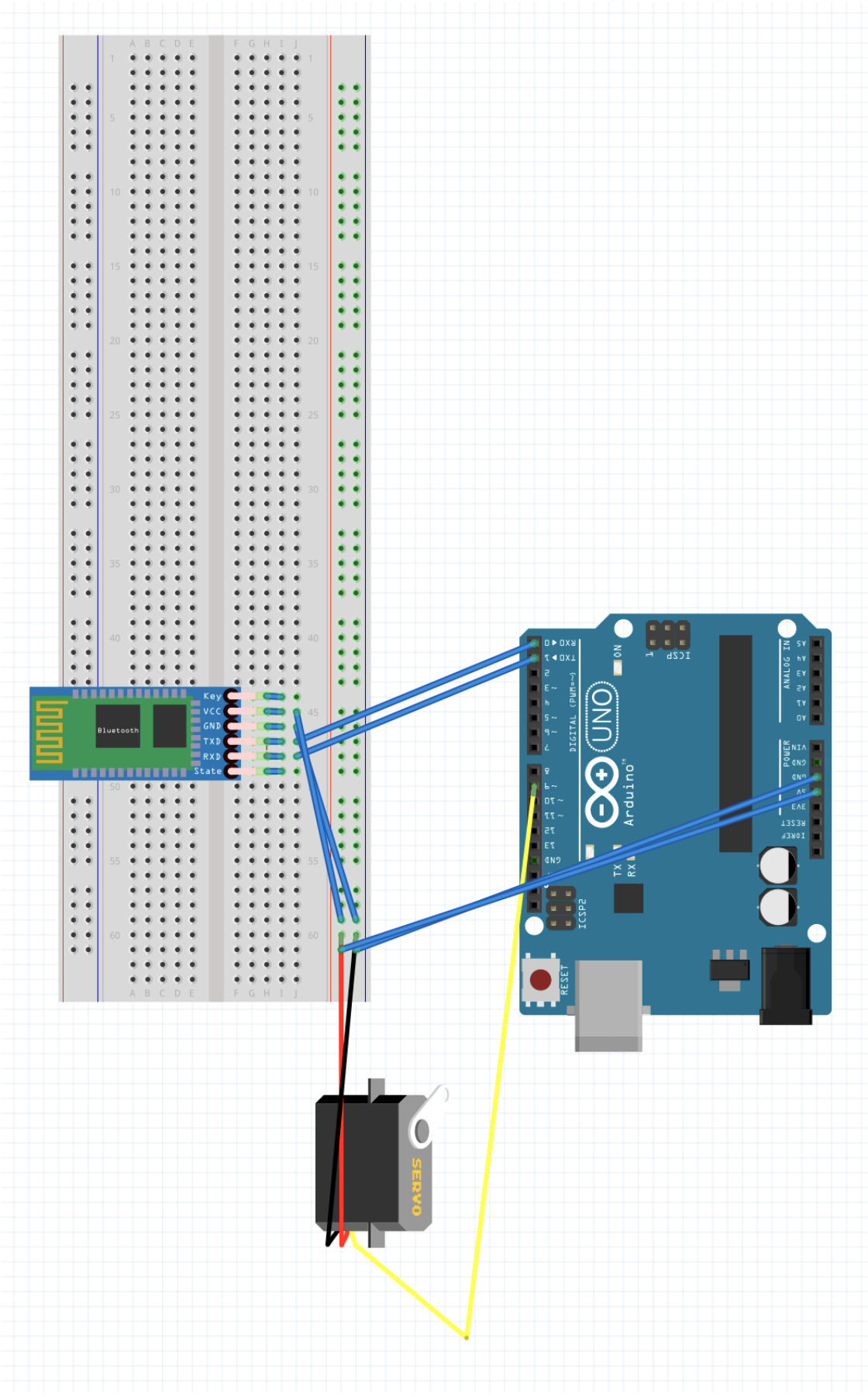

Рисунок Б.1 – Макет системи контролю доступу

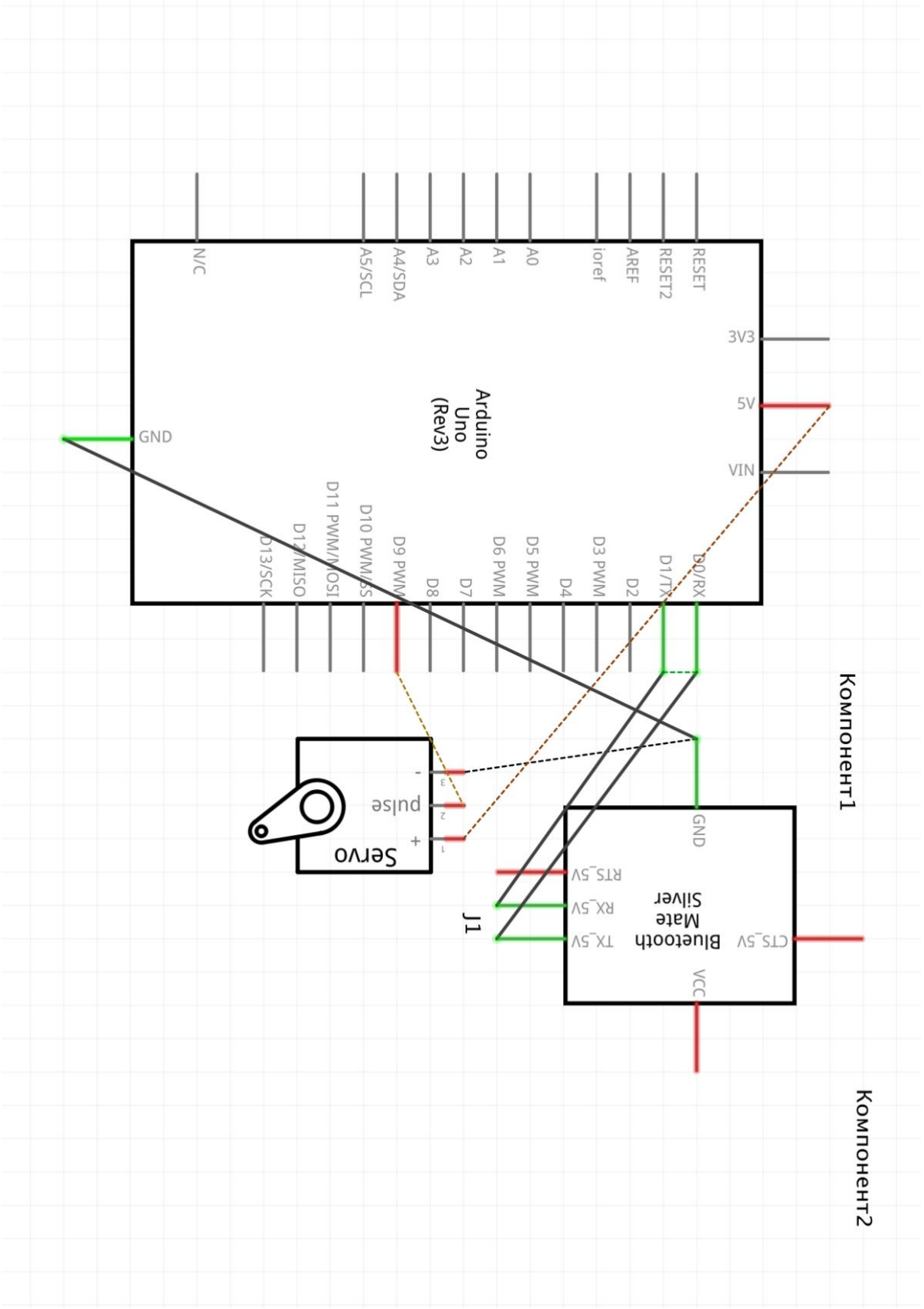

Рисунок Б.2 – Схема системи контролю доступу

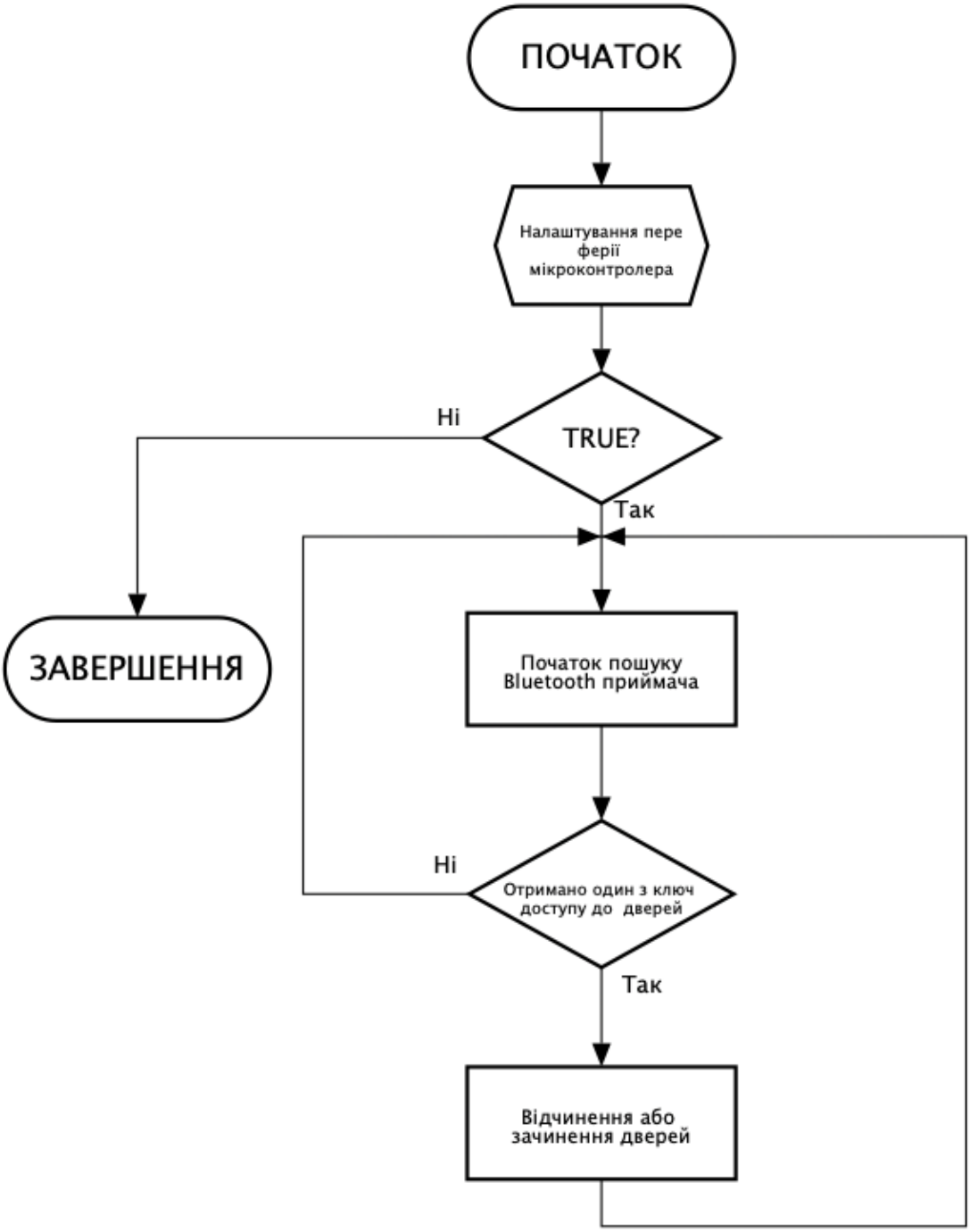

Рисунок Б.3 – Блок-схема роботи системи контролю доступу

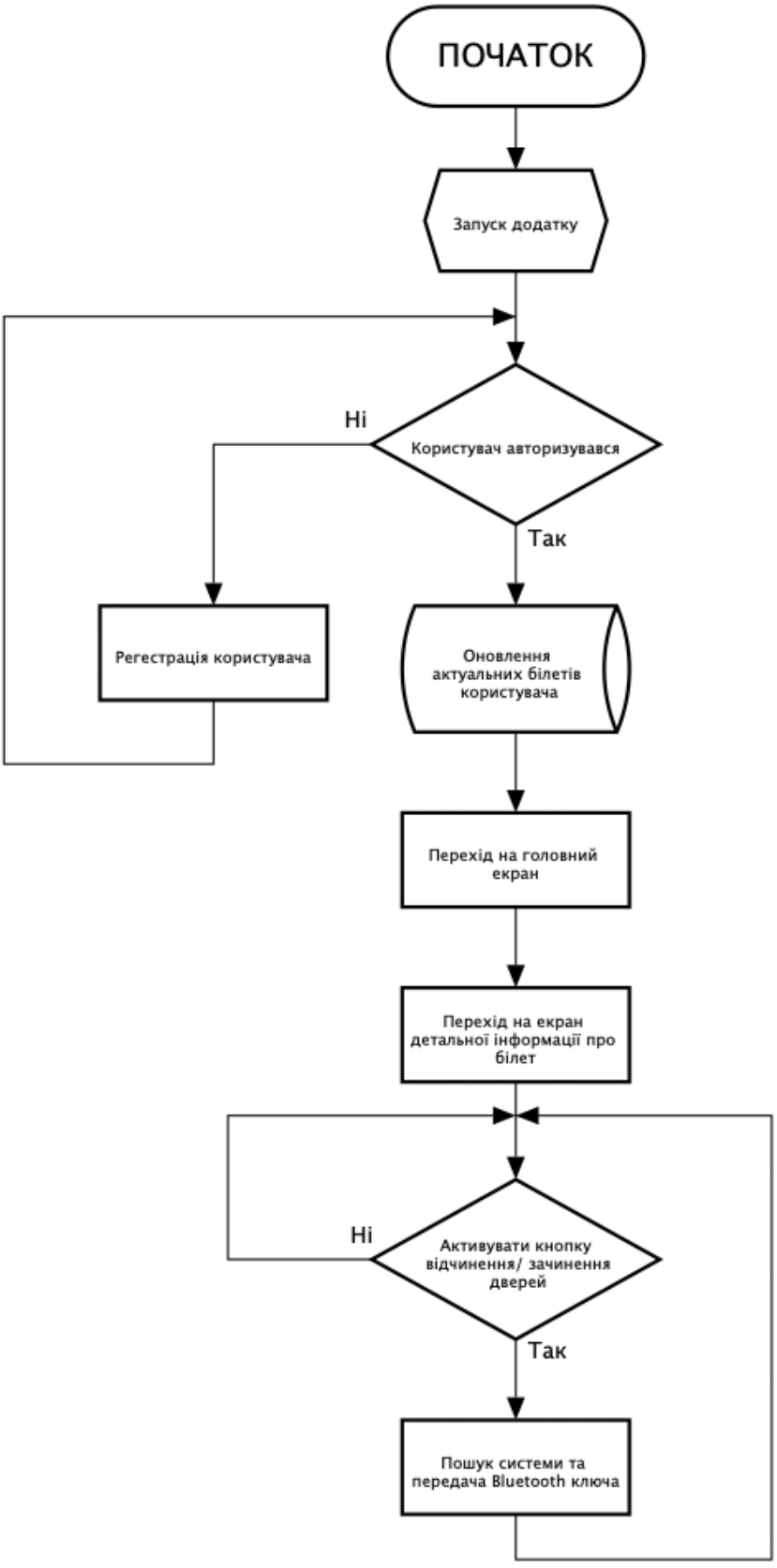

Рисунок Б.4 – Блок-схема роботи додатку для керування системи контролю доступу

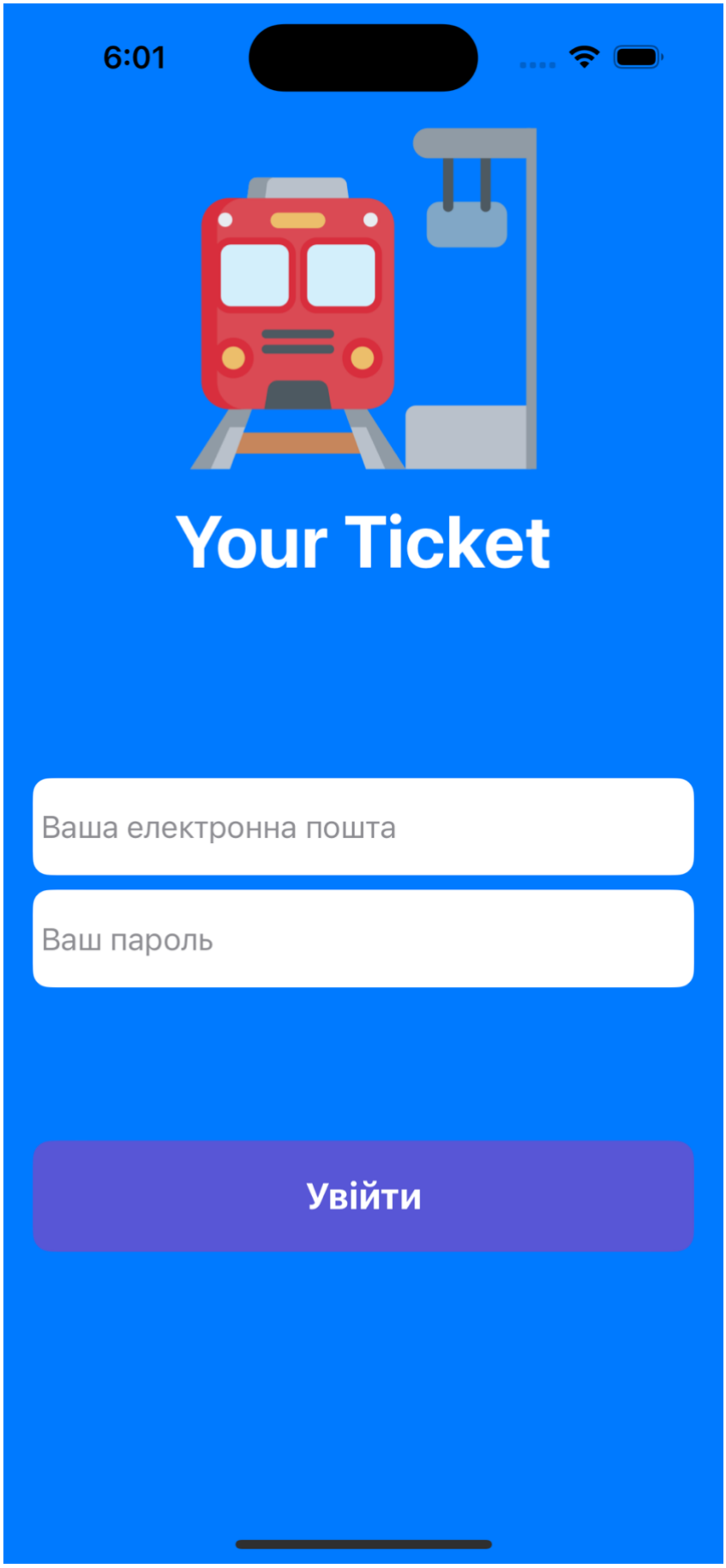

Рисунок Б.5 – Екран авторизації користувача в додатку

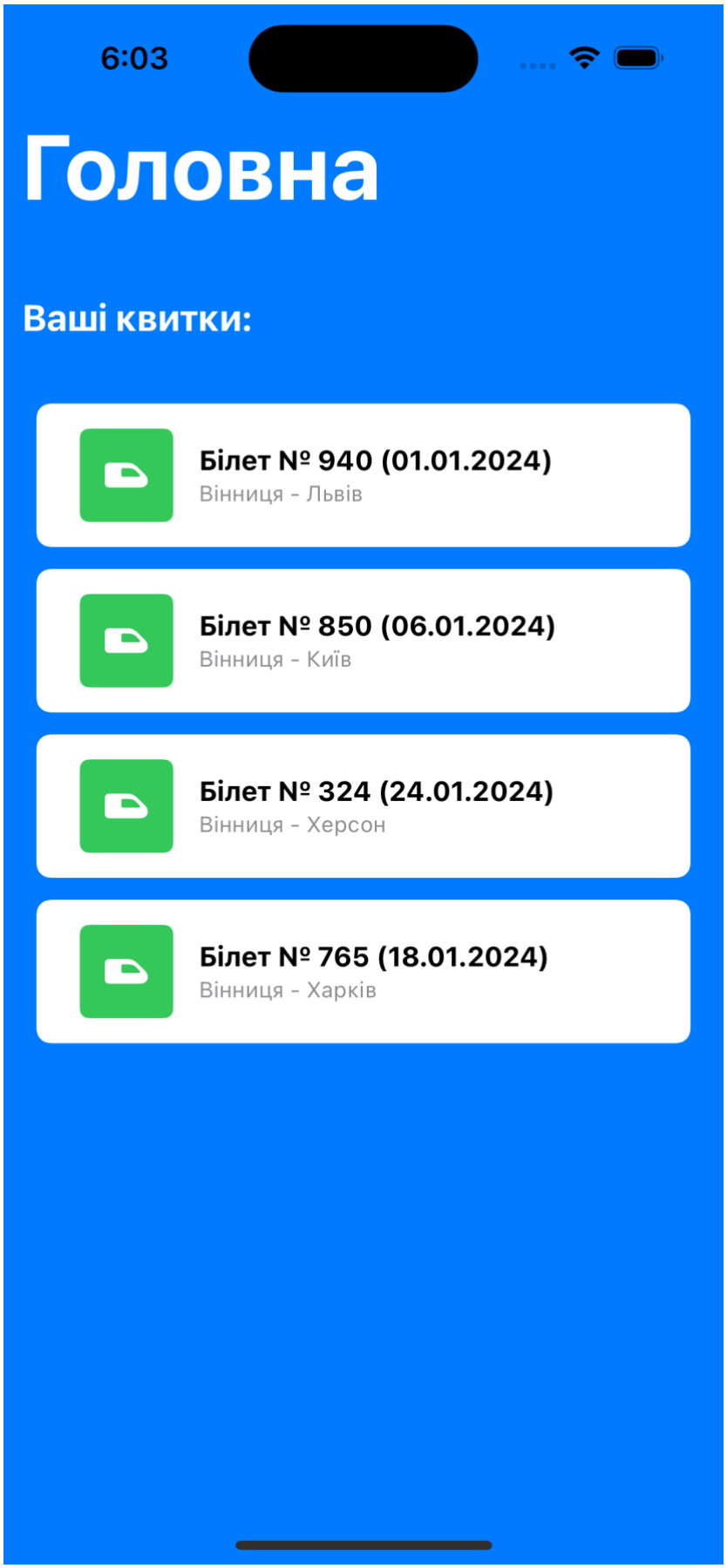

Рисунок Б.6 – Головний екран користувача в додатку

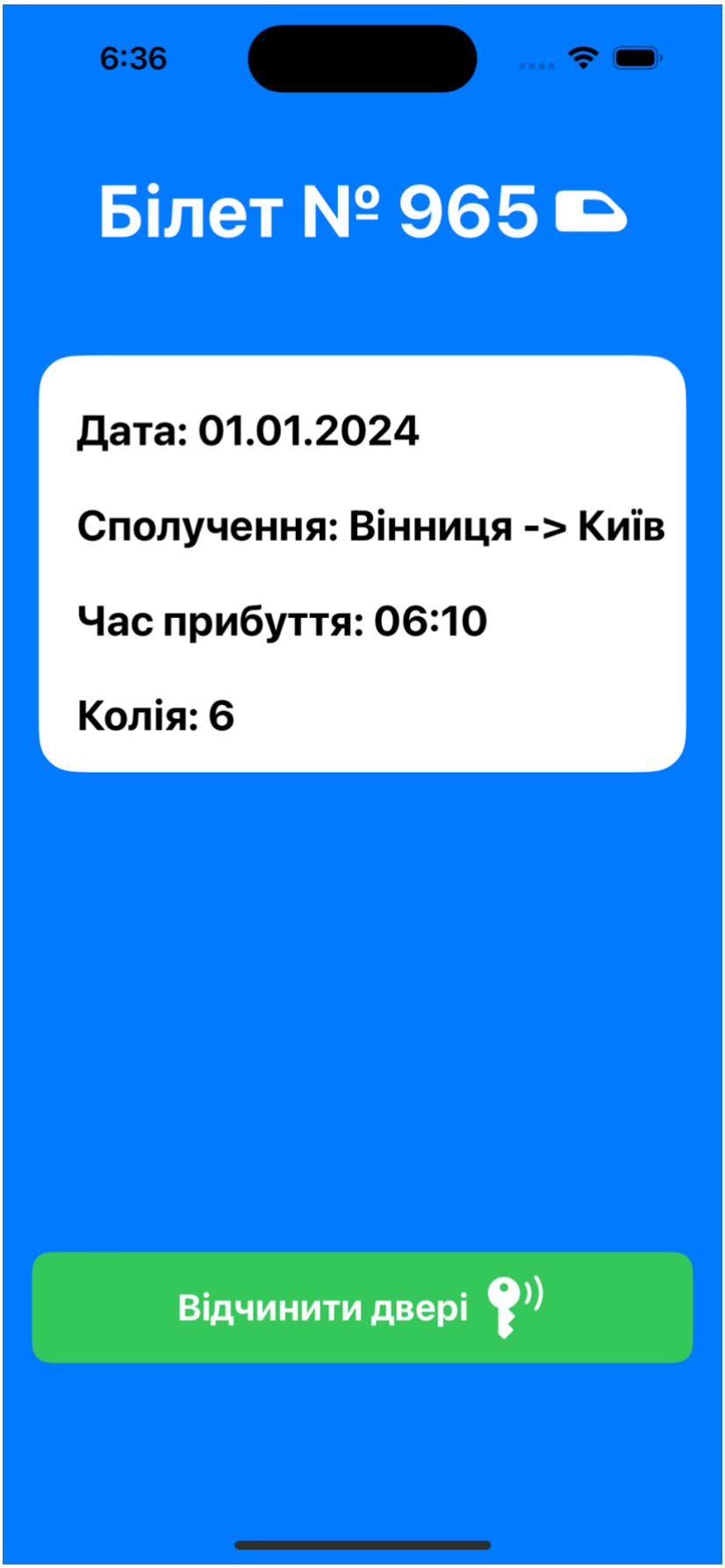

Рисунок Б.7 – Екран білету користувача в додатку

Додаток В

(обов’язковий)

Фрагмент лістингу програми

```
import SwiftUI
import CoreBluetooth

class BluetoothManager: NSObject, ObservableObject, CBCentralManagerDelegate {
    private var centralManager: CBCentralManager!
    private var peripheral: CBPeripheral?

    @Published var isScanning = false
    @Published var isBluetoothEnabled = false
    @Published var receivedData: String?

    override init() {
        super.init()
        centralManager = CBCentralManager(delegate: self, queue: nil)
    }

    func centralManagerDidUpdateState(_ central: CBCentralManager) {
        isBluetoothEnabled = central.state == .poweredOn
    }

    func startScanning() {
        guard isBluetoothEnabled else { return }
        isScanning = true
        centralManager.scanForPeripherals(withServices: nil, options: nil)
    }

    func stopScanning() {
        isScanning = false
        centralManager.stopScan()
    }

    func connect(to peripheral: CBPeripheral) {
        self.peripheral = peripheral
        centralManager.connect(peripheral, options: nil)
    }
```

```
    func sendData(_ data: Data) {
        guard let peripheral = peripheral else { return }
        peripheral.writeValue(data, for: peripheral.services!.first!.characteristics!.first!, type: .withResponse)
    }
}
```

Додаток Г

(обов'язковий)

ПРОТОКОЛ
ПЕРЕВІРКИ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Система контролю доступу до купе при використанні Bluetooth Low Energy на основі електронного залізничного квитка»

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ: кафедра АІІТ, ФІІТА, 1АКІТ-22м
(кафедра, факультет, навчальна група)

Науковий керівник: Маслій Р. В., доц. каф. АІІТ
(прізвище, ініціали, посада)

Показники звіту подібності Unicheck

Оригінальність 95.3% Схожість 4.7%

Аналіз звіту подібності (відмітити потрібне)

■ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень

Особа, відповідальна за перевірку _____ Роман МАСЛІЙ
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи

Автор _____ Вадим КОЦЮБНЯК
(підпис) (прізвище, ініціали)

Керівник роботи _____ Роман МАСЛІЙ
(підпис) (прізвище, ініціали)