**РЕФЕРАТ**

Метою даного дипломного проєкту є розробка програмного застосунку прокладання маршрутів та оптимізації процесу доставлення товарів.

Об'єктом дослідження є розробка програмного застосунку прокладання маршрутів та оптимізації процесу доставлення товарів.

У першому розділу було проведено і наведено детальний аналіз аналогів та було проаналізовано технології їх розробки, було визначено їхні головні переваги і недоліки. Було визначено постановку задачі.

В другому розділі було представлено розробку структури і структури схеми програмного застосунку прокладання маршрутів та оптимізації процесів доставлення товарів. Було проведено обґрунтування вибору забезпечення для створення системи, вибрано та обгрунтовання бази даних, створено блок-схеми алгортиму роботи системи .

В третьому розділі було здійснено реалізацію програмного інтерфейсу системи, реалізовано базу даних, протестовано роботу застосунку.

ABSTRACT

The purpose of this thesis project is to develop a software application for route planning and optimization of the goods delivery process.

The object of research is the development of a software application for route planning and optimization of the goods delivery process.

In the first section, a detailed analysis of analogs was conducted and presented, and their development technologies were analyzed, their main advantages and disadvantages were identified. The problem statement was defined.

In the second section, the development of the structure and schema of the software application for route planning and optimization of goods delivery processes was presented. The choice of software for creating the system was justified, the database was selected and justified, and flowcharts of the system's algorithm were created.

In the third section, the system's software interface was implemented, the database was implemented, and the application was tested.

**РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ПРОКЛАДАННЯ МАРШРУТІВ ТА ОПТИМІЗАЦІЇ ПРОЦЕСУ ДОСТАВЛЕННЯ ТОВАРІВ.**

ЗМІСТ

ВСТУП

1 АНАЛІЗ СИСТЕМИ ПРОКЛАДАННЯ МАРШРУТІВ ТА ОПТИМІЗАЦІЇ ПРОЦЕСУ ДОСТАВЛЕННЯ ТОВАРІВ

1.1 Огляд мобільних додатків для прокладання маршрутів та оптимізації доставлення товарів

1.2 Мобільний додаток доставки їжі «Uber Eats»

1.3 Програмний застосунок доставки «Glovo»

1.4 Мобільний додаток доставки товарів «Нова пошта»

1.5 Постановка задачі

2 РОЗРОБКА СТРУКТУРИ ПРОГРАМНОГО ЗАСТОСУНКУ

2.1 Алгоритм оптимізації маршрутів

2.2 Вибір інструментальних засобів розробки

2.3 Вибір середовища та проектування бази даних

2.4 Оптимізація блок схеми

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ПРОКЛАДАННЯ МАРШРУТУ

3.1 Розробка бази даних

3.2 Інтерфейс програмного застосунку

3.3. Тестування функціоналу

4 ОХОРОНА ПРАЦІ

4.1 Небезпечні і шкідливі фактори при роботі з ПК

4.2 Організація робочого місця з ПК

4.3 Вимоги до санітарно-гігієнічних параметрів робочих місць

4.4 Пожежна безпека

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Розрахунок часу на виготовлення програмного продукту

5.2 Економічне обґрунтування розробки та впровадження проектного рішення

ВИСНОВКИ

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

**ПЕРЕЛІК ОСНОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ**

API - Application Programming Interface (набір готових класів, процедур, функцій, структур і констант, що надаються програмним додатком),

БД – База даних,

JS – Java Script,

QL - structured query language (мова структурованих запитів),

СУБД – система управління базами даних,

GPS - Global Positioning System (Система глобального позиціювання),

ПК – Персональний комп’ютер.

ВСТУП

У сучасному світі, де розвиток технологій займає важливе місце, програмний розвиток є перспективною галуззю. Зростає попит на програмні продукти, які виконують усі необхідні функції, спрощують процеси та ефективно вирішують завдання різної складності.

У межах даної дипломної роботи буде розглянуто тему "Розроблення програмного застосунку прокладання маршрутів та оптимізації процесу". Головною метою даного застосунку є забезпечення користувача інструментом, з допомогою якого буде можливо прокладати маршрути з точною орієнтацією на картах та зменшувати витрати на проїзд.

У процесі розроблення програмного продукту будуть використані сучасні технології та програмні засоби, які дозволяють розроблювати продукти високої якості та ефективності. Результатом проведених досліджень та розробки буде програмний продукт, в якому буде реалізована функціональність, що відповідає завданню даних дипломної роботи.

Розроблення програмного застосунку прокладання маршрутів та оптимізації процесу - це актуальна тема, оскільки день у день люди користуються транспортом та постійно знаходяться в русі. У зв'язку з цим наявність програмного засобу, який забезпечує зручність та якість поїздок, є вельми важливим.

Крім того, розроблення програмного застосунку для прокладання маршрутів та оптимізації процесу допоможе покращити екологічну ситуацію в містах, зменшивши кількість транспорту на дорогах і, відповідно, викидів шкідливих речовин у повітря. Також, цей застосунок може бути корисним для компаній, які займаються логістикою, що дозволить їм ефективніше організовувати доставлення товарів та зменшувати витрати на цю операцію. Кінцевим результатом розробки буде готовий програмний продукт, який буде забезпечувати простоту використання та високу якість роботи, що своєю чергою зробить його популярним серед різних категорій користувачів.

Отже, розроблення програмного застосунку прокладання маршрутів та оптимізації процесу є важливою та актуальною задачею, яка може забезпечити зручність та ефективність поїздок, а також покращити екологічну ситуацію в містах та зменшити витрати на логістику. У даній роботі будуть використані сучасні технології та програмні засоби для розробки програмного продукту високої якості та ефективності. Результатом роботи буде готовий продукт, який буде корисним для широкого кола користувачів та допоможе забезпечити їхні потреби у прокладанні маршрутів та оптимізації процесу.

Для досягнення поставленої мети, у роботі буде використано декілька методів дослідження та розробки. Починаючи з аналізу систем прокладання маршрутів та оптимізацій процесів доставлення товарів, а саме які виконують подібні функції, з метою виявлення їх переваг та недоліків. Цей етап дозволить врахувати досвід попередніх розробок та уникнути можливих помилок у процесі створення власного застосунку.

На основі результатів аналізу будуть розроблені архітектура програми та база даних, які забезпечать оптимальну роботу продукту. Також, буде створено інтерфейс користувача, який буде зручним та інтуїтивно зрозумілим для різних категорій користувачів.

Після завершення розробки програми, буде проведене тестування її функціональності та продуктивності з метою виявлення можливих помилок та недоліків, які потрібно не допустити в розробці програмного застосунку.

Загалом, дана дипломна робота має на меті розробити програмний продукт, який забезпечить користувачів зручним та ефективним інструментом для прокладання маршрутів та оптимізації процесу. Результатом роботи буде програмний застосунок.

**1 АНАЛІЗ СИСТЕМ ПРОКЛАДАННЯ МАРШРУТІВ ТА ОПТИМІЗАЦІЇ ПРОЦЕСУ ДОСТАВЛЕННЯ ТОВАРІВ**

**1.1 Огляд мобільних додатків для прокладання маршрутів та оптимізації доставлення товарів**

На сьогоднішній день існує багато мобільних додатків, які допомагають прокладати маршрути та оптимізувати процес доставки товарів. Такі додатки використовуються в різних галузях, таких як доставка їжі, логістика, магазини тощо. Деякі з них використовують алгоритми штучного інтелекту та машинного навчання для оптимізації маршрутів та зниження часу доставки товарів.

Також ці додатки допомагають організовувати процес доставки більш ефективно, що зменшує витрати на транспортування товарів і збільшує задоволення клієнтів. Компанія отримує інформацію від користувача зображено на рисунку 1.1, а потім здійснює доставку.

|  |
| --- |
| Оформлення замовлення |
| ↓ |
| Приймання інформації |
| ↓ |
| Збереження й опрацювання замовлення |
| ↓ |
| Оформлення доставки |
| ↓ |
| Отримання товару |

Рисунок.1.1 Отримання доставки.

Одним з таких додатків є Google Maps, який дозволяє користувачам прокладати маршрути та обирати найбільш ефективний спосіб доставки товарів. Цей додаток також використовує алгоритми машинного навчання, щоб рекомендувати оптимальні маршрути з урахуванням трафіку і інших факторів. Інший додаток - Waze - також пропонує навігацію на основі реальних даних про дорожні умови, що дозволяє водіям уникати заторів і заощаджувати час на доставку.

Додатки для доставки їжі, такі як:

* Uber Eats.
* Grubhub.
* DoorDash.
* Glovo.

Вони використовують подібні алгоритми, щоб забезпечити швидку та ефективну доставку їжі клієнтам. Крім того, вони надають можливість користувачам відстежувати статус своєї доставки та оцінювати якість послуг.

У галузі логістики додатки, такі як:

* Descartes Systems Group.
* Route4Me.
* Нова пошта.

Ці додатки допомагають цим компаніям оптимізувати маршрути транспортування товарів та знизити витрати на доставку. Крім того, у багатьох магазинах, таких як Walmart та Amazon, додатки для прокладання маршрутів та оптимізації доставки допомагають організовувати логістичні процеси та забезпечувати швидку доставку товарів клієнтам.

Оптимізація доставки за допомогою мобільного застосунку може бути дуже ефективним способом покращення процесу доставки. За допомогою цих пунктів і створення мобільного застосунку можна оптимізувати процес доставки:

* Забезпечте можливість онлайн-замовлень через мобільний додаток. Це дозволить клієнтам швидко та зручно замовляти доставку зі своїх мобільних пристроїв.
* Використовуйте GPS для визначення місцезнаходження кур'єра та адрес доставки. Завдяки цьому ви можете забезпечити точніше планування маршруту та прогнозування часу доставки.
* Забезпечте можливість оплати замовлення через мобільний додаток. Це дозволить клієнтам оплачувати замовлення безпосередньо зі своїх мобільних пристроїв, що значно спрощує процес оплати та збільшує швидкість обробки замовлень.
* Забезпечте кур'єрам доступ до мобільного додатку для відстеження замовлень та маршруту доставки. Це дозволить кур'єрам бути в курсі змін в замовленнях та маршруті та вчасно реагувати на них.
* Використовуйте аналітику для відстеження часу доставки, частоти замовлень та інших метрик, щоб забезпечити ефективність процесу доставки та забезпечити найкращий досвід для клієнтів.
* Використовуйте систему повідомлень для сповіщення клієнтів про статус їх замовлення. Це дозволить клієнтам бути в курсі та отримувати сповіщення про зміни у статусі їх замовлення.
* Розробляйте та оновлюйте мобільний додаток для забезпечення найкращого користу вання з клієнтами. Регулярні оновлення додатку з новими функціями та можливостями можуть збільшити зацікавленість клієнтів і сприяти збільшенню обсягів продажів.
* Забезпечуйте можливість відгуків та оцінок для клієнтів. Це дозволить клієнтам залишати свої коментарі та відгуки про продукти та послуги, а також оцінювати якість обслуговування. Це може бути корисним для збору зворотного зв'язку та покращення процесу доставки.
* Використовуйте інтеграції з іншими системами, наприклад з системами управління запасами або з системами аналітики, щоб забезпечити більш ефективне та точне управління процесом доставки.
* Забезпечуйте доступ до клієнтської служби через мобільний додаток. Це дозволить клієнтам отримувати швидку та ефективну підтримку в разі потреби.

**1.2 Мобільний додаток доставки їжі «Uber Eats»**

Uber Eats є одним з найпопулярніших мобільних додатків доставки їжі у світі. Цей додаток забезпечує зручний та швидкий спосіб замовлення їжі з різних ресторанів та кафе, а також доставку її прямо до вашої двері.

Додаток Uber Eats має дуже простий та зрозумілий інтерфейс, що дозволяє користувачам з легкістю знаходити потрібну їжу та оформляти замовлення. Головна сторінка додатку містить список ресторанів та кафе, які працюють з Uber Eats, а також їх меню та рейтинги. Користувачі можуть шукати їжу за назвою ресторану, кухні, типу їжі, ціни та іншими параметрами.

Коли користувач знаходить потрібну їжу, він може додати її в кошик та оформити замовлення. Після цього користувач має можливість обрати спосіб оплати - кредитною карткою чи через платіжну систему Uber. Вікно вибору адресу зображене на рисунку 1.2.

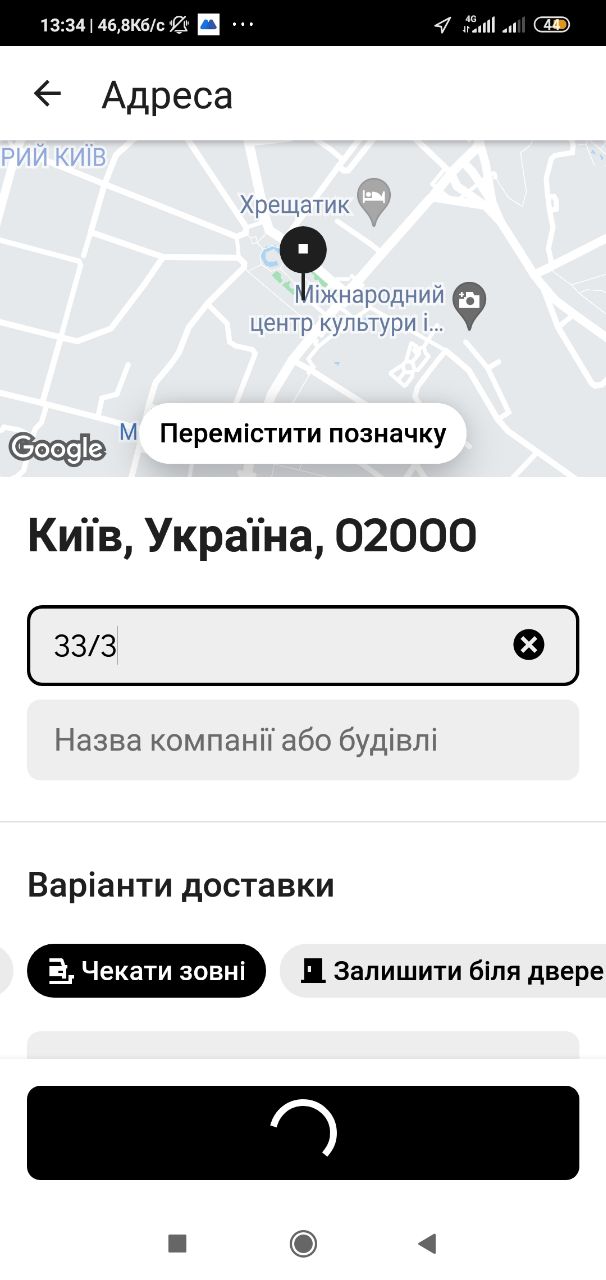


Рисунок 1.2 Вікно вибору закладу поблизу.

Після оформлення замовлення користувач може відстежувати статус доставки в режимі реального часу. Коли їжа готова та відправлена кур'єру, користувач отримає повідомлення про це. Кур'єр забере замовлення у ресторану та доставить його прямо до дверей користувача. Користувач також може зв'язатися з кур'єром прямо з додатку, якщо виникли проблеми з доставкою. Всю інформація про за стосунок можна знайти на офіційному сайті [2].

Його ключові переваги і недоліки представлені в таблиці 1.1. Також додаток Uber Eats має вбудовану підтримку , яка дозволяє користувачам зв'язатися зі службою підтримки в разі виникнення проблем з замовленням чи доставкою. Крім того, додаток має відгуки від інших користувачів, які дозволяють знайти найкращі ресторани та кафе та отримати поради щодо вибору їжі.

Таблиця 1.1 – Переваги та недоліки використання застосунку для доставки їжі «Uber Eats»

|  |  |
| --- | --- |
| Переваги | Недоліки |
| Мобільний додаток Uber Eats має простий та зрозумілий інтерфейс, що робить його легким у використанні | Uber Eats збирає високі комісійні збори від ресторанів, що може призвести до збільшення цін на їжу для клієнтів |
| Uber Eats має великий вибір ресторанів та страв, що дозволяє клієнтам вибирати з різноманітних варіантів | Час доставки може бути затриманим через різні причини, такі як пробки на дорозі або велика кількість замовлень |
| Клієнти можуть сплатити за замовлення за допомогою різних способів, таких як кредитні картки, PayPal та Google Pay | Деякі ресторани або страви можуть бути недоступні в деяких місцевостях |
| Uber Eats має програму лояльності, яка дозволяє клієнтам збирати бонусні бали та отримувати знижки на майбутні замовлення | Незважаючи на те, що Uber Eats намагається надати точний прогноз часу доставки, іноді цей прогноз може бути неточним |

Підбиваючи підсумки після аналізу даного застосунку ми можемо зрозуміти, що це зручний та простий мобільний додаток, який дозволяє замовити їжу з різних ресторанів та кафе та отримати доставку прямо до дверей. Додаток має простий та зрозумілий інтерфейс, зручну систему замовлення та оплати, а також програму лояльності та бонусів для користувачів. Uber Eats є одним з найпопулярніших додатків доставки їжі у світі та є надійним вибором для людей, які шукають зручний спосіб замовлення їжі.

**1.3 Програмний застосунок доставки «Glovo»**

Glovo - це програмний застосунок, що пропонує послуги доставки різних товарів до дому. Застосунок доступний для смартфонів на платформах iOS та Android. Головні переваги і недоліки програмного за стосунку представлені в таблиці 1.2. Головна ідея застосунку полягає в тому, щоб допомогти користувачам отримати товари та послуги, які вони потребують, просто замовивши їх з телефону.

Glovo дозволяє замовляти їжу з ресторанів, продукти харчування з супермаркетів, квіти з квіткових магазинів, товари з будь-якого магазину та навіть прокат транспорту. Користувачі можуть вибрати товари, які вони хочуть замовити, та зробити оплату безпосередньо зі свого телефону.

Для забезпечення доставки Glovo використовує свою власну мережу кур'єрів. Коли замовлення зроблене, кур'єр підтверджує його та прибуває до магазину або ресторану, щоб отримати товари. Після цього він прямує до місця доставки, де передає товари клієнту. Клієнт може відстежувати процес доставки у режимі реального часу та зв'язатися з кур'єром, якщо виникають питання.

Однією з великих переваг Glovo є те, що вона працює з безліччю ресторанів та магазинів. Замовлення можна робити в будь-якому місці та в будь-який час. Крім того, Glovo має можливість доставляти товари в будь-яке місце, незалежно від того, чи є це будинок, квартира, офіс або готель. Всю інформація про застосунок можна знайти на офіційному сайті [3]

Для замовлення в Glovo потрібно завантажити програму на свій телефон та створити обліковий запис. Користувач може вибрати з декількох способів оплати, включаючи оплату за допомогою картки, електронні гаманці або на місці доставки. Після цього можна починати робити замовлення. Glovo пропонує швидкий та зручний процес замовлення зображено на рисунку 1.3, який дозволяє замовити все, що потрібно, у декількох кліках.

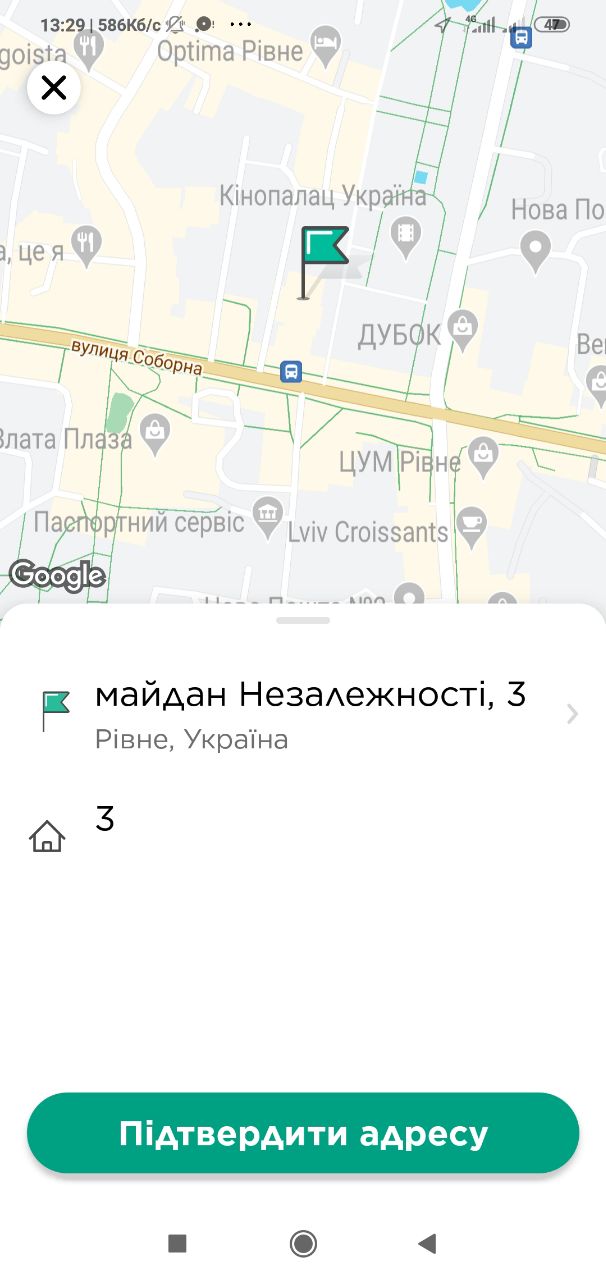


Рисунок 1.3 Вікно вибору адреси доставки.

Крім того, Glovo пропонує користувачам можливість вибирати з різних категорій товарів та послуг зображено на рисунку 1.4. Наприклад, користувач може замовити їжу з різних типів ресторанів, з різних кухонь, з різними типами доставки (безкоштовною або платною), або замовити продукти з різних супермаркетів. Крім того, Glovo дозволяє користувачам замовляти різні товари з магазинів, такі як електроніка, одяг, косметика, книги та інші.



Рисунок 1.4 Вікно вибору товару.

Окрім великого вибору товарів та послуг, Glovo пропонує зручну систему винагород для своїх користувачів. Наприклад, програма запропонує знижки та промокоди для нових користувачів, а також програму лояльності для постійних клієнтів. Крім того, Glovo пропонує можливість заробляти гроші як кур'єри, доставляючи товари замовлені іншими користувачами.

Таблиця 1.2 – Переваги та недоліки використання програмного за стосунку доставки «Glovo»:

|  |  |
| --- | --- |
| Переваги | Недоліки |
| Широкий вибір ресторанів та продуктів | Висока вартість доставки |
| Можливість вибору дати та часу доставки | Обмежені зони доставки |
| Зручність інтерфейсу додатка | Можливі затримки у доставці |
| Швидкість та оперативність виконання замовлення | Обмежені можливості для вибору часу доставки |
| Опції відстеження замовлення та сповіщення | Відсутність можливості оплати готівкою |
| Можливість використання промокодів та знижок | Обмежені можливості налаштування доставки |

Підбиваючи підсумки після аналізу даного за стосунку ми можемо зрозуміти побачити його переваги та недоліки і зрозуміти, що у цілому, Glovo - це зручний та швидкий спосіб отримати товари та послуги, які потрібні в будь-який час та в будь-якому місці.

Застосунок пропонує великий вибір товарів та послуг, зручні способи оплати, можливість відстежувати доставку та зв'язатися з кур'єром, а також систему винагород для користувачів. За допомогою Glovo можна замовити все, що потрібно, з комфортом та без зайвих зусиль.

**1.4 Мобільний додаток доставки товарів «Нова пошта»**

Мобільний додаток "Нова пошта" - це програмне забезпечення, призначене для зручної та швидкої доставки товарів. Це додаток дозволяє клієнтам замовляти доставку та отримувати товари, що були доставлені відправниками через службу "Нова пошта".

Додаток доступний на платформах Android та iOS і можна завантажити з відповідних магазинів додатків. Після завантаження додатка, користувачам необхідно зареєструватися та увійти в свій обліковий запис, щоб почати користуватися всіма можливостями додатка.

Додаток має чистий та сучасний дизайн, що дозволяє користувачам легко зорієнтуватися в його функціях та можливостях. Головне меню додатка складається з таких елементів як "Мої замовлення", "Картки отримувачів", "Адресна книга", "Тарифи та терміни доставки" та "Налаштування".

"Мої замовлення" - це розділ, в якому користувач може переглянути свої замовлення та статус їх виконання, зображено на рисунку 1.5. Додаток надає інформацію про те, коли був здійснений останній крок в доставці, такий як надсилання, транспортування, доставка, прибуття до відділення та інше. Крім того, у цьому розділі користувач може переглянути деталі замовлення, такі як вага, розмір, адреса відправника та отримувача.

"Картки отримувачів" - це розділ, в якому користувач може зберегти контактну інформацію отримувачів та використовувати її для швидкого заповнення замовлень. Додаток дозволяє зберігати необмежену кількість карток отримувачів, що забезезпечує зручний та швидкий процес замовлення доставки. Всю інформація про застосунок можна знайти на офіційному сайті [4].

"Тарифи та терміни доставки" - це розділ, в якому користувач може дізнатися про різні тарифи та терміни доставки, які пропонує служба "Нова пошта". Користувач може порівняти тарифи та вибрати найбільш підходящий для себе варіант. Крім того, додаток надає інформацію про додаткові послуги, такі як страхування вантажу, обережне поводження з вантажем та інше.

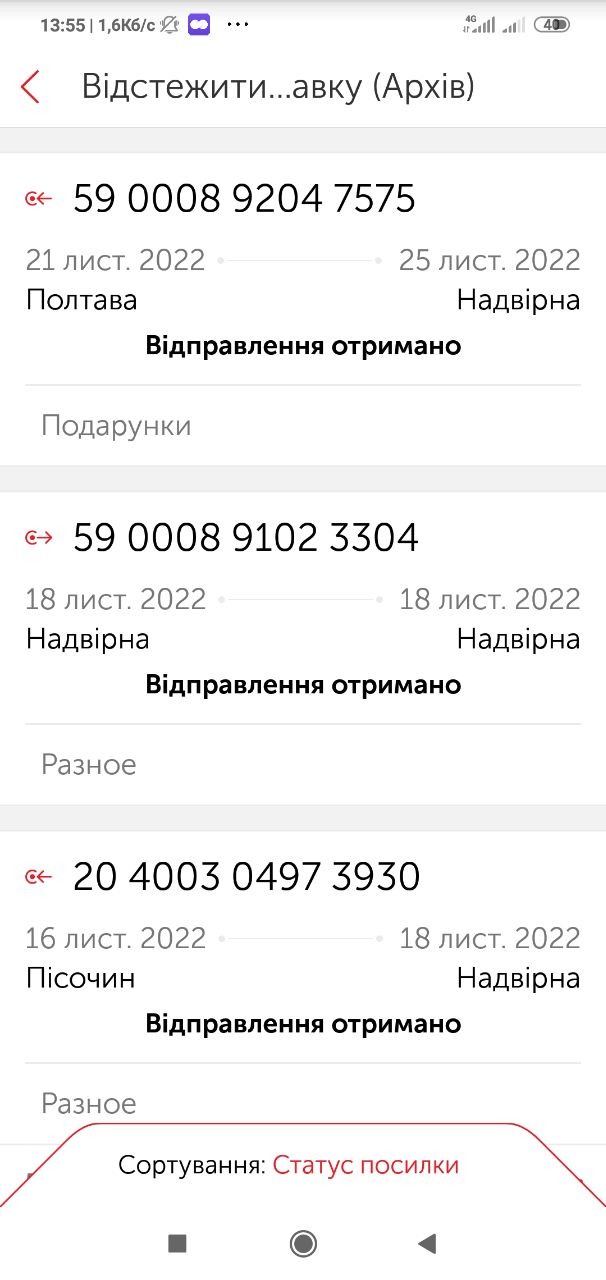


Рисунок 1.5 вкладка «Мої замовлення».

"Адресна книга" - це розділ, в якому користувач може зберігати адреси відправників та отримувачів, що дозволяє швидко заповнювати замовлення та зекономити час на пошук адреси. Додаток дозволяє зберігати необмежену кількість адрес в адресній книзі.

"Налаштування" - це розділ, в якому користувач може налаштувати свій обліковий запис, змінити пароль, відключити повідомлення та інше.

Один з основних функціоналів додатка "Нова пошта" - це можливість створювати нові замовлення доставки. Додаток надає широкі можливості для створення замовлення, користувач може вибрати тип доставки (стандартна, експрес, доставка у вихідні), вибрати адресу відправлення та отримання, ввести інформацію про вантаж (вагу, розміри, кількість), вибрати тариф та додаткові послуги (страхування, обережне поводження з вантажем та інше).

Додаток "Нова пошта" надає можливість відстежувати статус замовлення в режимі реального часу зображено на рисунку 1.6, що дозволяє користувачам бути в курсі, де знаходиться їх замовлення та коли вони можуть очікувати доставку. Крім того, додаток надає інформацію про орієнтовну дату та час доставки.

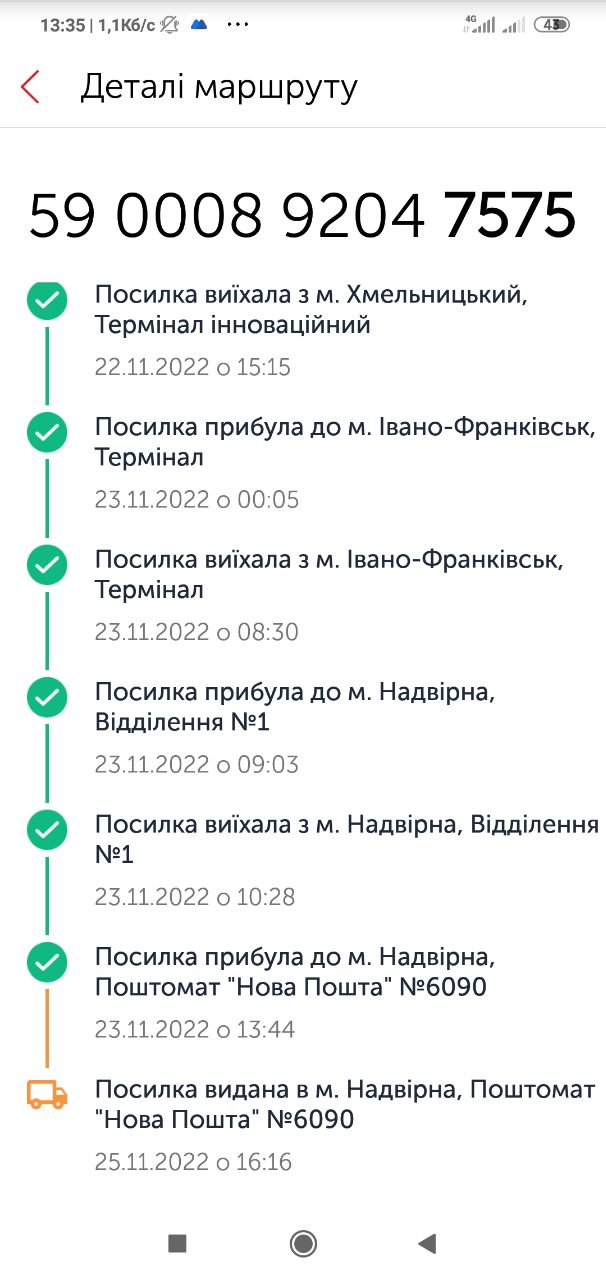


Рисунок 1.6 вкладка «Мої замовлення».

Також, користувач може відслідковувати відправлення за допомогою "Штрих-коду", який він отримує при створенні замовлення доставки. Користувач може відсканувати штрих-код з допомогою камери свого смартфона та отримати інформацію про статус свого замовлення.

"Склади Нової Пошти" - це розділ, в якому користувач може знайти інформацію про склади Нової Пошти, які знаходяться поруч з ним. Переваги і недоліки представлені в таблиці 1.3. Також користувач може знайти найближчий склад та отримати інформацію про його розташування та графік роботи зображено на рисунку 1.7. Окрім цього, додаток дозволяє здійснювати онлайн-оплату за доставку вантажу та слідкувати за статусом оплати.

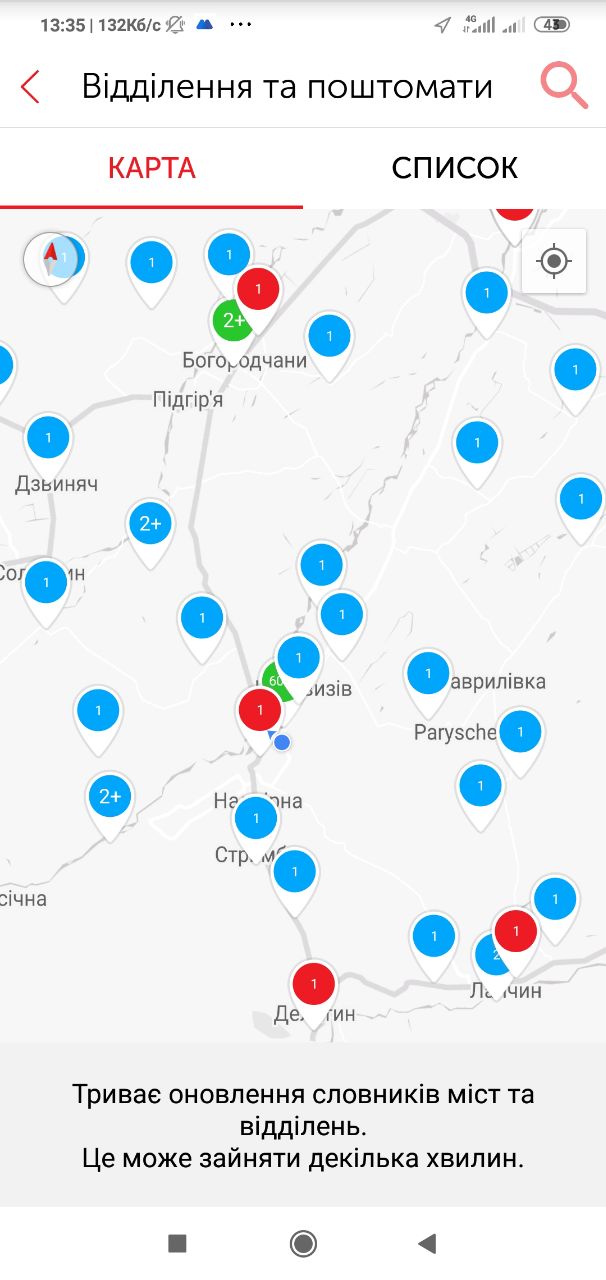


Рисунок 1.7 вкладка «Мої замовлення».

Інтерфейс додатка "Нова пошта" є зручним та інтуїтивно зрозумілим. Користувач може легко знайти потрібний розділ та виконати необхідні дії. Додаток також підтримує різні мови, включаючи українську, російську та англійську.

Таблиця 1.3 Переваги та недоліки використання мобільного додатку доставки товарів «Нова пошта»:

|  |  |
| --- | --- |
| Переваги | Недоліки |
| Додаток дозволяє здійснювати замовлення та відслідковувати їх статус в будь-який час та в будь-якому місці | Додаток потребує підключення до Інтернету для роботи, що може бути проблемою для користувачів, які мають обмежений доступ до мережі |
| Доставка замовлення зазвичай займає всього кілька днів, що є дуже швидко для більшості користувачів | Іноді можуть виникати технічні проблеми, такі як зависання додатка або затримки в оновленні статусу відправлення |
| Додаток має широкий функціонал, який дозволяє користувачам здійснювати замовлення, сплачувати за послуги доставки, відстежувати статус відправлень та багато іншого | Доставка здійснюється тільки в межах України і Польщи, тому додаток не є зручним для користувачів, які хочуть здійснювати доставку в інші країни |
| "Нова пошта" має добру репутацію та гарантує безпечну доставку відправлень | Вартість послуг доставки може бути високою, що може бути проблемою для деяких користувачів |

Зважаючи на цю таблицю, можна побачити, що переваги мобільного додатку "Нова пошта" переважають над його недоліками. На жаль, деякі з недоліків, такі як залежність від Інтернету та технічні проблеми, є типовими для більшості мобільних додатків, і їх неможливо уникнути повністю. Однак, є деякі недоліки, які можуть бути поліпшені, наприклад, зниження вартості доставки, яка може привести до збільшення популярності сервісу серед користувачів.

**1.5 Постановка задачі.**

Об’єкт проєкту: "Розроблення програмного застосунку прокладання маршрутів та оптимізації процесу доставлення товарів" є створення програми, яка допоможе підприємствам з логістики та доставки ефективно організувати процес доставки товарів до клієнтів.

Предмет проєкту: розробка програмного за стосунку.

Мета проєкту полягає в тому щоб зменшети час та витрати на доставку, що досягається завдяки автоматизованому прокладанню оптимального маршруту з урахуванням різноманітних факторів, таких як відстань між точками, тип транспорту, наявність дорожнього руху, графік доставки та інші.

Завдання проєкту:

1. Розробити алгоритм прокладання оптимального маршруту з урахуванням різних факторів, який буде використовуватися в програмному застосунку.
2. Розробити інтерфейс користувача, який буде інтуїтивно зрозумілим та зручним для використання.
3. Реалізувати функцію моніторингу місцезнаходження транспортних засобів та обробки цієї інформації в програмному застосунку.
4. Розробити модуль для визначення оптимального способу доставки товарів в залежності від їх характеристик та інших факторів.
5. Розробити функцію прогнозування часу доставки та повідомлення про затримки.
6. Реалізувати функцію визначення оптимальної маршрутної схеми для багатосторонніх доставок та доставок до точок з однаковою адресою.
7. Розробити функцію звітування та аналізу результатів роботи програмного застосунку для підприємств-користувачів.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ПРОКЛАДАННЯ МАРШРУТУ

**3.1 Розробка бази даних**

Для реалізації додатку прокладання маршруту було створено базу даних за допомогою firebase.

У ній було реалізовано процес реалізації авторизації-реєстрації користувача в самому додатку зображено на рисунку 3.1, за допомогою: номера телефону, електронної адреси або Google акаунта.

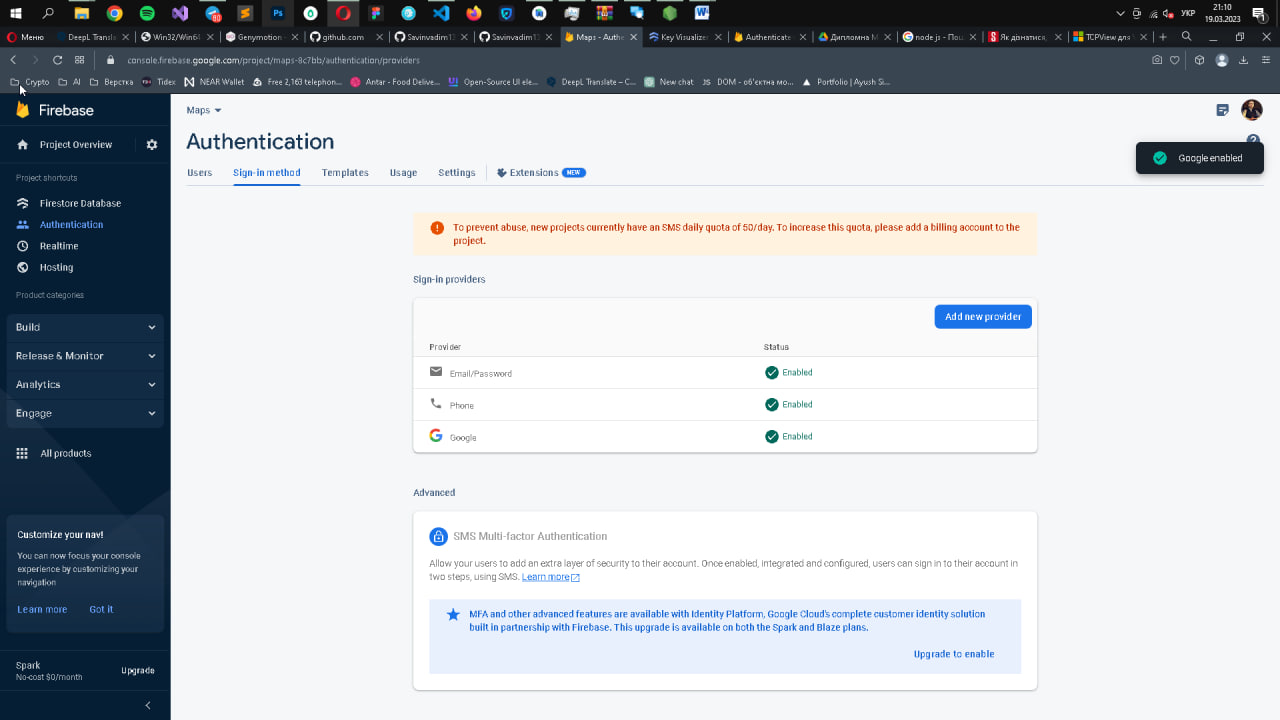


Рисунок 3.1 Методи авторизації

Структурно база даних поділяється на 3 таблиць: Метод входу, Логін,Пароль, зображено на рисунку 3.2.

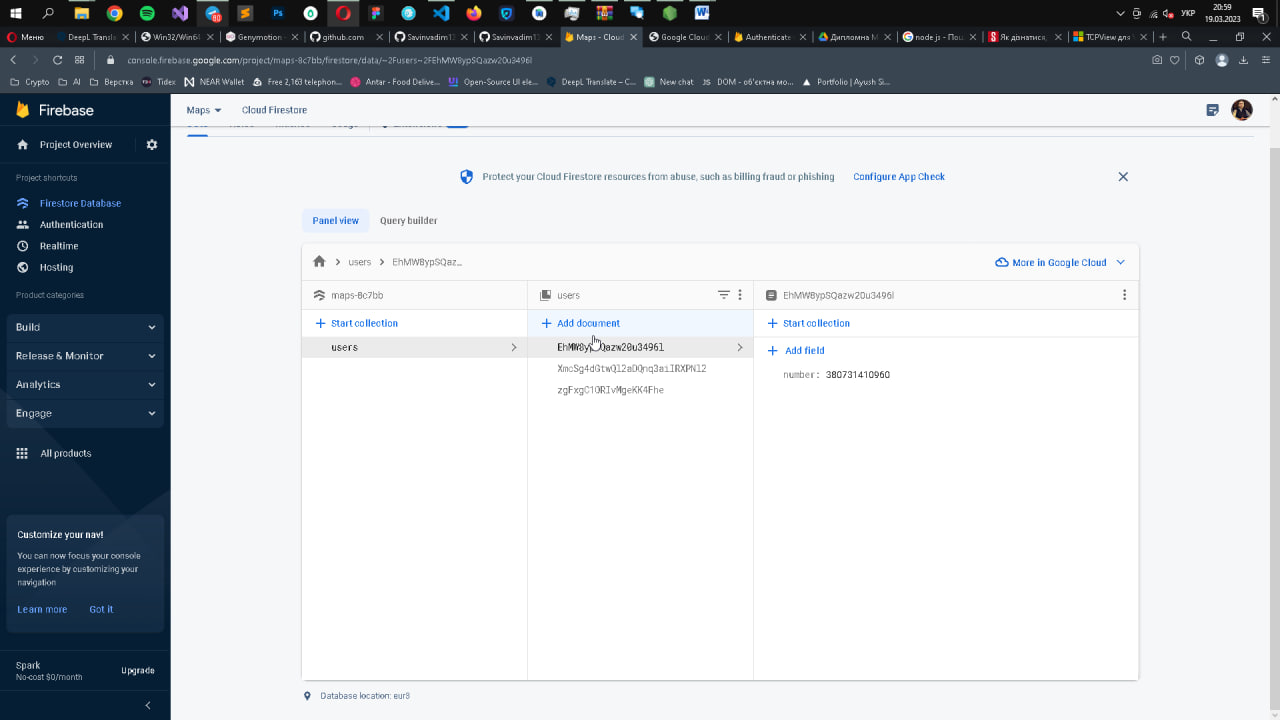


Рисунок 3.2 База даних

Таблиця «Метод входу» містить відомості про метод входу, який вибрав користувач це може бути: електронна адреса, мобільний телефон або за допомогою Google account.

Таблиця «Логін» містить відомості про логін, який використав користувач під час реєстрації, це може бути електронна адреса, його номер телефону, який він пізніше може змінити в профілі на своє Прізвище і ім’я або буде використані дані автоматично взяти з Google account.

Таблиця «паролі» містить відомості про паролі. Через первинний ключ «пароль\_id» вона поєднується з «Логіном»

Також на рисунку 3.3 зображено кількість авторизацій в програму і представлено графік активності.

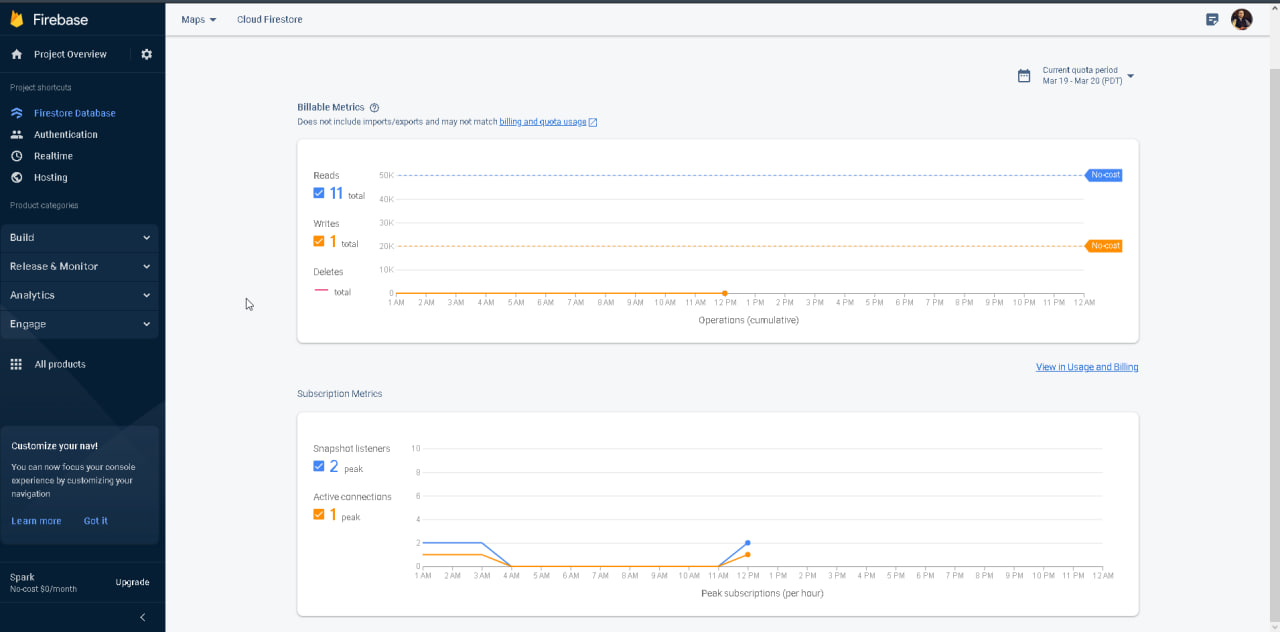


Рисунок 3.3 Активність

**3.2 Програмний інтерфейс системи**

У результаті виконання дипломного проекту було створено мобільний за стосунок за допомогою React Native, який працює з можливостями прокладання оптимального маршруту для доставки товару. Додаток має зручний та зрозумілий інтерфейс для користувача та функціонал. При запуску додатку відкривається вікно реєстрації, в якому користувач пропонується здійснити авторизацію, зображено на рисунку 3.4, за допомогою номера телефона, електронну адресу, або власного Google аканта, що зменшує час для входу [1].

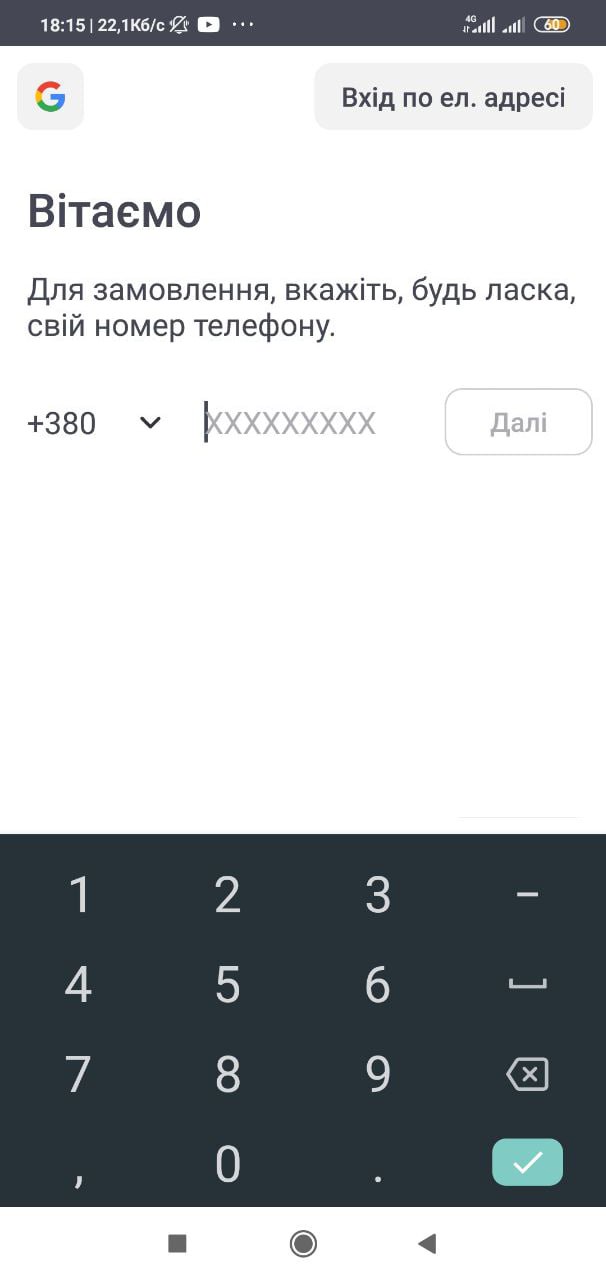


Рисунок 3.4 Процес авторизації

Авторизуватися в додатку можуть тільки користувачі, які досягнули віку 14 років, ці дані перевіряються під час входу користувача через гул акант або. Після успішної авторизації користувача буде переадресована на його профіль зображено на рисунку 3.5, в ньому він може прочитати інструкцію, зображено на рисунку 3.7 в вкладці «Допомога», також може переглянути, які методи оплати йому доступні зображено на рисунку 3.4.

В вкладці «Доставка» він може переглянути де знаходиться його посилка в реальному часі [9].

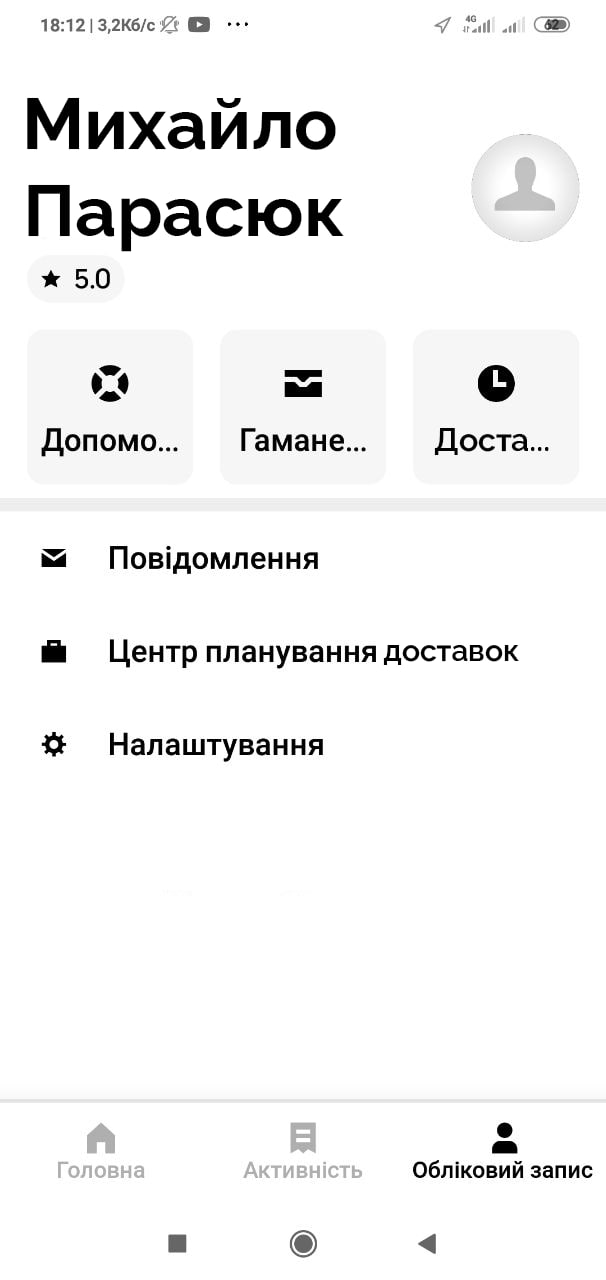


Рисунок 3.5 Обліковий запис

В вкладці «Оплата» користувач може вибрати тип оплати, представлено 2 типа за допомогою «Google Pay», і метод в якому потрібно добавити свою картку.

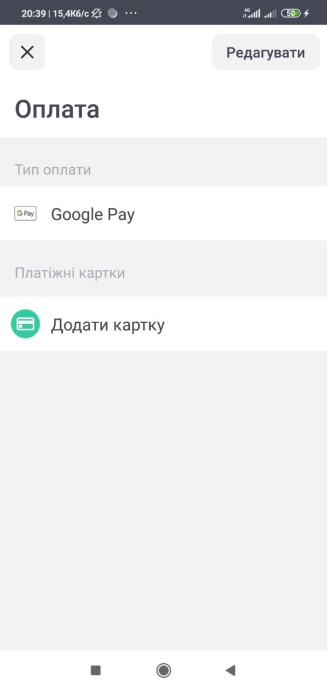


Рисунок 3.6 Оплата

В вкладці «Доставка», користувач має змогу переглянути детальну інструкцію, якими методами, здійснюються доставки, також які ресторани і заклади мають змогу здійснити доставку. Також він може залишити відгук, якщо його замовлення не надійшло до нього, або прийшли зайві страви і товари.

Користувач може залишити відгук про те що з нього стягнуло плату за скасовану доставку, також користувач має змогу залишити відгук про те що його замовлення прибуло пізніше, ніж очікувалося [4].

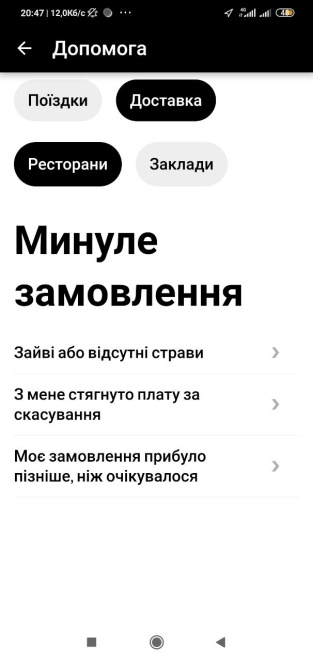


Рисунок 3.7 Допомога

В вікні «Вибір адрес», зображено на рисунку 3.8, за допомогою пошуку, користувач зможе знайти місце доставки яке йому потрібне, також він може зберегти місце адреси в вибрані, за допомогою кнопки «Головна», користувач буде направлений в свій профіль.

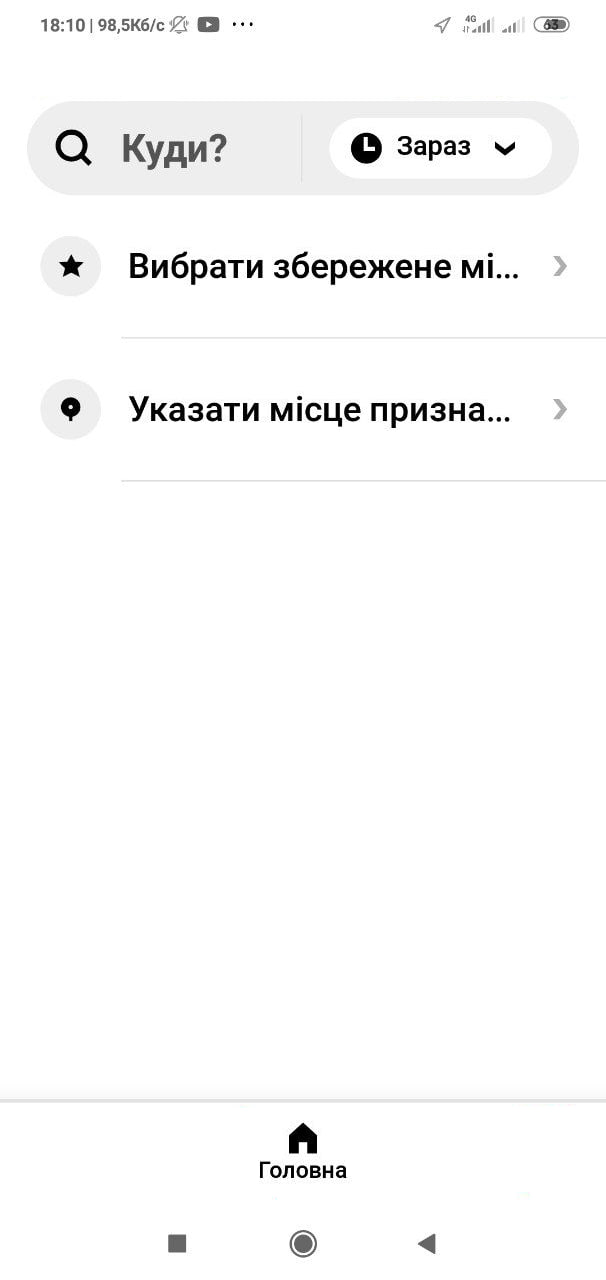


Рисунок 3.8 Вибір адреси

Якщо користувач введе адресу, або натисне на кнопку «Указати місце призначення», він буде пере направлений на вікно з картою, де зможе вибрати оптимальний маршрут доставки і тим транспорту за допомогою якого буде здійснюватися доставка.

Можна вибрати тип транспорта і від цього буде змінюватися час доставки і ціна:

* Звичайна доставка
* Доставка велосипедом
* Доставка самокатом

Від цього буде залежати швидкість доставки, сума доставки і планування оптимального маршруту до місця призначення.

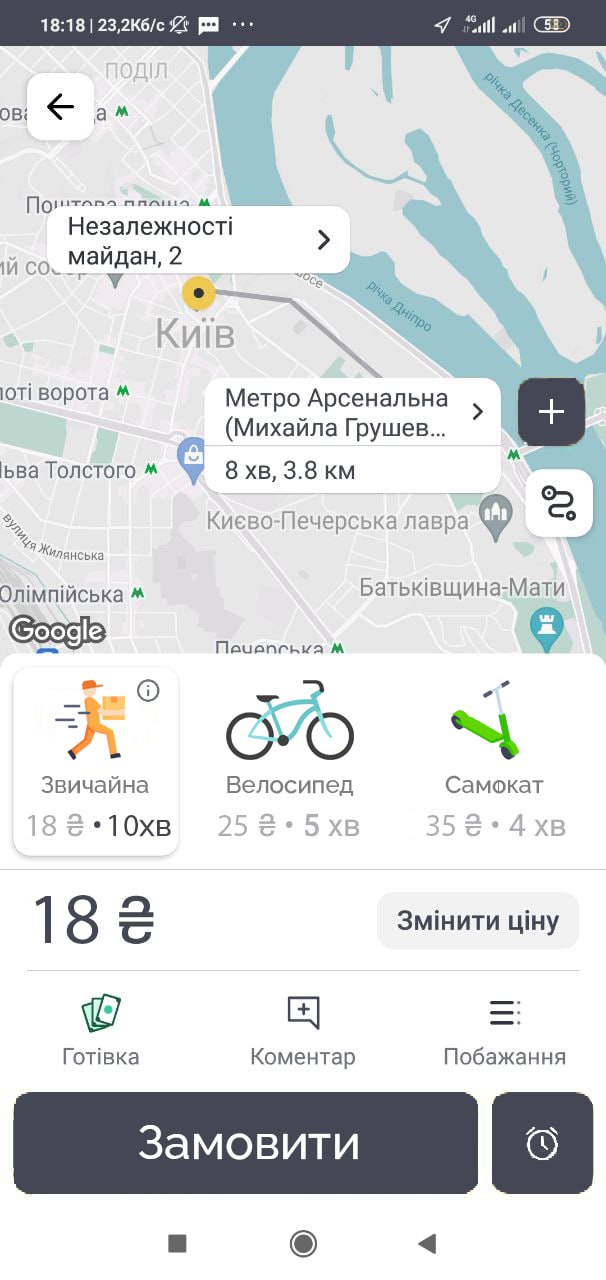


Рисунок 3.9 Оптимальний шлях доставки

Для підтвердження здійснення замовлення користувач буде пере направлений в вкладку зображено на рисунку 3.10, де потрібна вказати свій мобільний номер телефону. Для коректного продовження користувач повинен вказати номер, який використовував для реєстрації в додатку, в іншому випадку, він не зможе отримати свою доставку.

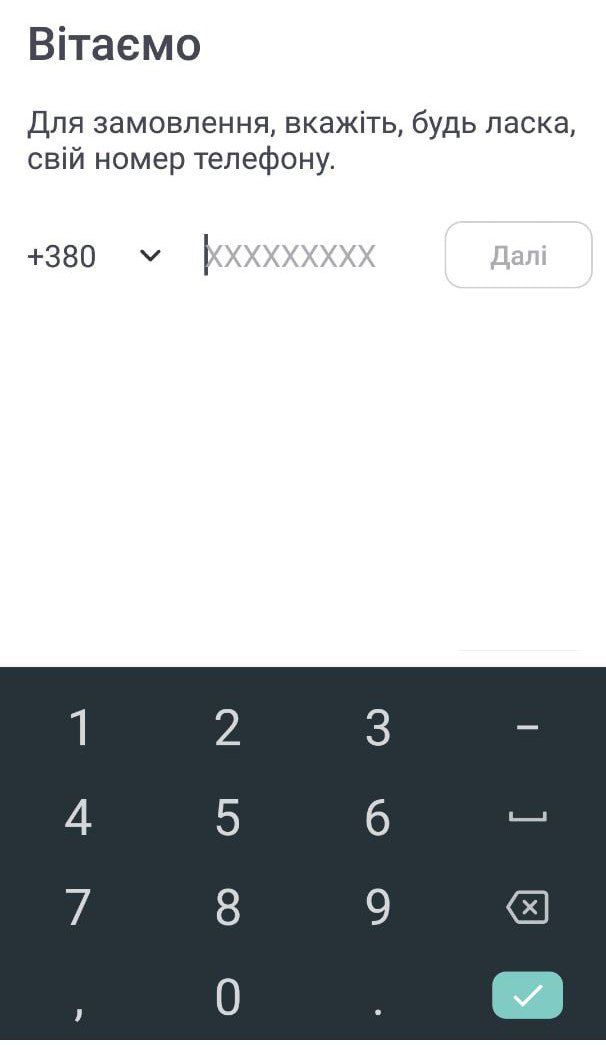


Рисунок 3.10 Підтверження

Після підтвердження користувач буде перенаправлений на вкладку в додатку під назвою «Поповнити», де можна буде здійснити оплату, але спочатку потрібно буде вибрати спосіб оплати зображено на рисунку 3.11.

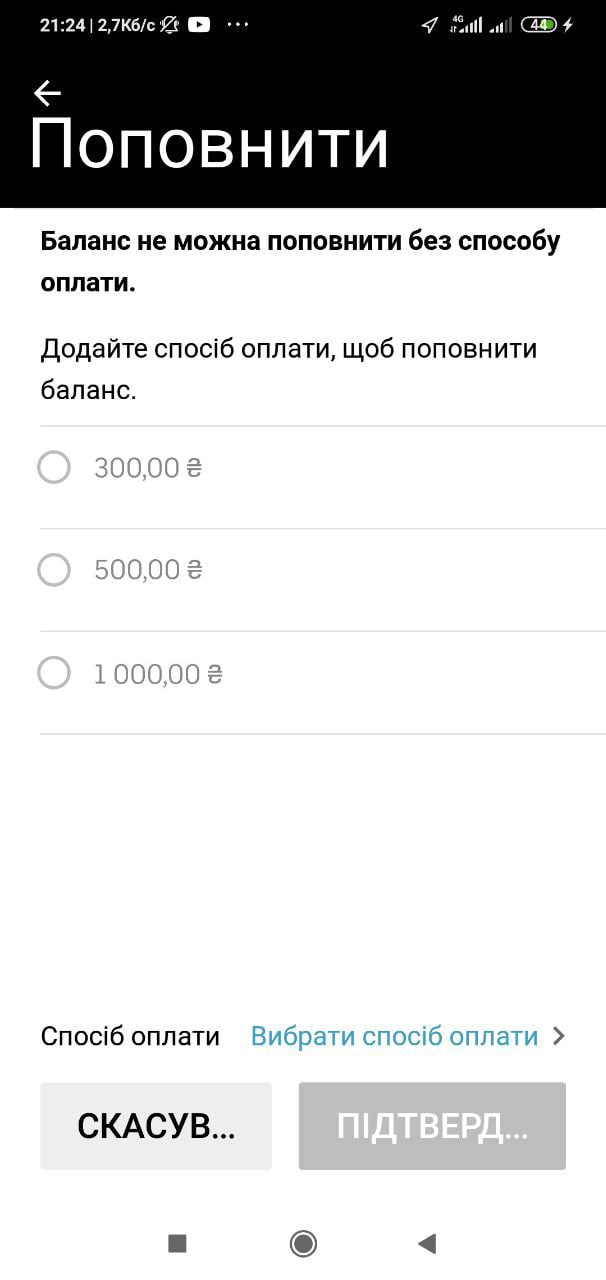


Рисунок 3.11 Поповнення балансу

В вкладці «Оплата», зображено на рисунку 3.12 користувач повинен здійснити вибір методу оплати, представлено 2 типа за допомогою яких можна здійснити оплату це:

* «Google Pay»,
* «Додати картку».

Google Pay - це сервіс електронних платежів, розроблений компанією Google, який дозволяє користувачам відправляти та отримувати гроші через мобільні пристрої. За допомогою Google Pay можна здійснювати безконтактні платежі на місцях, де приймаються платежі з використанням NFC-терміналів, а також онлайн-платежі з мобільних пристроїв.

Для того, щоб користувач міг використовувати Google Pay, він повинен зареєструватися в сервісі та додати свої банківські картки. Для додавання карток можна скористатися камерою свого смартфона, щоб зчитати дані картки, або вручну ввести номер картки та інші необхідні дані.

Для зручності було добавлено цей метод, він набагато швидший ніж добавлення картки вручну.

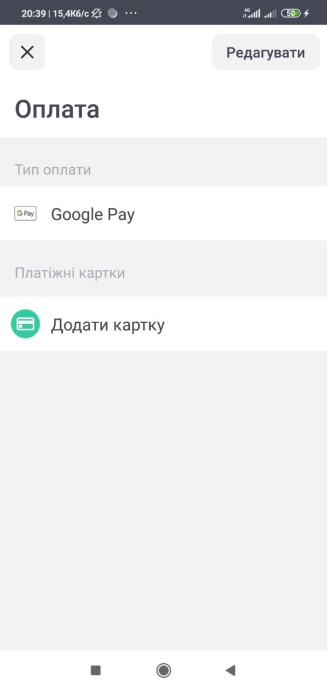


Рисунок 3.12 Вибір типу оплати

Підводячи підсумок даного пункту, можна зробити висновки що було зроблено зручний і зрозумілий для користувачів інтерфейс, було зроблено оптимальне прокладання маршрутів, які за допомою API отримують від «Google Maps» дані про ускладнений рух на тій чи іншій ділянці дороги, також було добавлено зручний метод оплати це «Google Pay».

**3.3. Тестування функціоналу**

Мобільний додаток було протестовано з точки зору користувача, тобто методом чорного ящика. Тестування проводилось по основному функціоналу додатку.

Тестування авторизації користувача було зображено на рисунках 3.13, якщо користувач ввів не правильний формат електронної адреси, кнопка «Вввійти» не буде працювати

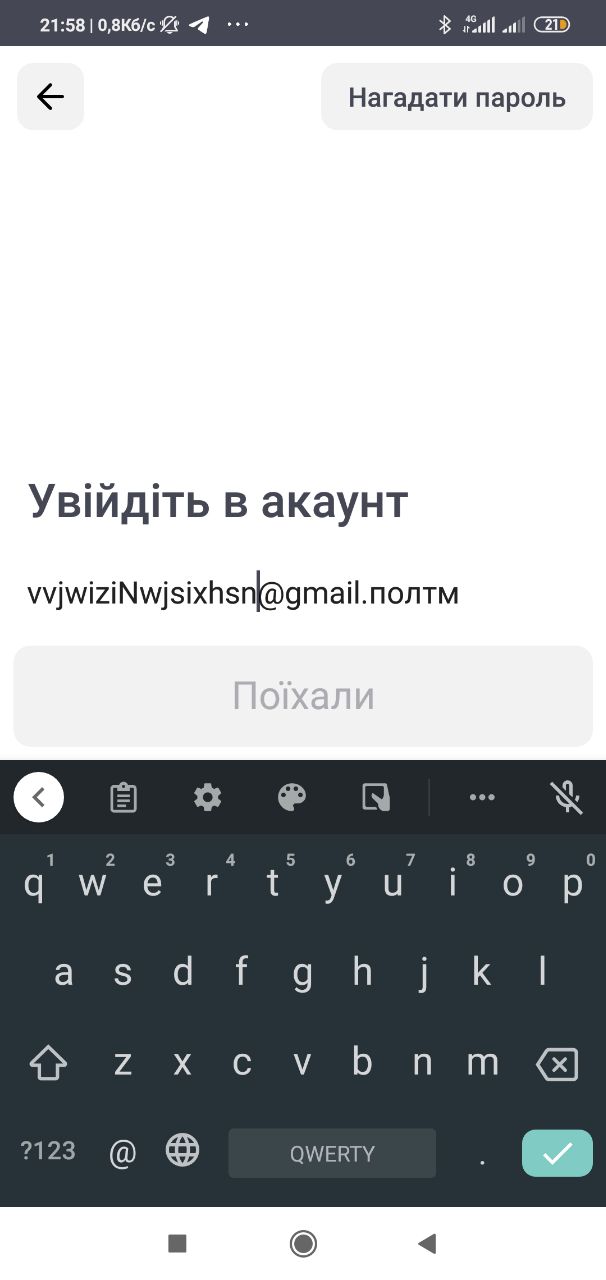


Рисунок 3.13 Тестування авторизація через пошту

Згідно з результатами тестування, авторизація працює успішно, оскільки користувач може перейти на наступну вкладку тільки при коректно введених даних зображено на рисунку 3.14.

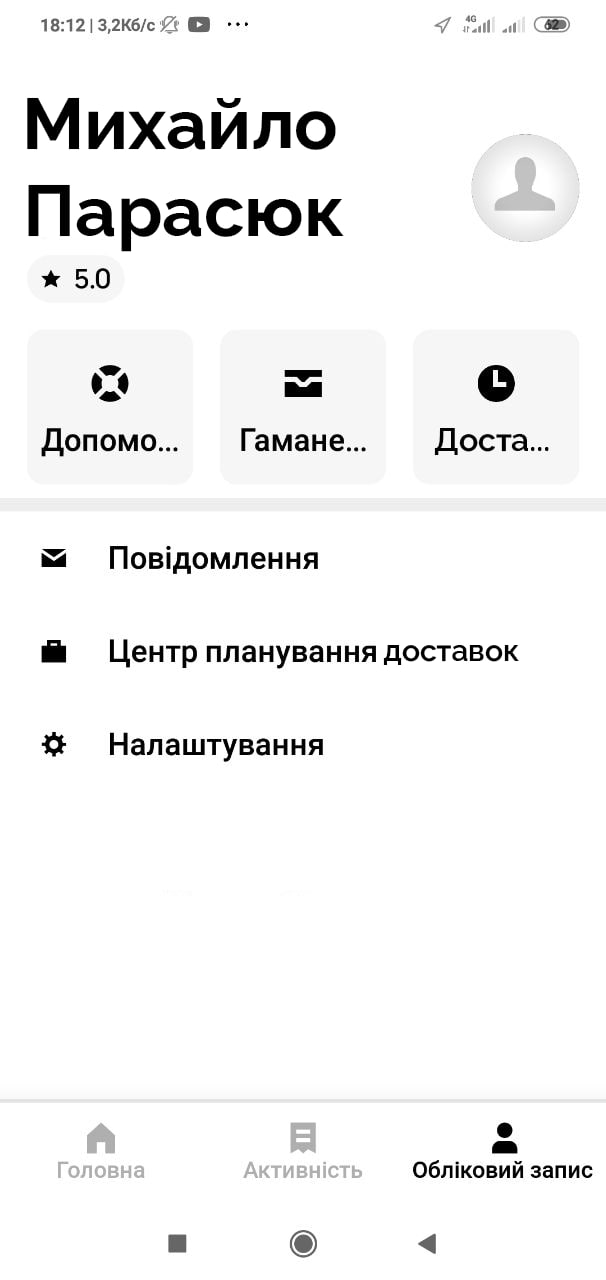


Рисунок 3.14 Успішний вхід

Тестування використання карти і вибір типу доставки є основним функціоналом додатку. Його результат представлено на рисунках 3.15. Результат тестування показує, що оптимальний маршрут прокладається правильно і час, сума доставки буде відрізнятися від вибору методу доставки.

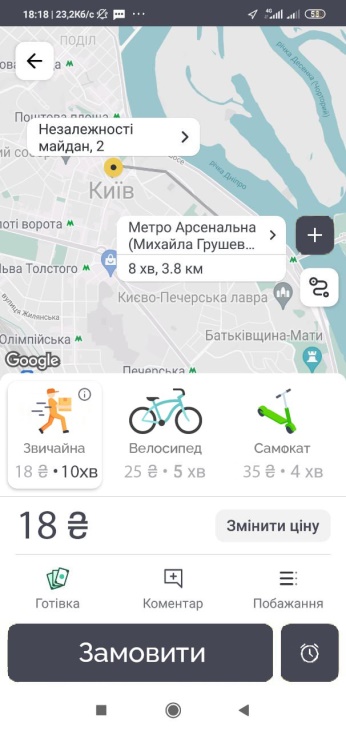


Рисунок 3.15 Тестування карти

Також було проведено тестування методу оплати зображено на рисунку 3.16, що теж вважається основним функціоналом програми, без якої вона не зможе працювати.

Якщо користувач не вибере на яку суму він повинен здійснити поповнення і не вибере метод яким має здійснити оплату користувач не зможе нажати кнопку «Підтвердити» і перейти на наступне вікно програми «Підтвердження»

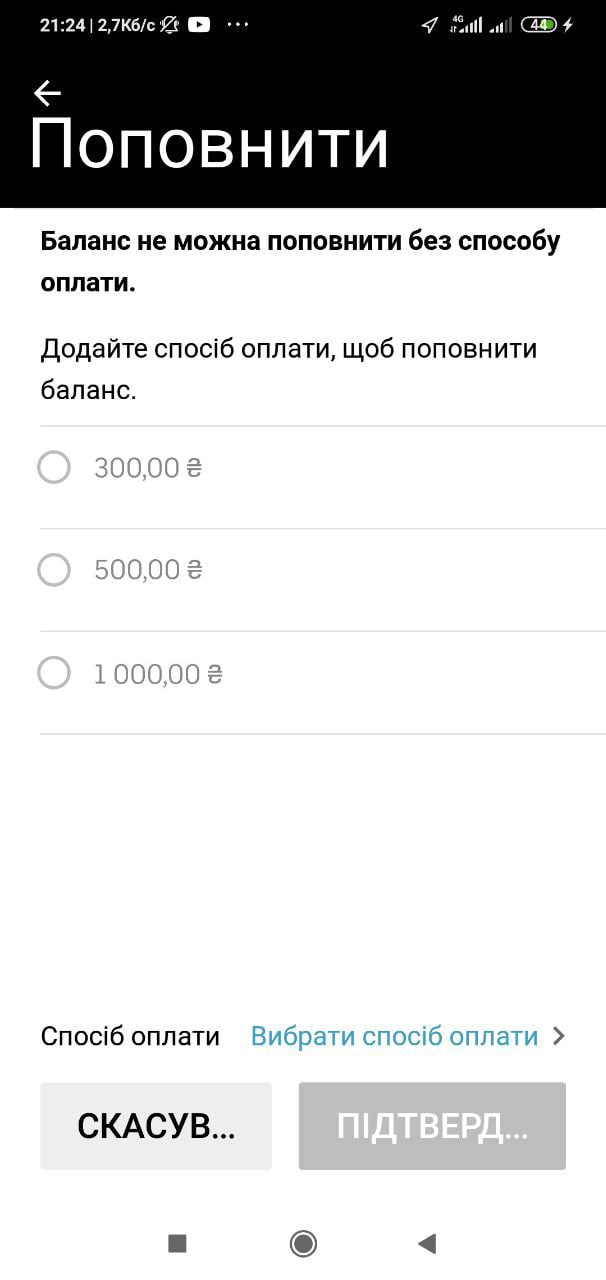


Рисунок 3.16 Тестування оплати

Тестування додатку методом чорного ящика пройшло успішно. Завдяки його результатам було зроблено висновок про те, що додаток працює коректно, тобто виконує свій функціонал.

Якщо в користувача виникнли якісь запитання або проблеми з використанням додатку він може переглянути вкладку «Допомогу» зображено на рисунку 3.17, де зможе прочитати, як вирішити ту чи іншу помилку.

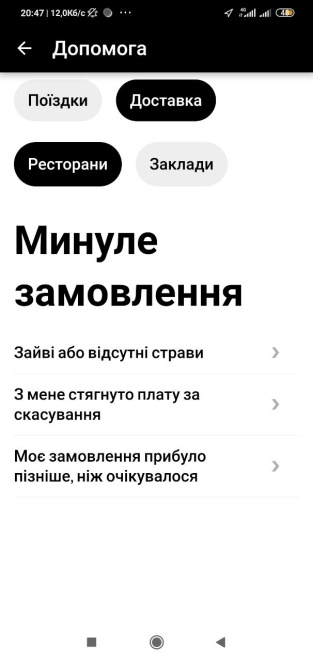


Рисунок 3.17 Тестування оплати

4 ОХОРОНА ПРАЦІ

**4.1 Небезпечні і шкідливі фактори при роботі з ПК.**

В останні час більше уваги стало приділятися поліпшенню умов праці користувачів електронно-обчислювальних машин та відео дисплейних терміналів, незважаючи на те, що якість і безпеку самих ПК і ВДТ постійно поліпшують. У більшості розвинених країнах, в тому числі в США, Німеччині, Швеції, питання про небезпеку роботи за дисплеями піднялася до рівня національної проблеми. У Німеччині робота за дисплеями входить в список 40 найбільш шкідливих і небезпечних професій.

До основних шкідливих факторів при роботі з комп'ютером відносять: тривале сидяче положення, електромагнітне випромінювання, навантаженнРефе на зір, перевантаження кистьових суглобів, можливість захворювань органів дихання, алергії, порушення нормального перебігу вагітності та ін.

Тривале сидяче положення приводить до напруги м'язів шиї, голови, рук і плечей, остеохондрозу, у дітей - ще й до сколіозу. Тривале сидяче положення ще приводить до застою крові в тазових органах і, як наслідок, до простатиту й геморою. Не секрет, що малорухливий спосіб життя призводить до ожиріння. Остеохондроз виникає при порушенні міжхребцевих дисків, яке призводить до випинання в яку або сторону (грижі міжхребцевого диска). Грижа може зашкодить спинний мозок і нервові відростки. Наслідки можуть бути найрізноманітнішими, від болів в спині і кінцівках, до паралічу кінцівок і смерті. Одна з поширених причин остеохондрозу - дистрофія м'язів спини. Людина, провідний в основному сидячий спосіб життя, цілком може захворіти остеохондрозом. Ознаки початку захворювання: дискомфорт у спині та больові відчуття, головні болі, порушення роботи внутрішніх органів. До факторів ризику захворювання гемороєм відносять: сидячий спосіб життя, ожиріння, надмірне вживання копчених, гострих, солоних і пряних продуктів, запальні захворювання малого таза та ін. Ожиріння виникає через нераціональне харчування, малорухомого і в тому числі сидячого способу життя, неадекватної реакції на стресові ситуації, надмірно довгий сон, застосування гормональних препаратів, перевантаження організму харчовими жирами і ін Ожиріння призводить до збільшення навантаження на серце, зміни конфігурації та положення серця в грудній порожнині, підвищення вмісту холестерину в крові, в результаті він відкладається на стінках судин ( атеросклероз). Підвищений скупчення жиру всередині грудної порожнини впливає на роботу органів дихання, що призводить до появи задишки та гіпоксії органів і тканин.

Навантаження на зір*.* Людське око реагує на найдрібнішу вібрацію тексту і на мерехтіння екрану. М'язи ока, керуючі кришталиком, перебувають у постійній напрузі, що обов'язково призводить до втрати гостроти зору. Немаловажне значення для профілактики зорових дисфункцій надають: правильний чи рекомендований підбір кольору, шрифтів, компоновки вікон у використовуваних додатках, орієнтація дисплея монітора. Тривала робота за комп'ютером - це величезне навантаження на очі, оскільки зображення на моніторі складається не з безперервних ліній, як на папері, а з окремих точок, які світяться і мерехтять. У користувача неминуче погіршується зір, очі починають сльозитися, з'являється головний біль, втома, зображення двоїться і спотворюється.

Перевантаження суглобів кистей рук приводить головним чином до такого явища, як синдром зап'ястного каналу.

Робота за комп'ютером і стреси*.* Стрес - це емоційні переживання, внутрішнє напруження, викликані подіями в житті. Стрес виникає, в першу чергу, при втраті або пошкодженні інформації. Причини: відсутність резервних копій, комп'ютерні віруси, поломки жорстких дисків, робочі помилки. Іноді стреси є причиною інфарктів. Стреси бувають емоційно позитивними і емоційно негативними, короткочасними і довгостроковими, гострими і хронічними, фізіологічними і психологічними (інформаційними й емоційними). Робота за комп'ютером є одним з факторів, що викликають стрес (стресором). Реакція організму на стрес являє собою запуск біохімічних процесів, які спрямовані на придушення екстремальній ситуації. Стресові ситуації і пов'язані з ними переживання викликають в організмі численні негативні зрушення.

Робота за комп'ютером і органи дихання*.* Захворювання органів дихання у даному контексті носять в основному алергічний характер. Це пояснюється тим, що за час довгої роботи комп'ютера корпус і плати останнього виділяють в повітря ряд шкідливих речовин, а так само комп'ютер створює навколо себе електростатичне поле, яке притягує пил, який осідає в легенях. Так же комп'ютер деіонізірует навколишнє середовище і зменшує вологість повітря. Алергія – це підвищена чутливість організму до різних подразників, що виявляється в специфічних реакціях при контакті з ними. Це викликає такі симптоми алергії як риніт, сльозоточивість, шкірний висип, анафілактичний шок. Комп'ютер є досить серйозним джерелом низки алергенів. Приміром, корпус монітора, нагріваючись до 50-55 ° С починає виділяти в повітря пари тріфенілфосфата. Крім монітора нагрівається і материнська плата, блок живлення, процесор, відеокарта, які так само можуть виділяти в навколишнє середовище шкідливі органічні та неорганічні речовини (фтор-, хлор-, фосфоровмісні). Крім того, в комп'ютері є дуже багато місць, де накопичується пил і бруд, розмножуються мікроби і грибки. Пил отримує від екрану монітора слабкий статичний заряд, якого вистачає, що б пил осідав на тілі користувача і в його дихальних шляхах. При алергії підвищується стомлюваність, посилюється дратівливість і знижується імунітет. Алергія провокує ряд захворювань: екзему, гемолітичну анемію, бронхіальну астму та ін Найбільш тяжким проявом алергії є анафілактичний шок, який супроводжується утрудненням дихання, судомами, втратою свідомості, зниженням артеріального тиску і часто смерті.

**4.2 Організація робочого місця з ПК.**

При розміщенні робочих місць з персональними комп'ютерами відстань між робочими столами з відПКоніторами (у напрямі тилу поверхні одного відПКонітора і екрану іншого відПКонітора) повинно бути не менше 2,0 м, а відстань між бічними поверхнями відПКоніторів - не менше 1,2 м.

Робочі місця з персональними комп'ютерами в приміщеннях з джерелами шкідливих виробничих факторів розміщуються в ізольованих кабінах з організованим повітрообміном. При виконанні творчої роботи, що вимагає значного розумового напруження або високої концентрації уваги, робочі місця з персональними комп'ютерами рекомендується ізолювати один від одного перегородками висотою 1,5-2,0 м.

Екран від монітора повинен знаходитися від очей користувача на відстані 600-700 мм, але не ближче 500 мм з урахуванням розмірів алфавітно-цифрових знаків і символів.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання з урахуванням його кількості і конструктивних особливостей, характеру виконуваної роботи. При цьому допускається використання робочих столів різних конструкцій, що відповідають сучасним вимогам ергономіки. Поверхня робочого столу повинна мати коефіцієнт відбиття 0,5-0,7.

Висота робочої поверхні столу для дорослих користувачів повинна регулюватися в межах 680-800 мм; при відсутності такої можливості висота робочої поверхні столу повинна складати 725 мм. Модульними розмірами робочої поверхні столу для ПК, на підставі яких повинні розраховуватися конструктивні розміри, слід вважати: ширину - 800, 1000, 1200 і 1400 мм, глибину - 800 і 1000 мм при нерегульованої його висоті, рівній 725 мм.

Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною - не менше 500 мм, глибиною на рівні колін - не менше 450 мм, на рівні витягнутої ноги - не менше 650 мм.

Конструкція робочого стільця (крісла) повинна забезпечувати підтримку раціональної робочої пози під час роботи на персональному комп'ютері, дозволяти змінювати позу з метою зниження статичного напруження м'язів шейноплечевой області і спини для попередження розвитку втоми. Тип робочого стільця (крісла) слід вибирати з урахуванням зростання користувача, характеру та тривалості роботи з ПК. Робочий стілець (крісло) повинен бути підйомно-поворотним, регульованим по висоті і кутам нахилу сидіння і спинки, а також відстані спинки від переднього краю сидіння, при цьому регулювання кожного параметра повинні бути незалежною, легко здійснюваної мати надійну фіксацію. Поверхня сидіння, спинки та інших елементів стільця (крісла) повинна бути напівм'якої, з нековзним, слабо електризується і повітропроникним покриттям, що забезпечує легке очищення від забруднень.

Робоче місце користувача ПК слід обладнати підставкою для ніг, має ширину не менше 300 мм, глибину не менше 400 мм, регулювання по висоті в межах до 150 мм і за кутом нахилу опорної поверхні підставки до 20 град. Поверхня підставки повинна бути рифленою і мати по передньому краю бортик висотою 10 мм.

Клавіатуру слід розташовувати на поверхні столу на відстані 100-300 мм від краю, зверненого до користувача, або на спеціальній, регульованою по висоті робочої поверхні, відокремленої від основної стільниці.

**4.3 Вимоги до санітарно-гігієнічних параметрів робочих місць.**

Згідно «Гігієнічним критеріям оцінки і класифікації умов праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу», умови праці користувачів персональних комп'ютерів можна віднести до 3-го класу шкідливості допустимим умовам праці. Користувачам необхідно знати елементарні правила і санітарні норми при роботі з комп'ютерами.

Вимоги до зашумленості приміщень.

У виробничих приміщеннях при виконанні основних або допоміжних робіт з використанням ППК рівні шуму на робочих місцях не повинні перевищувати гранично допустимих значень, встановлених для даних видів робіт відповідно до діючих санітарно епідеміологічним нормативам.

При виконанні робіт з використанням ПК у виробничих приміщеннях рівень вібрації не повинен перевищувати допустимих значень вібрації для робочих місць відповідно до згідно санітарно-епідеміологічним нормативам.

У приміщеннях всіх типів освітніх та культурно-розважальних закладів, в яких експлуатуються ПК, рівень вібрації не повинен перевищувати допустимих значень для житлових і загальноміських будівель відповідно до діючих санітарно-епідеміологічними нормативами.

Обладнання яке шумить (друкові пристрої, сервери і т.п.), рівні шуму якого перевищують нормативні, повинне розміщуватися поза приміщень з ПК .

Вимоги до мікроклімату.

Забезпечення здорових та безпечних умов праці у виробничій сфері досягається за рахунок дотримання діючих нормативних документів, для існуючих об'єктів шляхом встановлення таких фактичних значень факторів, які не суперечать нормативним.

Трудова діяльність людини завжди проходить у визначених метеорологічних умовах, які визначаються поєднанням температури повітря, швидкості руху повітря, його відносною вологістю, барометричним тиском, тепловим випромінюванням від нагрітих поверхонь.

Мікроклімат виробничих приміщень або метрологічні умови визначаються такими параметрами:

а) температурою повітря t, ºС;

б) відносною вологістю φ,%;

с) швидкість руху повітря на робочому місці V, м/с.

Всі види робіт в виробничих приміщеннях поділяють на 3 категорії по важкості робіт. Робота оператора ПК відноситься до категорії робіт з затратами енергії до 150 ккал/год, це згідно з ГОСТ 12.1-005-88 відповідає категорії – легка – 1 б.

Параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин на робочих місцях , що комплектуються, повинні відповідати вимогам пункту [ДСаНПіН 3.3.2-007-98](http://dnaop.com/html/40949/doc-%D0%94%D0%A1%D0%B0%D0%9D%D0%9F%D1%96%D0%9D_3.3.2-007-98) «Санітарні норми мікроклімату виробничих приміщень»; ГОСТ 12.1.005-88, ССБГ «Загальні санітарно – гігієнічні вимоги до повітря робочої зони» ,СН 4088-86, СН 2152-80 «Санітарно-гігієнічні норми допустимих рівнів іонізації повітря виробничих та громадських приміщень» і наведені в таблиці 4.1 та таблиці 4.2.

Таблиця 4.1 Нормовані параметри мікроклімату для приміщень з ПК для категорії робіт 1-б.

|  |  |  |  |
| --- | --- | --- | --- |
| Пора року | Оптимальна температура повітря, ºС | Оптимальна відповідна вологість повітря, % | Оптимальна швидкість руху повітря, м/с |
| Холодний | 21-23 | 40-00 | 0,1 |
| Теплий | 22-24 | 40-00 | 0,1 |

Для нормалізації мікроклімату в даному випадку використовують раціональне вентилювання та опалення.

Характеристики наведені в таблиці 4.2 та 4.3.

Таблиця 4.2 Рівні іонізації повітря приміщення при роботі в приміщенні з ПК.

|  |  |  |
| --- | --- | --- |
| Рівні | Кількість іонів в 1 см³ повітря , n | |
| n+ | n- |
| Мінімально необхідні | 400 | 600 |
| Оптимальні | 1500-300 | 3000-5000 |
| Максимально допустимі | 50000 | 50000 |

Таблиця 4.3 Характеристика системи вентиляції.

|  |  |  |  |
| --- | --- | --- | --- |
| Виробниче приміщення | Вид вентиляції | Вентиляційне обладнання | Кратність повітряного обміну , 1/год |
| Цех | Правий витік | Вентилятор  В-Ц 14-46-2,5 | 1-10 |

Вимоги до освітленості.

Відповідно до ДНАОП 0.00-1.31-99 освітлення в приміщенні з ПК має бути суміщеним.

Для оцінки освітлення використовують такі величини: люкс (лк) – рівень освітленості поверхні площею 1 м², на яку падає, рівномірно розподіляючись світловий потік в 1 люмен; люмен - світловий потік від еталонного точкового джерела в одну міжнародну свічку, розташованого в вершині тілесного кута в 1 стерадіан; нит - яскравість поверхні, що світиться і від якої в перпендикулярному напрямку поширюється світло силою в 1 свічку з 1 м²; свічка – сила свічки, як точкового джерела, яке випромінює світловий потік в 1 лм, який рівномірно розподілений в середині тілесного кута в 1 стерадіан.

Відчуття світла при дії на око людини викликають електромагнітні хвилі так званого оптичного діапазону. Область оптичних електромагнітних випромінювань розташована між областю рентгенівського випромінювання та областю радіовипромінювання.

Гігієнічні вимоги до виробничого освітлення основані на психофізичних особливостях сприйняття світла і його впливу на організм людини.

Освітлення в робочих приміщеннях може здійснюватися природнім та штучним освітленням. При недостатньому природному освітленні використовують загальне освітлення. Останнє представляє собою освітлення, при якому водночас використовують природне і штучне освітлення. Загальним називають освітлення, світильники якого освітлюють всю площу приміщення.

Природне освітлення повинно проникати через бічні світло прорізи, зорієнтовані як правило на північ, чи північний схід і забезпечувати коефіцієнт природної освітленості не нижче 1,5%. Розрахунки проводять згідно СНіП 11-4-79.

При виробничій потрібні дозволяється експлуатувати ПК у приміщеннях без природного освітлення за узгодженням з органами державного нагляду за охороною праці та органами і уповноваженими санітарно епідеміологічної служби.

Вікна приміщення повинні мати регульовані пристрої для відкривання та жалюзі, штори, зовнішні шторки. Для запобігання появи відблисків на екранах моніторів забороняється використовувати люмінесцентну лампу типу ЛБ.

Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк. У разі неможливості забезпечити даний рівень освітленості системою загального освітлення припускається застосування

світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану та збільшення освітленості екрану більше ніж до 300 лк.

Також необхідно передбачити обмеження прямої близькості від джерела природного та штучного освітлення при цьому яскравість поверхонь що світиться (вікна, лампи) і перебувають і полі зору повинна бути не більшою за 200 кд/ м². Необхідно обмежувати нерівномірність розподілу яскравості в полі зору осіб , що працюють з потерпілими, при цьому відношення яскравості робочих поверхонь не повинно перевищувати 3:1, а робочих поверхонь і навколишніх предметів ( стін, обладнання) 5:1.

Розрахунок системи освітлення проводиться за допомогою коефіцієнта використання світлового потоку, що рівний відношенню світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп. Його величини залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризується коефіцієнтами відбивання стін і стелі.

Для виключення попадання на екрани дисплеїв прямих світлових потоків світильники загального освітлення встановлюють збоку робочого місця, паралельно лінії зору оператора і стінки з вікнами. Таке розміщення світильників дозволяє встановлювати їх послідовне включення в залежності від величини природної освітленості і виключає подразнення очей смугами світла та тіні, що виникає при поперечному розташуванні світильників.

Розрахунок місцевого світлового потоку виконується, тому що в даному випадку рекомендується система змінного освітлення для запобігання відбитому блиску від поверхні стола і екрану монітору.

Отже для забезпечення нормальних умов роботи, в відповідності з нормативними вимогами, необхідно використовувати дане число світильників визначеної потужності для освітлення робочого приміщення.

**4.4 Пожежна безпека.**

При використанні ПК ймовірна небезпека різного роду загорянь. У сучасних комп'ютерах дуже висока щільність розміщення елементів електронних систем, в безпосередній близькості один від одного розташовуються сполучні дроти,комунікаційні кабелі. При протіканні по них електричного струму виділяється

значна кількість теплоти, що може привести до підвищення температури окремих вузлів до 80-100 ºС. При цьому можливі оплавлення ізоляції сполучних проводів, їх оголення і, як наслідок, коротке замикання, яке супроводжується іскрінням, яке веде до неприпустимих перевантажень елементів електронних схем. Перенагрівання, вони згорають з розбризкуванням іскор.

Для відводу надлишкового тепла від ПК служать системи вентиляції і кондиціонування повітря. Однак ці системи також представляють додаткову пожежну небезпеку для машинного залу та інших приміщень, так як, з одного боку, повітроводи забезпечують подачу кисню, що є окислювачем, в усі приміщення, а з іншого – при виникненні пожежі швидко поширюють вогонь і продукти горіння по всіх приміщеннях і пристроїв, з якими вони пов'язані.

Живлення до електроустановок подається по кабельних лініях, які становлять особливу пожежну небезпеку. Наявність пального ізоляційного матеріалу, ймовірних джерел запалювання в вигляді електричних іскор і дуг, розгалуженість і важко доступність роблять кабельні лінії місцями найбільш ймовірного виникнення і розвитку пожежі.

Експлуатація ПК пов'язана з необхідністю проведення обслуговуючих, ремонтних і профілактичних робіт. При цьому використовують різні мастильні матеріали, легкозаймисті рідини, прокладають тимчасові електропроводки, ведуть пайку та чистку окремих вузлів і деталей. Виникає додаткова пожежна небезпека, яка потребує відповідних заходів пожежної профілактики. Для попередження загоряння всі види кабелів слід прокладати в металевих газонаповнених трубах. У машинних залах кабельні лінії прокладають під технологічними знімними полами, які виконують з негорючих або важко горючих матеріалів з межею вогнестійкості не менше 0,5 год. У приміщеннях обчислювального центру пожежні крани встановлюють в коридорах, на майданчиках сходових клітин, біля входів. ручні вуглекислотні вогнегасники встановлюють в приміщеннях з розрахунку один вогнегасник на 40-50 м2.

У разі пожежі спрацьовує в приміщеннях автоматична установка пожежогасіння. Найчастіше застосовуються газові АУП. АУП забезпечена світловою та звуковою сигналізацією.

Для запобігання поширенню вогню під час пожежі з однієї частини будівлі на іншу влаштовують протипожежні перепони у вигляді протипожежних стін, перегородок, перекриттів, зон, тамбурів-шлюзів, дверей, вікон, люків, клапанів.

У будівлі на випадок виникнення пожежі передбачається не менше двох евакуаційних виходів; але через машинний зал, який має теж менше двох виходів, не повинні проходити шляхи евакуації співробітників, працюючих в інших підрозділах. В інших виробничих приміщеннях допускається проектувати один вихід, якщо відстань від найбільш віддаленого місця до виходу не перевищує 25 м, а кількість працюючих в зміні не більше 25 осіб. Проходи, коридори і робочі місця не слід захаращувати архівними матеріалами, папером. на евакуаційних шляхах встановлюють як природне, так і штучне аварійне освітлення.

Для зберігання носіїв інформації використовуються вогнетривкі металеві шафи, двері в сховище також повинні бути вогнетривкими. Комплекс організаційних і технічних заходів пожежної профілактики дозволяє запобігти пожежі, а в разі його виникнення забезпечити безпеку людей, обмежити поширення вогню, а також створити умови для успішного гасіння пожежі.

 У таких ситуаціях як обрив проводів живлення, несправності заземлення (інших пошкоджень пов'язаних з фізичної справністю ПК), появі гару, потрібно відразу вимкнути живлення. Потім повідомити про виниклу ситуацію керівнику. До усунення несправностей не приступати до роботи. В разі появи задимлення і / або виникнення пожежі необхідно повідомити в пожежну охорону за телефоном «01», повідомити керівника про те, що трапилося, сповістити співробітників (по можливості), повідомити про пожежу / задимлення на пост охорони. Після відкрити запасні виходи в будівлі, відключити електроживлення. Якщо немає ризику для життя, то слід приступити до гасіння пожежі первинними засобами пожежогасіння: вогнегасник, пісок / земля, вода (тільки після відключення електроживлення).

# 5 ЕКОНОМІЧНА ЧАСТИНА

## **5.1 Розрахунок часу на виготовлення програмного продукту.**

Таблиця 5.1 Перелік робіт.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Робота | Тривалість, днів | Виконавець |
| 1 | Написання технічного завдання | 1 | Консультант |
| 2 | Проектування системи класів та залежностей | 3 | Розробник |
| 3 | Налаштування середовища розробки | 1 | Розробник |
| 4 | Проектування бази даних | 3 | Розробник БД |
| 5 | Розробка макету користувацького інтерфейсу | 3 | Дизайнер |
| 6 | Розробка елементів користувацького інтерфейсу | 5 | Дизайнер |
| 7 | Написання основних класів роботи системи | 5 | Розробник |
| 8 | Розробка модулів реєстрації та авторизації користувачів | 2 | Розробник |
| 9 | Розробка модулю «Users» | 15 | Розробник |
| 10 | Розробка модулю «Goods» | 25 | Розробник |
| 11 | Тестування інтерфейсу користувача | 1 | Тестувальник |
| 12 | Тестування функціоналу додатку | 4 | Тестувальник |
| 13 | Написання документації та посібників користувача | 4 | Розробник |

Всього на розробку проекту було витрачено 72 днів. Кількість днів, відпрацьованих кожним виконавцем, зведені в таблиці 5.2.

Таблиця 5.2 Кількість днів відпрацьованих кожним виконавцем.

|  |  |
| --- | --- |
| Працівник | Кількість відпрацьованих днів |
| Розробник | 55 |
| Дизайнер | 8 |
| Тестувальник | 5 |
| Розробник БД | 3 |
| Консультант | 1 |

Для планування і керування розробкою проекту побудовано мережевий діаграм, який відповідає переліку робіт в таблиці 5.1. В мережевому графіку детально показано, що, в якій послідовності, коли, за який час, для чого необхідно виконати, щоб забезпечити закінчення всіх робіт не пізніше заданого терміну.

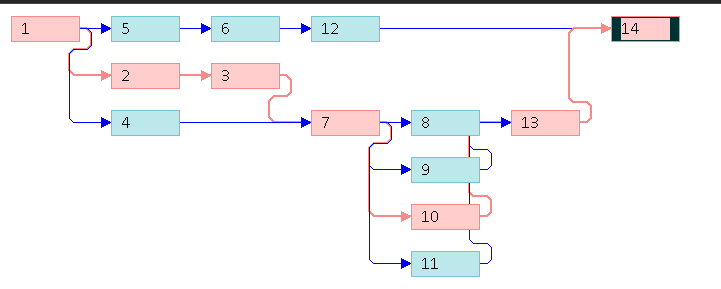


Рисунок 5.1 Мережевий граф.

Критичний шлях – це найбільш тривала по часу послідовність робіт, які ведуть від вихідної до завершальної події. Величина критичного шляху визначає термін виконання всього комплексу планування робіт. Зміна тривалості будь–якої роботи, що лежить на критичному шляху, відповідним чином змінює термін настання завершальної події, тобто дату досягнення кінцевої мети.

Як видно з рис.5.1 критичним шляхом є роботи під номером 1, 2, 3, 7, 10, 12 та 13, вони мають найдовший час виконання.

Розгорнутий мережевий графік з ранніми, пізніми термінами та резервами зображено на рис.5.2 та рис.5.3.

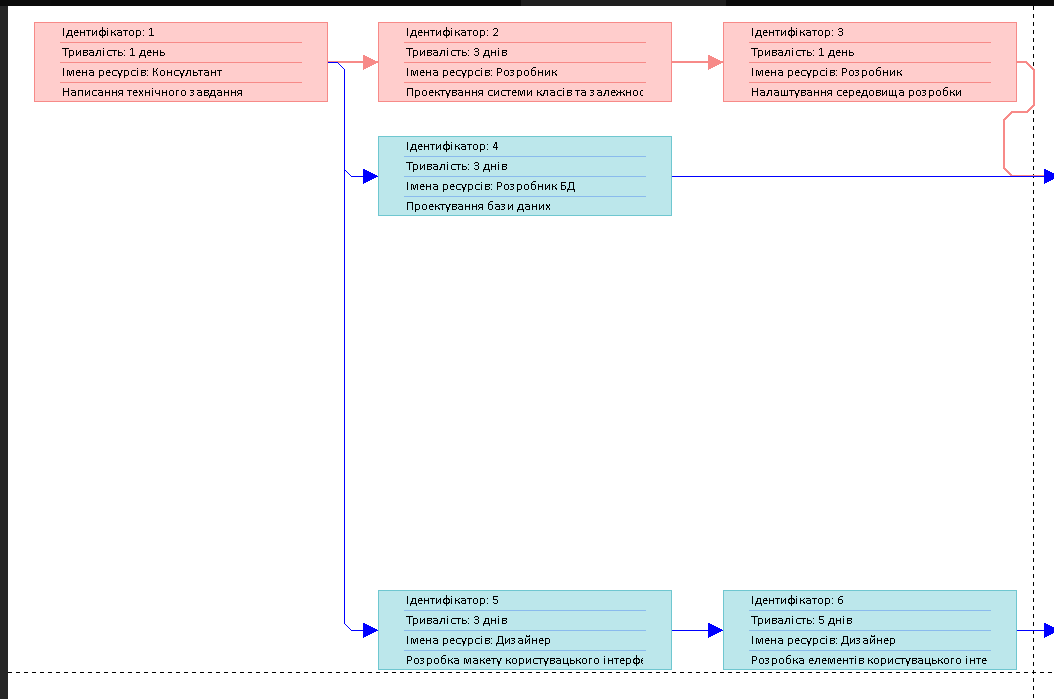


Рисунок 5.2 Розгорнутий мережевий графік.

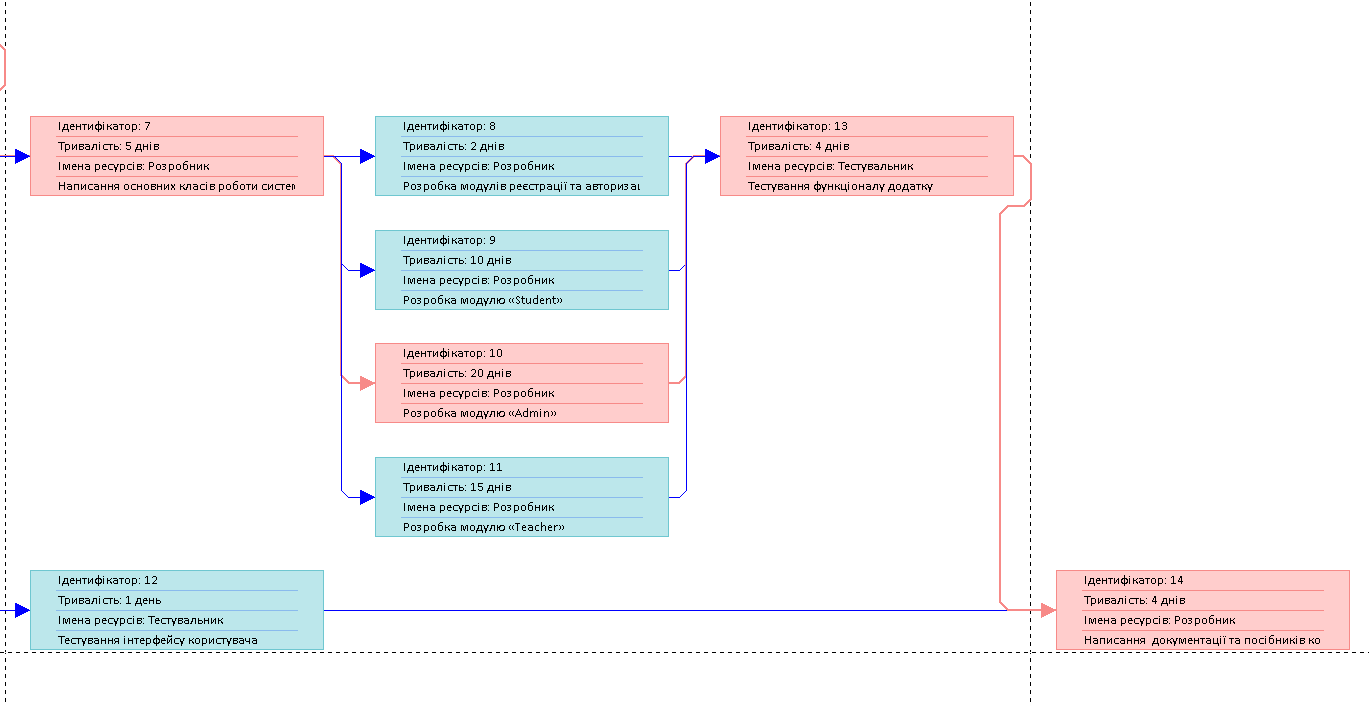


Рисунок 5.3 Продовження розгорнутого мережевого графіку.

Пізній час розробки проекту обчислюється сумою тривалості усіх робіт що лежать на критичному шляху.

Тп = Т1 + Т2 + Т3 + Т7 + Т10 + Т12 + Т13 = 43 (дня).

## **5.2 Економічне обґрунтування розробки та впровадження проектного рішення.**

Для економічного обґрунтування розробки та впровадження проектного рішення обчислюється ряд економічних показників:

* витрати на розробку та впровадження проектного рішення.
* витрати на оплату праці розробників;
* єдиний соціальний внесок;
* вартість додаткових виробів, що закуповуються;
* транспортно-заготівельні витрати;
* накладні витрати;
* інші витрати;
* витрати на відлагодження і дослідну експлуатацію програмного забезпечення на ПК.
* експлуатаційні витрати;
* ціна споживання проектного рішення;
* показники економічної ефективності.

Розрахунок витрат на розробку та впровадження проектного рішення.

Розрахунок витрат на розробку та впровадження проектного рішення, що визначається за формулою (5.1) та включає в себе наступні складові:

витрати на оплату праці розробників ();

єдиний соціальний внесок ();

вартість додаткових виробів, що закуповуються ();

транспортно-заготівельні витрати ()

накладні витрати ();

інші витрати ().

Формула має наступний вигляд:

  (5.1)

де – витрати на розробку програмного засобу, грн.;

– витрати на відлагодження і дослідну експлуатацію програмного забезпечення на ПК, грн.

Визначення витрат на оплату праці та відрахувань на соціальні заходи.

Відповідно до Закону України « Про оплату праці» заробітна плата – це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Для обчислення середньоденної заробітної плати кожного з працівників можна використати наступну формулу:

 (5.2)

де– основна місячна заробітна плата розробника 1-ої спеціальності, грн.;

 – місячний фонд робочого часу, днів (23 дня).

Обчислення витрат на оплату праці усіх розробників проекту можна здійснити за формулою:

 (5.3)

де  – чисельність розробників проекту і-ої спеціальності, чол.;

– час, витрачений на розробку проекту працівником і-ої спеціальності, дні;

 – денна заробітна плата розробника і-ої спеціальності, грн.;

Таблиця 5.3 Зарплата робітників.

|  |  |
| --- | --- |
| Працівник | *ЗПi* |
| Розробник |  |
| Дизайнер |  |
| Тестувальник |  |
| Розробник БД |  |
| Консультант |  |

Тоді витрати на оплату праці усіх розробників проекту становлять:

Додаткова заробітна плата становить 15% від суми основної заробітної плати.

 (5.4)

де, Кдопл. – коефіцієнт додаткових виплат працівникам, 0.15.

Розробник:

Дизайнер:

Тестувальник:

Розробник БД:

Консультант:

Звідси загальні витрати на оплату праці (*Во.п.*) визначаються за формулою:

 (5.5)

Розробник :

Дизайнер:

Тестувальник:

Розробник БД:

Консультант:

Всього:

Витрати на оплату праці працівникам тягнуть за собою додаткові зобов'язання підприємства по сплаті ЄСВ ().

Розробник:

Дизайнер:

Тестувальник:

Розробник БД:

Консультант:

Загальний

Нарахування єдиного внеску складають 22 відсотки до бази нарахування згідно із Законом України «Про внесення змін до Податкового кодексу України та деяких законодавчих актів України щодо забезпечення збалансованості бюджетних надходжень у 2016 році» № 909-VIII від 24 грудня 2015 року.

Проведені розрахунки витрат на оплату праці зведені у таблиці 5.4.

Таблиця 5.4 Зведені розрахунки витрат на оплату праці.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| №п/п | Категорія працівників | Основна заробітна плата, грн. | | | Додаткова заробітна плата, грн. | Нарахування на ФОП, грн. | Всього витрати на оплату праці, грн. |
| Денна заробітна плата розробника, грн. | К-сть відпрацьов. днів | Фактично нарах. з/пл., грн. |
| 1 | Розробник | 333,3 | 55 |  |  |  | 25719,1 |
| 2 | Розробник БД | 333,3 | 3 |  |  |  | 1552,9 |
| 3 | Дизайнер | 291,3 | 5 |  |  |  | 3269,6 |
| 4 | Консультант | 291,3 | 1 | 291,3 |  |  | 408,7 |
| 5 | Тестувальник | 291,3 | 8 |  |  |  | 2043,5 |
| Разом | | 1540,5 | 72 |  | 3511,5 | 5922,7 | 32845 |

Визначення вартості додаткових виробів, що закуповуються.

Витрати на додаткові вироби, що закуповуються () визначаються за їхніми фактичними цінами з урахуванням найменування, номенклатури та необхідної їх кількості в проекті. Вихідні дані та результати розрахунків оформляються у таблиці «Розрахунок витрат на куповані вироби[7].

Таблиця 5.5 Розрахунок витрат на куповані вироби.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Найменування купованих виробів | Марка, тип | Кількість на розробку, шт. | Ціна за одиницю, грн. | Сума витрат, грн. | Сума транспортно-заготівельних витрат, грн. |
| Маркерна дошка | UkrBoards,350х500мм, біла | 1 | 461 | 461 | 50,7 |
| USB-накопичувач | Transcend, 32gb | 1 | 259 | 259 | 28,5 |
| Всього |  | 2 | 720 | 720 | 79,2 |

Визначення транспортно-заготівельних витрат.

Транспортно-заготівельні витрати () для обрахунку даного типу витрат складають 11% від суми придбання купованих виробів.

Визначення накладних витрат.

Накладнi витрати () проектних органiзацiй передбачають витрати на управлiння, загальногосподарськi, невиробничi витрати. Вони становлять 25% витрат на оплату працi.

Визначення інших витрат.

Iншi витрати () — це витрати, якi не врахованi в попереднiх статтях витрат. Їх розраховують за встановленими вiдсотками до витрат на оплату працi (6%).

Визначення витрат на розробку проектного рішення.

Після вирахування всіх витрат, можна виконати підрахунок витрат на розробку проектного рішення, використовуючи формулу:

 (5.6)

Визначення витрат на відлагодження і дослідну експлуатацію програмного забезпечення.

Витрати на відлагодження і дослідну експлуатацію програмного забезпечення визначають згідно з формулою:

 (5.7)

де – вартість однієї години роботи ПК, грн./год.

– кількість годин роботи ПК на відлагодження програми, год.

Загальна кількість днів роботи на комп’ютері рівна 77. Середній щоденний час роботи на ПК - 2 год., тому:

Експлуатаційні витрати, що припадають на 1год.машинного часу, можуть бути визначені за витратою електроенергії:

(5.8)

де  *=* 340 Вт - споживана потужність ПК;

= 5 - вартість 1 кВт/год електроенергії для підприємства.

Отже, за формулою (5.8)

ег =0.34\*5 = 1.7 грн/год.

Тоді:

Виходячи з формули розрахунок витрат на розробку та впровадження проектного рішення отримуємо наступні результати:

Підготовка кошторису на розробку проектного рішення.

Результати розрахунків для формування кошторису зведені у таблиці 5.6.

Таблиця 5.6 Кошторис витрат на розробку проектного рішення.

|  |  |
| --- | --- |
| Найменування елементів витрат | Сума витрат, грн. |
| Витрати на розробку проектного рішення, у т.ч.: |  |
| Витрати на оплату праці |  |
| Єдиний соціальний внесок |  |
| Витрати на додаткові вироби, що закуповуються | 720 |
| Транспортно-заготівельні витрати |  |
| Накладні витрати |  |
| Інші витрати |  |
| Витрати на відлагодження і дослідну експлуатацію системи |  |
| Всього |  |

Визначення експлуатаційних витрат.

Визначення одноразових експлуатаційних витрат проектного рішення та аналогу.

При порівнянні програмних засобів в експлуатаційні витрати включають вартість підготовки даних () і вартість годин роботи ПК (). Одноразові експлуатаційні витрати визначаються за формулою:

 (5.9)

Вартість підготовки даних для експлуатації проектного рішення (аналогу) () визначаються за формулою:

 (5.10)

Середньогодинна ставка робітника визначається за формулою:

 (5.11)

b – коефіцієнт відрахувань на соціальні заходи, становить 0.22;

Ф – фонд робочого часу (год).

Отже:

Середньогодинна ставка робітників над проектним рішенням:

Системний адміністратор:

В аналога діє щомісячний орендний тариф на службу підтримки**:**

Служба підтримки: вартість оренди служби підтримки – 3000грн/міс.

Вартість підготовки даних для експлуатації проектного рішення (аналогу):

Витрати на відлагодження і дослідну експлуатацію аналога становлять:

=340 Вт - споживана потужність ПК;

=5 - вартість 1 кВт/год електроенергії для споживачів;

- кількість годин роботи ПК.

ег =0.34\*5 = 1,7 грн/год.

Одноразові експлуатаційні витрати на розробку становлять:

Визначення річних експлуатаційних витрат проектного рішення та аналогу.

Експлуатаційні витрати – витрати виробництва, пов'язані з підтриманням у працездатному стані використовуваних систем, машин та устаткування. В інформаційній сфері це поточні витрати об’єкту необхідні для надання обсягу послуг за певний період часу (рік, квартал)[8].

Річні експлуатаційні витрати визначаються за формулою:

 (5.12)

Де *N А* = 12 (міс.);

Вихідні дані та результати розрахунків витрат на підготовку даних для експлуатації проектного рішення зведені у таблицю 5.7.

Таблиця 5.7 Розрахунок витрат на підготовку даних для роботи на ПК

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категорія персоналу | Чисельність співробітників  і-ої категорії, чол. | Час роботи співробітників  і-ої категорії, год. | Середньогодинна ЗП співробітника і-ої категорії, грн. | Витрати на підготовку даних, грн. |
| Проектне рішення | | | | |
| Розробник | 1 | 254 | 44,4 | 44,4 |
| Всього | 1 | 254 | - | 11709,4 |
| Категорія персоналу | Чисельність співробітниківі-ої категорії, чол. | Тривалість оренди, міс. | Вартість оренди в місяць, грн. | Витрати на підготовку даних, грн. |
| Розробник | 1 | 12 | 4000 | 4000 |
| Всього | 1 | 12 | 4000 | 48000 |

Проектне рішення було розроблено із меншими фінансовими затратами порівняно із досліджуваним аналогом, тому можна зробити висновок, що розроблена модель є більш рентабельною і доступнішою ніж аналог.

Розрахунок ціни споживання проектного рішення.

Ціна споживання () – це витрати на придбання і експлуатацію проектного рішення за весь строк його служби:

 (5.13)

де – ціна придбання проектного рішення, грн.;

 – теперішня вартість витрат на експлуатацію проектного рішення (за весь час його експлуатації), грн.

Визначення ціни придбання проектного рішення.

 (5.14)

де Р – норматив рентабельності 26%;

капітальні витрати;

 – витрати на прив'язку та освоєння проектного рішення на конкретному об’єкті, грн.;

– витрати на доукомплектування технічних засобів на об'єкті, грн.;

Визначення теперішньої вартості витрат на експлуатацію проектного рішення (за весь час його експлуатації).

Теперішня вартість витрат на експлуатацію проектного рішення розраховується за формулою:

** (5.15)

де – річні експлуатаційні витрати в t-ому році, грн.

При умові, що впродовж всього строку експлуатації , тоді:

 (5.16)

де pv – ставка дисконта на період Т(приT=4), яка визначається залежно від процентної ставки R(при R=0,14) і періоду експлуатації Т.

Таблиця 5.8 Теперішня вартість витрат на експлуатацію.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Т* | 0 | 1 | 2 | 3 | 4 |
|  | 1 | 0,88 | 0,77 | 0,68 | 0,59 |

PV = 0,88 + 0,77 + 0,68 + 0,59 = 2,92.

Тоді:

+ 11709,4\* 2,92 = 81611,2 грн.

53000 + 48000 \* 4 = 245000 грн.

Визначення показників економічної ефективності.

Економічний ефект в сфері експлуатації (грн.):

 (5.17)

Економічний ефект в сфері проектування (грн.):

 (5.18)

Отже Епр>0 та Еекс>0, тоді розраховується:

Додатковий економічний ефект в сфері експлуатації (грн.):

 (5.19)

Додатковий економічний ефект в сфері проектування (грн.):

 (5.20)

Термін окупності витрат на проектування рішення (років):

; (5.21)

Результуючі показники економічної ефективності зводяться у таблицю.

Таблиця 5.9 Показники економічної ефективності проектного рішення.

|  |  |  |  |
| --- | --- | --- | --- |
| Найменування показників | Одиниці вимірювання | Значення показників | |
| Аналог | Проектне рішення |
| 1. Капітальні вкладення | грн. | - |  |
| 2. Ціна придбання | грн. | 53000 |  |
| 3. Річні експлуатаційні витрати | грн. | 48000 | 11709,4 |
| 4. Ціна споживання | грн. | 245000 | 81611,2 |
| 5. Економічний ефект в сфері експлуатації | грн. | - |  |
| 6. Додатковий економічний ефект в сфері експлуатації | грн. | - |  |
| 7. Економічний ефект в сфері проектування | грн. | - |  |
| 8. Додатковий економічний ефект в сфері проектування | грн. | - |  |
| 9. Термін окупності витрат на проектування рішення | місяці | - | 1 |

Висновок.

В результаті проведеного економічного аналізу проектного рішення було розраховано витрати на розробку, ціну придбання, експлуатаційні витрати, економічний ефект та термін окупності. Значення були порівняні із значеннями існуючого аналогу, що довело переваги над готовими рішеннями.

ВИСНОВКИ

Результатом виконання дипломного проєкту було розроблено програмного застосунку прокладання маршрутів та оптимізації процесу доставлення товарів. За допомогою застосунку можна прокласти маршрут для доставлення товарів та також оптимізувати доставлення.

В проєкт проведено дослідження щодо існуючих програмних рішень для прокладання маршрутів та оптимізації процесу доставлення товарів, розглянуто різні підходи та методи розробки подібних застосунків. Було обрано оптимальну технологію для реалізації системи.

Було розв'язувати наступні задачі:

* ознайомлення з програмними застосунками для прокладання маршрутів.
* дослідження алгоритм оптимізації маршрутів та компонентів сучасних систем програмних застосунків;
* ознайомлення з сучасними технологіями та засобами розробки програмних застосунків;
* вивчення засобів та програм розробки програмних застосунків, налаштування середовища розробки та системи керування версіями;
* ознайомлення з методами зображення різнотипної інформації(тексту, зображень, тощо) в програмному застосунку;
* вивчення принципів розробки алгоритмів оптимізації маршрутів;
* розроблення програмного застосунку прокладання маршрутів та оптимізації процесу доставлення товарів;

Під час роботи було проаналізовано принципи побудови систем управління даними та розроблено модульну структуру програмного застосунку.

В результаті проведених робіт над проєкт було розроблено програмний застосунок готовий до використання  користувачами.

При розробці програмного застосунку закріплено знання зі створенням застосунків з такими мовами як: HTML, SCSS, JavaScript, TypeScript, React, React Native,  та роботою з базами даних FireStore.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Градi Буч. Об'єктно-орієнтований аналіз і проектування. Біном, СПБ, 2000. -560с.
2. Веб-сайт режим доступу: <https://www.ubereats.com> – 17.03.2023.
3. Веб-сайт режим доступу: <https://glovoapp.com/ua/uk/> - 19.03.2023.
4. Веб-сайт режим доступу: <https://novaposhta.ua> - 19.03.2023.
5. Стівен Хольцнер HTML5 за 10 хвилин [5-е видання] [Пер. з англ.] / М .: Видавничий дім «Вільямс», 2011. - 240 с.
6. Усенко Н. М. 'Економіка та організація компанії IT-індустрії': опорний конспект [Електронний ресурс] / Н. М. Усенко. – Режим доступу : http://www.ikt. hneu.edu.ua.
7. Лазарєва С. Ф. Економіка та організація інформаційного бізнесу : навч. посібн. / С. Ф. Лазарєва. – К. : КНЕУ, 2002. – 213 с.
8. Усенко Н. М. 'Економіка та організація компанії IT-індустрії': опорний конспект [Електронний ресурс] / Н. М. Усенко. – Режим доступу : http://www.ikt. hneu.edu.ua.
9. Дженніфер Нідерст Роббінс «Навчання веб-дизайну»;
10. Бибо Б., Кац И.,«jQuery. Подробное руководство по продвинутому JavaScript».
11. Р. Никсон, «Создаем динамические веб-сайты с помощью MySQL, Javascript и CSS», 3-е изд. (2016).
12. Автором є команда TypeScript  «TypeScript Довідник»;
13. Чарльз Белл, Мэтс Киндал, Ларс Талманн, «Обеспечение высокой доступности систем на основе MySQL».
14. Бэрон Шварц, Петр Зайцев, Вадим Ткаченко, Джереми Заводны, Арьен Ленц, Дерек Боллинг, «MySQL. Оптимизация производительности».
15. Максим Кузнецов, Игорь Симдянов, «Самоучитель MySQL 5».
16. Рэнди Джей Яргер, Джордж Риз, Тим Кинг, «MySQL и mSQL. Базы данных для небольших предприятий и Интернета».
17. Марійн Гавербеке «Красномовний JavaScript»;
18. Эрик Мейер, «CSS. Каскадные таблицы стилей. Подробное руководство».
19. Бер Бибо, Иегуда Кац, jQuery. «Подробное руководство по продвинутому JavaScript».
20. Герд Вагнер «Создание клиентских веб-приложений на простом JavaScript»
21. Організація баз даних: практичний курс: Навч. посіб. для студ. / А. Ю. Берко, О. М. Верес; Нац. ун-т «Львів. політехніка». — Л., 2003. — 149 c.
22. Сотниченко В.М. Економіка програмного забезпечення./ Сотниченко В.М. – Київ: ДержавнийУніверситетТелекомунікацій, 2016. – 43с.
23. Калбертсон Роберт, Браун Крис, Кобб Гэри. Швидке тестування. — М.: «Вильямс», 2002. – 374 с.
24. Борис Чорний «Професійний TypeScript. Розробка масштабованих JavaScript-додатків» / М .: Видавничий дім «Print2print», 2021. - 352 с.
25. Ден Махарр «TypeScript розкрито» / М .: Видавничий дім «Apress», 2013. – 104 с.
26. Кайл Сімпсон «Ви не знаєте JS. Типи та граматичні конструкції» / М .: Видавничий дім «Пітер Прес», 2019. – 240 с.
27. Девід Фланаган «JavaScript. Детальний посібник» / М .: Видавничий дім «Пітер Прес», 2013. – 1081 с.
28. Ден Вандеркам «Ефективний TypeScript: 62 способи покращити код» / М .: Видавничий дім «Print2print», 2020. - 288 с.

# Додатки

**Додаток А**

**А.2 App.js**

import 'react-native-gesture-handler';

import React, {useEffect} from 'react';

import {StatusBar, PermissionsAndroid, Platform} from 'react-native';

import Geolocation from '@react-native-community/geolocation';

import { withAuthenticator } from 'aws-amplify-react-native'

import Router from './src/navigation/Root';

navigator.geolocation = require('@react-native-community/geolocation');

import Amplify from 'aws-amplify'

import config from './aws-exports'

Amplify.configure(config)

const App: () => React$Node = () => {

  const androidPermission = async () => {

    try {

      const granted = await PermissionsAndroid.request(

        PermissionsAndroid.PERMISSIONS.ACCESS\_FINE\_LOCATION,

        {

          title: "Uber App Camera Permission",

          message:

            "App needs access to your location " +

            "so you can take awesome rides.",

          buttonNeutral: "Ask Me Later",

          buttonNegative: "Cancel",

          buttonPositive: "OK"

        }

      );

      if (granted === PermissionsAndroid.RESULTS.GRANTED) {

        console.log("You can use the location");

      } else {

        console.log("Location permission denied");

      }

    } catch (err) {

      console.warn(err);

    }

  }

  useEffect(() => {

    if (Platform.OS === 'android') {

      androidPermission();

    } else {

      // IOS

      Geolocation.requestAuthorization();

    }

  }, [])

  return (

    <>

      <StatusBar barStyle="dark-content" />

      <Router />

    </>

  );

};

export default withAuthenticator(App);

**А.2 CovidMessage.js**

import React from "react";

import { View, Text } from "react-native";

import styles from './styles';

const CovidMessage = (props) => {

  return (

    <View style={styles.container}>

      <Text style={styles.title}>Travel only if necessary</Text>

      <Text style={styles.text}>

        Upgrading this package often requires the font files linked to your projects to be updated as well. If the automatic linking works for you, running this again should update the fonts.

      </Text>

      <Text style={styles.learnMore}>Learn more</Text>

    </View>

  );

};

export default CovidMessage;

А.3 HomeMap.js

import React, { useState, useEffect } from "react";

import { Image } from "react-native";

import MapView, {PROVIDER\_GOOGLE, Marker} from 'react-native-maps';

import {API, graphqlOperation} from 'aws-amplify';

import { listCars } from '../../graphql/queries';

// import cars from '../../assets/data/cars';

const HomeMap = (props) => {

  const [cars, setCars] = useState([]);

  useEffect(() => {

    const fetchCars = async () => {

      try {

        const response = await API.graphql(

          graphqlOperation(

            listCars

          )

        )

        setCars(response.data.listCars.items);

      } catch (e) {

        console.error(e);

      }

    };

    fetchCars();

  }, [])

  const getImage = (type) => {

    if (type === 'UberX') {

      return require('../../assets/images/top-UberX.png');

    }

    if (type === 'Comfort') {

      return require('../../assets/images/top-Comfort.png');

    }

    return require('../../assets/images/top-UberXL.png');

  };

  return (

    <MapView

      style={{width: '100%', height: '100%'}}

      provider={PROVIDER\_GOOGLE}

      showsUserLocation={true}

      initialRegion={{

        latitude: 28.450627,

        longitude: -16.263045,

        latitudeDelta: 0.0222,

        longitudeDelta: 0.0121,

      }}>

      {cars.map((car) => (

        <Marker

          key={car.id}

          coordinate={{latitude: car.latitude, longitude: car.longitude}}

        >

          <Image

            style={{

              width: 70,

              height: 70,

              resizeMode: 'contain',

              transform: [{

                rotate: `${car.heading}deg`

              }]

            }}

            source={getImage(car.type)}

          />

        </Marker>

      ))}

    </MapView>

  );

};

export default HomeMap;

А.4 HomeSearch.js

import React from "react";

import { View, Text, Pressable } from "react-native";

import AntDesign from "react-native-vector-icons/AntDesign";

import MaterialIcons from "react-native-vector-icons/MaterialIcons";

import Entypo from "react-native-vector-icons/Entypo";

import {useNavigation} from '@react-navigation/native'

import styles from './styles.js';

const HomeSearch = (props) => {

  const navigation = useNavigation();

  const goToSearch = () => {

    navigation.navigate('DestinationSearch')

  }

  return (

    <View>

      {/\*  Input Box \*/}

      <Pressable onPress={goToSearch} style={styles.inputBox}>

        <Text style={styles.inputText}>Where To?</Text>

        <View style={styles.timeContainer}>

          <AntDesign name={'clockcircle'} size={16} color={'#535353'} />

          <Text>Now</Text>

          <MaterialIcons name={'keyboard-arrow-down'} size={16} />

        </View>

      </Pressable>

      {/\* Previous destination \*/}

      <View style={styles.row}>

        <View style={styles.iconContainer}>

          <AntDesign name={'clockcircle'} size={20} color={'#ffffff'} />

        </View>

        <Text style={styles.destinationText}>Spin Nightclub</Text>

      </View>

      {/\* Home destination \*/}

      <View style={styles.row}>

        <View style={[styles.iconContainer, {backgroundColor: '#218cff'}]}>

          <Entypo name={'home'} size={20} color={'#ffffff'} />

        </View>

        <Text style={styles.destinationText}>Spin Nightclub</Text>

      </View>

    </View>

  );

};

export default HomeSearch;

А.5 OrderMap.js

import React from "react";

import { Image } from "react-native";

import MapView, {PROVIDER\_GOOGLE, Marker} from 'react-native-maps';

const OrderMap = ({ car }) => {

  const getImage = (type) => {

    if (type === 'UberX') {

      return require('../../assets/images/top-UberX.png');

    }

    if (type === 'Comfort') {

      return require('../../assets/images/top-Comfort.png');

    }

    return require('../../assets/images/top-UberXL.png');

  };

  return (

    <MapView

      style={{width: '100%', height: '100%'}}

      provider={PROVIDER\_GOOGLE}

      showsUserLocation={true}

      initialRegion={{

        latitude: 28.450627,

        longitude: -16.263045,

        latitudeDelta: 0.0222,

        longitudeDelta: 0.0121,

      }}>

      {car && (<Marker

        coordinate={{latitude: car.latitude, longitude: car.longitude}}

      >

        <Image

          style={{

            width: 70,

            height: 70,

            resizeMode: 'contain',

            transform: [{

              rotate: `${car.heading}deg`

            }]

          }}

          source={getImage(car.type)}

        />

      </Marker>)}

    </MapView>

  );

};

export default OrderMap;

**А.6 RouteMap.js**

import React from "react";

import MapView, {PROVIDER\_GOOGLE, Marker} from 'react-native-maps';

import MapViewDirections from 'react-native-maps-directions';

const GOOGLE\_MAPS\_APIKEY = 'AIzaSyDFhFUaYpyAjNE4Eq-sWCGWjrr6kyGnhbQ';

const RouteMap = ({ origin, destination }) => {

  const originLoc = {

    latitude: origin.details.geometry.location.lat,

    longitude: origin.details.geometry.location.lng,

  };

  const destinationLoc = {

    latitude: destination.details.geometry.location.lat,

    longitude: destination.details.geometry.location.lng,

  };

  return (

    <MapView

      style={{width: '100%', height: '100%'}}

      provider={PROVIDER\_GOOGLE}

      showsUserLocation={true}

      initialRegion={{

        latitude: 28.450627,

        longitude: -16.263045,

        latitudeDelta: 0.0222,

        longitudeDelta: 0.0121,

      }}>

      <MapViewDirections

        origin={originLoc}

        destination={destinationLoc}

        apikey={GOOGLE\_MAPS\_APIKEY}

        strokeWidth={5}

        strokeColor="black"

      />

      <Marker

        coordinate={origin}

        title={'Origin'}

      />

      <Marker

        coordinate={destination}

        title={"Destination"}

      />

    </MapView>

  );

};

export default RouteMap;

**А.7 TypeRow.js**

import React from "react";

import { View, Image, Text, Pressable } from "react-native";

import styles from './styles.js';

import Ionicons from "react-native-vector-icons/Ionicons";

const UberTypeRow = (props) => {

  const {type, onPress, isSelected} = props;

  const getImage = () => {

    if (type.type === 'UberX') {

      return require('../../assets/images/UberX.jpeg');

    }

    if (type.type === 'Comfort') {

      return require('../../assets/images/Comfort.jpeg');

    }

    return require('../../assets/images/UberXL.jpeg');

  }

  return (

    <Pressable

      onPress={onPress}

      style={[styles.container, {

        backgroundColor: isSelected ? '#efefef' : 'white',

      }]}

    >

      {/\*  Image \*/}

      <Image

        style={styles.image}

        source={getImage()}

      />

      <View style={styles.middleContainer}>

        <Text style={styles.type}>

          {type.type}{' '}

          <Ionicons name={'person'} size={16} />

          3

        </Text>

        <Text style={styles.time}>

          8:03PM drop off

        </Text>

      </View>

      <View style={styles.rightContainer}>

        <Ionicons name={'pricetag'} size={18} color={'#42d742'} />

        <Text style={styles.price}>est. ${type.price}</Text>

      </View>

    </Pressable>

  );

};

export default TypeRow;

**А.8 Types.js**

import React from "react";

import { View, Text, Pressable } from "react-native";

import styles from './styles.js';

import UberTypeRow from '../UberTypeRow';

import typesData from '../../assets/data/types';

const UberTypes = ({ typeState, onSubmit }) => {

  const [selectedType, setSelectedType] = typeState;

  return (

    <View>

      {typesData.map((type) => (

        <UberTypeRow

          type={type}

          key={type.id}

          isSelected={type.type === selectedType}

          onPress={() => setSelectedType(type.type)}

        />

      ))}

      <Pressable onPress={onSubmit} style={{

        backgroundColor: 'black',

        padding: 10,

        margin: 10,

        alignItems: 'center',

      }}>

        <Text style={{color: 'white', fontWeight: 'bold'}}>

          Confirm Uber

        </Text>

      </Pressable>

    </View>

  );

};

export default UberTypes;

А.9 CustomDrawer.js

import React from "react";

import { View, Text, Pressable } from "react-native";

import {DrawerContentScrollView, DrawerItemList} from '@react-navigation/drawer';

import { Auth } from 'aws-amplify';

const CustomDrawer = (props) => {

  return (

    <DrawerContentScrollView {...props}>

      <View style={{backgroundColor: '#212121', padding: 15}}>

        {/\* User Row \*/}

        <View style={{

          flexDirection: 'row',

          alignItems: 'center',

        }}>

          <View style={{

            backgroundColor: '#cacaca',

            width: 50,

            height: 50,

            borderRadius: 25,

            marginRight: 10,

          }}/>

          <View>

            <Text style={{color: 'white', fontSize: 24}}>Vadim Savin</Text>

            <Text style={{color: 'lightgrey'}}>5.00 \*</Text>

          </View>

        </View>

        {/\* Messages Row \*/}

        <View style={{

          borderBottomWidth: 1,

          borderBottomColor: '#919191',

          borderTopWidth: 1,

          borderTopColor: '#919191',

          paddingVertical: 5,

          marginVertical: 10,

        }}>

          <Pressable

            onPress={() => {console.warn('Messages')}}>

            <Text style={{color: '#dddddd', paddingVertical: 5,}}>Messages</Text>

          </Pressable>

        </View>

        { /\* Do more \*/}

        <Pressable

          onPress={() => {console.warn('Make Money Driving')}}>

          <Text style={{color: '#dddddd', paddingVertical: 5,}}>Do more with your account</Text>

        </Pressable>

        {/\* Make money \*/}

        <Pressable onPress={() => {console.warn('Make Money Driving')}}>

          <Text style={{color: 'white', paddingVertical: 5}}>Make money driving</Text>

        </Pressable>

      </View>

      <DrawerItemList {...props} />

      {/\* Make money \*/}

      <Pressable onPress={() => { Auth.signOut() }}>

        <Text style={{padding: 5, paddingLeft: 20}}>Logout</Text>

      </Pressable>

    </DrawerContentScrollView>

  );

};

export default CustomDrawer;

**А.10 Home.js**

import React from "react";

import HomeScreen from "../screens/HomeScreen";

import { createStackNavigator } from '@react-navigation/stack'

import DestinationSearch from "../screens/DestinationSearch";

import SearchResults from "../screens/SearchResults";

import OrderScreen from "../screens/OrderScreen";

const Stack = createStackNavigator();

const HomeNavigator = (props) => {

  return (

    <Stack.Navigator

      screenOptions={{

        headerShown: false,

      }}

    >

      <Stack.Screen name={"Home"} component={HomeScreen} />

      <Stack.Screen name={"DestinationSearch"} component={DestinationSearch} />

      <Stack.Screen name={"SearchResults"} component={SearchResults} />

      <Stack.Screen name={"OrderPage"} component={OrderScreen} />

    </Stack.Navigator>

  );

};

export default HomeNavigator;

А.11 Root.js

import React from "react";

import {View, Text} from 'react-native';

import { NavigationContainer } from '@react-navigation/native';

import { createStackNavigator } from '@react-navigation/stack'

import { createDrawerNavigator } from "@react-navigation/drawer";

import HomeNavigator from "./Home";

import CustomDrawer from "./CustomDrawer";

const Drawer = createDrawerNavigator();

const DummyScreen = (props) => (

  <View style={{flex: 1, justifyContent: 'center', alignItems: 'center'}}>

    <Text>{props.name}</Text>

  </View>

)

const RootNavigator = (props) => {

  return (

    <NavigationContainer>

      <Drawer.Navigator drawerContent={

        (props) => (

          <CustomDrawer {...props} />)

      }>

        <Drawer.Screen name="Home" component={HomeNavigator} />

        <Drawer.Screen name="Your Trips">

           {() => <DummyScreen name={"Your Trips"} />}

        </Drawer.Screen>

        <Drawer.Screen name="Help">

          {() => <DummyScreen name={"Help"} />}

        </Drawer.Screen>

        <Drawer.Screen name="Wallet">

          {() => <DummyScreen name={"Wallet"} />}

        </Drawer.Screen>

        <Drawer.Screen name="Settings">

          {() => <DummyScreen name={"Settings"} />}

        </Drawer.Screen>

      </Drawer.Navigator>

    </NavigationContainer>

  );

};

export default RootNavigator;

**A.11 MainActivity.java**

package com.driverapp;

import com.facebook.react.ReactActivity;

public class MainActivity extends ReactActivity {

/\*\*

\* Returns the name of the main component registered from JavaScript. This is used to schedule

\* rendering of the component.

\*/

@Override

protected String getMainComponentName() {

return "DriverApp";

}

}

**A.12 App-test.js**

\* @format

import 'react-native';

import React from 'react';

import App from '../App';

// Note: test renderer must be required after react-native.

import renderer from 'react-test-renderer';

it('renders correctly', () => {

renderer.create(<App />);

});