КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

Кваліфікаційна робота

На здобуття ступеня магістра

за освітньо-науковою програмою «Інформатика»

спеціальності 122 «Комп'ютерні науки»

на тему:

ПЛАТФОРМА ДЛЯ МІЖНАРОДНОЇ ДОСТАВКИ ПОСИЛОК МАНДРІВНИКАМИ

Виконав студент 2-го курсу магістратури
Михайлов Нікіта Олегович _____

Науковий керівник:
к.ф.-м.н., доцент
Омельчук Людмила Леонідівна _____

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Роботу розглянуто і допущено до захисту на засіданні кафедри теорії та технології програмування
«____» _____ 201_ р.,
протокол №____
Завідувач кафедри
М. С. Нікітченко

Київ – 2021

РЕФЕРАТ

Дипломна робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел (32 найменування) та 8 додатків.

Робота містить 20 рисунків. Загальний обсяг роботи становить 50 сторінок, основний текст роботи викладено на 40 сторінках.

ПЛАТФОРМА ДЛЯ МІЖНАРОЖДНОЇ ДОСТАВКИ, ДОСТАВКА ПОСИЛОК МАНДРІВНИКАМИ, SPRING, POSTRESQL, STRIPE, HIBERNATE.

Об'єктом роботи є доставка товарів через онлайн платформу. Предметом роботи є застосунок для інтернаціональної доставки посилок мандрівниками.

Метою роботи є дослідження сучасних веб-технологій та створення застосунку який дозволяє мандрівникам доставляти посилки подорожуючи.

В роботі використано такі методи розроблення, як розробка програмного продукту на основі еволюційної моделі та командна розробка продукту. Інструменти розроблення: Spring Framework, Hibernate, реляційна система керування базами даних PostgreSQL, сховище медіа файлів Cloudinary, платіжна систему Stripe, сервіс відправки СМС-повідомлень Vonage та технологія JWT.

Результати роботи: імплементовано аплікацію для міжнародної доставки посилок мандрівниками. А також наведено огляд сучасних технологій програмування (Spring, PostreSQL, Hibernate, JWT).

ЗМІСТ

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

AES – Advanced encryption standart (Розширений стандарт шифрування);

API – Application programming interface (Інтерфейс програмних застосунків);

CIDR – Classless Inter-Domain Routing (Безкласова маршрутизація);

DML – Data manipulation language (Мова маніпуляції даними);

ERD – Entity-relationship diagram (Діаграма зв'язку сутностей);

EE – Enterprise Edition (Корпоративне видання);

GNU – GNU's Not UnixDOM;

IP – Internet Protocol (Інтернет протокол);

iOS – I Operation System (Операційна система «і»);

JDBC – Java Database Connectivity (З'єднання з БД на Джава);

JSON – JavaScript Object Notation (Запис об'єктів ДжаваСкрипт);

JSP– Java Server Pages (Серверні сторінки Джава);

JWT – JSON Web Token (Джейсон веб токен);

MAC - Media Access Control (Управління доступом до носія);

MVC – Model-view-controller (Модель-представлення-контроллер);

MMS – Multimedia Messaging Service (Послуга мультимед. повідомлень);

ORM – Object-relational mapping (Об'єктно реляційна проекція);

POJO – Plain Old Java Objects (Простий старий Джава об'єкт);

REST – Representational State Transfer (Передача репрезентативного стану);

SaaS – Software as a Service (Програма як послуга);

SMS – Short Messaging Service (Служба коротких повідомлень);

SQL – Structured Query Language (Мова структурованих запитів);

UI – User Interface (Інтерфейс користувача);

UX – User Experience (Досвід користувача);

UML – Unified Modeling Language (Уніфікована мова моделювання);

URL – Uniform Resource Locator (Уніфікована адреса ресурсу);

XML – Extensible Markup Language (Розширювана мова розмітки);

БД – база даних;

ОС – операційна система;

СКБД – система керування базами даних.

ВСТУП

Оцінка сучасного стану об’єкта розробки. На сьогоднішній день із розвитком інформаційних технологій і ростом мобільності людей з’являється потреба у швидкій та фінансово рентабельній доставці посилок. На ринку існують широко відомі компанії гіганти як DHL [1], Hermes [2], FedEx [3] тощо. Але терміни доставки та ціна яка збільшується прямопропорційно її швидкості можуть бути нераціональними. Тому з метою покращення системи відповідно до потреб клієнтів і задоволення індивідуальних критеріїв користувачів виникла необхідність розробки нового продукту.

Актуальність роботи та підстави для її виконання. Розроблений в рамках дипломної роботи програмний продукт «Evopost» дозволяє доставити посилку дешево і у короткі терміни, а мандрівникам – зекономити на подорожі. Застосунок призначений для використання широкою цільовою аудиторією, починаючи зі студентів, що навчаються закордоном, закінчуючи робітниками, які прямують у відрядження.

Таким чином, розробка застосунку, який дозволяє мандрівникам доставляти посилки подорожуючи є **актуальною** темою.

Об'єкт дослідження – процес доставки товарів через онлайн платформу, **предмет дослідження** – застосунок для інтернаціональної доставки посилок.

В роботі використано такі **методи дослідження**, як спостереження, порівняння, аналіз та синтез. Були застосовані такі **методи розробки**, як розробка програмного продукту на основі еволюційної моделі та командна розробка продукту.

Мета і задачі дослідження. В рамках роботи над дипломною роботою перед автором була поставлена мета розробити застосунок для міжнародної доставки посилок мандрівниками. При цьому автору необхідно було систематизувати, закріпити та розширити теоретичні знання і практичні навички з інформаційних технологій та застосування їх при розв’язанні задачі

розробки застосунку. Крім того було поставлено за мету зібрати повноцінну команду з дизайнерів та розробників для пришвидшення реалізації продукту.

Поставлена мета передбачає вирішення таких основних задач:

- вибір засобів реалізації завдання;
- аналіз ринку доставки;
- вивчення фреймворку Spring [4-5];
- вивчення та використання в проєкті систему онлайн платежів;
- забезпечення верифікації користувачів шляхом підтвердження номеру мобільного телефону;
- реалізація мобільного застосунку.

Отримані результати: розроблено застосунок (проєкт «Evopost»), який дозволяє мандрівникам доставляти посилки.

У роботі використано такі технології як: Spring Framework [4-5]– універсальний фреймворк для написання бек-енду мовою Java, Hibernate [6-7] – ORM фреймворк для створення відображення між об'єктами та реляційними структурами, об'єктно-реляційну СКБД PostgreSQL [8-9] для збереження інформації про користувачів застосунку, сховище медіа файлів Cloudinary [10-11] для збереження фотографій користувачів, платіжну систему Stripe [12-13] для безпечних онлайн переказів та сервіс відправки СМС-повідомлень Vonage [14-15] для верифікації користувачів. Щоб забезпечити надійну автентифікацію та авторизацію користувачів було використано технологію JWT [16].

Можливі сфери застосування. За «Evopost» може бути застосованим користувачами, що часто подорожують, а також люди, які живуть у різних містах зі своїми рідними та друзями і мають потребу у швидких та частих передачах посилок.

РОЗДІЛ 1.
АНАЛІЗ РИНКУ

Для вдалого запуску проекту, перш за все, потрібно ретельно проаналізувати ринок, виявити головних конкурентів, знайти їх слабкі сторони та головні недоліки. Це допоможе оцінити реальні шанси на успіх а також визначити чого саме не вистачає користувачам вже існуючих сервісів та реалізувати це у власному унікальному продукті.

Серед існуючих гравців ринку доставки можна виокремити наступні компанії:

1. PiggyBee [17].
2. BuddyExpress [18].
3. HeyParcel [19].
4. Zaldee [20].
5. Travoney [21].
6. Koorier [22].
7. AirWillBill [23].
8. Nimber [24].
9. TravelPost [25].
10. Grabr [26].

Кожен з вище перелічених сервісів не є досконалим. Далі ми детально розберемо їх головні недоліки.

PiggyBee доступний лише у веб-версії, що є критичним на сьогоднішній день, адже люди все більше користуються мобільними застосунками і все рідше використовують браузерні версії продуктів.

BuddyExpress можна завантажити лише на iOS пристрої, що урізає кількість потенційних користувачів більш ніж удвічі.

HeyParcel підтримується тільки на платформі Android, що також значно обмежує цільовий ринок. У сервісі відсутні безпечні платежі, що може знижувати рейтинг довіри до продукту.

Zaldee та AirWillBill доступні тільки в англійській версії, а також як і HeyParcel не передбачають безпечне проведення платежів.

Попри новітній дизайн застосунок Travoney можуть встановити лише користувачі Android-смартфонів.

На відміну від Travoney Koorier має досить застарілий дизайн, користування застосунком є занадто складним та інтуїтивно незрозумілим. Відсутність вбудованої платіжної системи і веб-сайту автоматично знижують рейтинг даного сервісу.

Застосунок Nimber розроблено виключно під локальний ринок, тому масштабування цього сервісу обмежено.

TravelPost є досить гарним конкурентом, але користувачів можуть очікувати складнощі під час користування застосунком, велика кількість багів а також висока комісія за проведення платежів.

Сервіс Grabr розрахований лише на доставку товарів із магазинів Сполучених Штатів Америки.

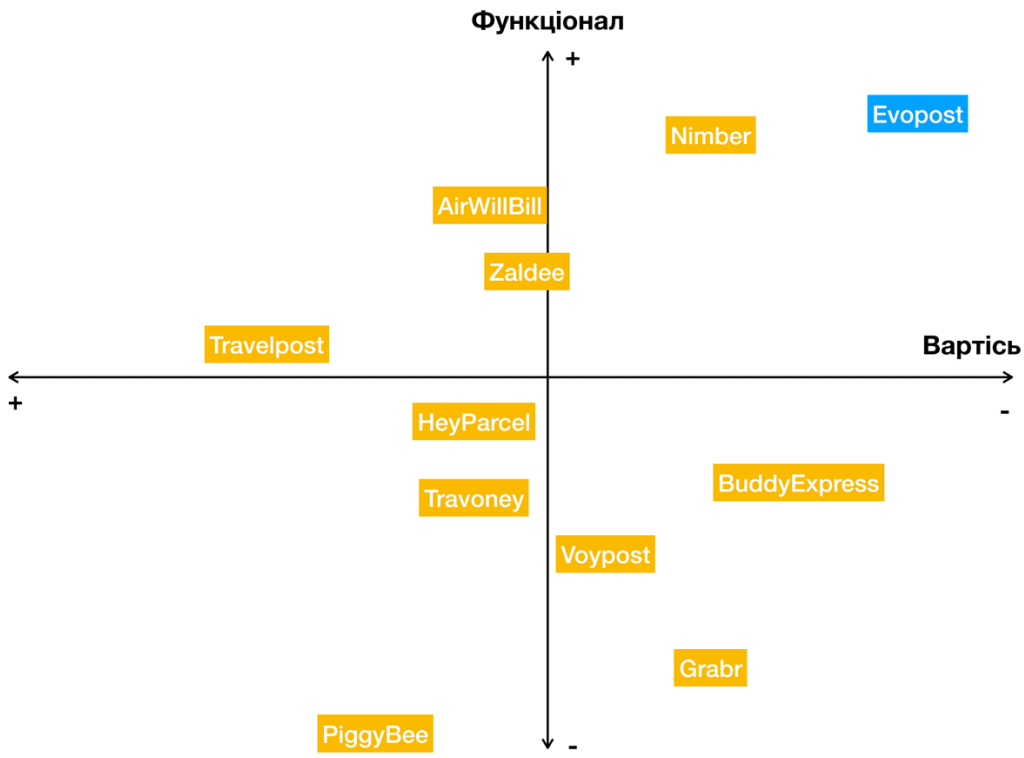


Рисунок 1. Конкуренти на координатній площині

Високий попит на міжнародну доставку посилок та помірна конкуренція на ринку є сприятливими умовами для розробки програмного продукту “Evopost”. Він матиме наступний функціонал, що буде виокремлювати його серед конкурентів:

- Безпечна система оплати.
- Страхування посилок.
- Відстеження посилок у реальному часі.
- Модель довіри на основі відгуків та рейтингів.
- Опція «Купи для мене».
- Вбудований планер зустрічей.
- Система штучного інтелекту для виявлення спаму.
- Вбудований сервіс обміну повідомленнями.
- Запланована інтеграція з авіакомпаніями та платформами онлайн оголошень.

Комісія сервісу за проведення платежів буде становити 10%, що істотно нижче, ніж у конкурентів. У той же час цього буде достатньо для покриття витрат на розробку та маркетинг, а також подальших інвестицій спрямованих на розвиток продукту.

РОЗДІЛ 2.
ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

Для реалізації серверної частини «Evopost» через свою високу популярність і легкість використання були розглянуті та обрані наступні технології: Spring Framework [4-5], Hibernate [6-7], об'єктно-реляційна система керування базами даних PostgreSQL, сховище медіа файлів Cloudinary, платіжну систему Stripe, сервіс відправки СМС-повідомлень Vonage та технологію JWT. У цьому розділи наведено огляд використаних технологій, описано їх деякі переваги та недоліки.

2.1. Фреймворк Spring

Spring Framework [4-5] – універсальний фреймворк з відкритим вихідним кодом для Java-платформи.

Фреймворк був вперше випущений під ліцензією Apache 2.0 license в червні 2003 року. Перший стабільний реліз 1.0 був випущений в березні 2004. Spring 2.0 був випущений в жовтні 2006, Spring 2.5 - в листопаді 2007, Spring 3.0 в грудні 2009, і Spring 3.1 в грудні 2011. Поточна версія Spring – 5.

Незважаючи на те, що Spring не забезпечував якусь конкретну модель програмування, він став широко поширеним в Java-співтоваристві головним чином як альтернатива і заміна моделі Enterprise JavaBeans. Spring надає велику свободу Java-розробникам в проєктуванні; крім того, він надає добре документовані і легкі у використанні засоби вирішення проблем, що виникають при створенні застосунків корпоративного масштабу.

Spring [5] відомий як джерело розширень (features), потрібних для ефективної розробки складних бізнес-застосунків поза великовагових програмних моделей, які історично були домінуючими в промисловості. Ще одне його перевага в тому, що він ввів раніше невикористовувані

функціональні можливості в сьогоднішні панівні методи розробки, навіть поза платформи Java.

Цей фреймворк пропонує послідовну модель і робить її придатною до більшості типів застосунків, які вже створені на основі платформи Java.

2.2. Фреймворк Hibernate

Hibernate [6-7] — засіб відображення між об'єктами та реляційними структурами (object-relational mapping, ORM) для платформи Java. Hibernate є вільним програмним забезпеченням, яке поширюється на умовах GNU Lesser General Public License. Hibernate надає легкий для використання каркас (фреймворк) для відображення між об'єктно-орієнтованою моделлю даних і традиційною реляційною базою даних.

Метою Hibernate є звільнення розробника від значних типових завдань із програмування взаємодії з базою даних. Розробник може використовувати Hibernate як при розробці з нуля, так і для вже існуючої бази даних.

Hibernate піклується про зв'язок класів з таблицями бази даних (і типів даних мови програмування із типами даних SQL), і надає засоби автоматичної побудови SQL запитів й зчитування/запису даних, і може значно зменшити час розробки, який зазвичай витрачається на ручне написання типового SQL і JDBC коду. Hibernate генерує SQL виклики і звільняє розробника від ручної обробки результуючого набору даних, конвертації об'єктів і забезпечення сумісності із різними базами даних.

Hibernate [7] забезпечує прозору підтримку збереження даних, тобто їхньої персистентності (англ. persistence) для «POJO»-об'єктів, себто для звичайних Java-об'єктів; єдина сувора вимога до класу, що зберігається — конструктор за замовчанням.

Mapping (зіставлення, буквально — картування) Java класів з таблицями бази даних здійснюється за допомогою конфігураційних XML файлів або Java анотацій. При використанні файлу XML, Hibernate може генерувати скелет

вихідного коду для класів тривалого зберігання (persistent). У цьому немає необхідності, якщо використовується анотація. Hibernate може використовувати файл XML або анотації для підтримки схеми бази даних.

Забезпечуються можливості з організації відношення між класами «один-до-багатьох» і «багато-до-багатьох». На додаток до управління зв'язками між об'єктами, Hibernate також може керувати рефлексивними асоціаціями, де об'єкт має зв'язок «один-до-багатьох» з іншими примірниками свого власного типу даних.

Hibernate підтримує відображення користувацьких типів значень. Це робить можливим такі сценарії:

- перевизначення типу за замовчуванням SQL, який Hibernate вибирає при відображенні стовпчика властивості;
- картування перерахованого типу Java до колонок БД, так ніби вони є звичайними властивостями;
- картування однієї властивості в декілька колонок.

Hibernate забезпечує збереження POJO (Plain Old Java Objects — простих старих об'єктів Java). Єдина сувора вимога для персистентного класу — конструктор без аргументів, не обов'язково публічний. Для правильної поведінки деяких програм також потрібна особлива увага до методів equals() і hashCode().

Колекції об'єктів даних, як правило, зберігаються у вигляді колекцій Java-об'єктів, таких як набір (Set) і список (List). Підтримуються узагальнені класи (Generics), введені в Java 5. Hibernate може бути налаштований на «ледачі» (відкладені) завантаження колекцій. Відкладені завантаження є варіантом за замовчуванням, починаючи з Hibernate 3.

Зв'язані об'єкти можуть бути налаштовані на каскадні операції. Наприклад, батьківський клас, User може бути налаштований на каскадне збереження і/або видалення свого нащадка UserRating. Це може скоротити час розробки і забезпечити цілісність. Функція перевірки зміни даних (dirty checking) дозволяє уникнути непотрібного запису дій в базу даних, виконуючи

SQL оновлення тільки при зміні полів персистентних об'єктів.

2.3. СКБД PostgreSQL

PostgreSQL [8-9] — об'єктно-реляційна система керування базами даних. Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно з іншими проєктами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.

Сервер PostgreSQL написаний мовою C. Розповсюджується у вигляді набору текстових файлів із source кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп'ютері і скопіювати в деякий каталог. Весь процес детально описаний в документації до PostgreSQL [9].

У PostgreSQL є підтримка індексів наступних типів: B-дерево, хеш, R-дерево, GiST, GIN. При необхідності можна створити нові типи індексів.

PostgreSQL [9] підтримує одночасну модифікацію БД декількома користувачами за допомогою механізму Multiversion Concurrency Control.

PostgreSQL підтримує великий набір вбудованих типів даних:

- числові типи;
- цілі;
- з фіксованою крапкою;
- з нефіксованою крапкою;
- грошовий тип;
- символьні типи довільної довжини;
- двійкові типи;
- типи «дата/час»;

- булевий тип;
- перерахування;
- геометричні примітиви;
- мережеві типи;
- IP і IPv6-адреси;
- CIDR-формат;
- MAC-адреса;
- UUID-ідентифікатор;
- XML-дані;
- JSON-дані;
- масиви;
- псевдотипи.

Крім того, користувач може самостійно створювати нові необхідні йому типи та програмувати для них механізми індексування за допомогою GiST.

PostgreSQL може бути розширено користувачем для власних потреб практично в будь-якому аспекті.

Таблиці можуть успадковувати характеристики та набори полів від інших таблиць (батьківських). При цьому дані, які додаються до породженої таблиці, автоматично будуть брати участь (якщо це не вказано окремо) в запитах до батьківської таблиці. Цей функціонал в поточний час не є повністю завершеним. Однак він достатній для практичного використання.

Тригери визначаються як функції, що ініціюються DML-операціями. Наприклад, операція INSERT може запускати тригер, що перевіряє доданий запис на відповідність певним умовам.

2.4. Сховище медіа файлів Cloudinary

Cloudinary [10-11] - SaaS компанія, яка надає хмарні послуги управління зображеннями та відео. Це дає можливість користувачам завантажувати, зберігати і керувати зображеннями та відео на веб-сайтах і у застосунках.

Cloudinary використовується у веб- та мобільних застосунках понад 6 000 компаніями, включаючи Condé Nast, Dropbox, Forbes, Outbrain, Taboola та Answers.com. Журнал Inc [27] назвав Cloudinary "золотим стандартом" управління зображеннями в Інтернеті.

Cloudinary надає бібліотеку Java з відкритим кодом для подальшого спрощення інтеграції. Ця бібліотека має наступний функціонал:

- створення URL-адреси зображення;
- API: завантаження файлів, адміністрування та багато іншого;
- бібліотека тегів JSP для полегшення перетворення, завантаження та зберігання у медіа файлів у Java EE веб-застосунках;
- завантаження файлів на сервері + пряме непідписане завантаження файлу з браузера за допомогою плагіна jQuery [28].

Бібліотека розроблена для Java 6 / JSP 2.0 і буде також сумісна з більш високими версіями.

2.5. Платіжна система Stripe

Програмне забезпечення Stripe [12-13] дозволяє приватним особам та підприємствам здійснювати та отримувати платежі через Інтернет. Stripe забезпечує технічну, банківську та анти-шахрайську інфраструктуру, необхідну для роботи в онлайн-платіжних системах.

Stripe надає API, які розробники можуть використовувати для інтеграції обробки платежів у своїх веб-сайтах та мобільних застосунках. У квітні 2018 року компанія випустила інструменти для блокування шахрайських транзакцій [13].

У 2018 році компанія розширила свої послуги, включивши платіжний продукт для онлайн-бізнесу. Послуга функціонує в рамках платформи Stripe, що дозволяє підприємствам керувати періодичними доходами та виставленням рахунків за підпискою. Stripe дозволяє зберігати клієнтів, додавати довільні метадані при створенні платежу, щоб було простіше

орієнтуватися в проведених платежах, задавати параметр “description” при платежі для його більш інформативного опису, і багато іншого.

У липні 2018 року у Stripe додали підтримку кредитних карток Mastercard та Visa, а згодом Apple та Google Pay.

2.6. Сервіс відправки СМС-повідомлень Vonage

Vonage [14-15] пропонує хмарні комунікації та плани дзвінків для приватних клієнтів і підприємств, включаючи малий і середній бізнес, компанії середнього ринку та великі підприємства.

Vonage взяв за основу глобальну телекомунікаційну мережу і перетворив її у хмарну платформу зв'язку з можливостями здійснення аудіо та відео дзвінків у веб-і мобільних застосунках, надсилання та отримання SMS, MMS та IP-повідомлень, інтеграції двофакторної аутентифікації для посилення та навіть заміни традиційного входу в систему тощо.

2.7. Стандарт JWT

JSON Web Token (JWT) [16] – це відкритий стандарт (RFC 7519) для створення токенів доступу, реалізований на JSON форматі. Як правило, використовується для передачі даних авторизації в клієнт-серверних застосунках. Токени створюються сервером, підписуються секретним ключем і передаються клієнту, який в подальшому використовує даний токен для підтвердження своєї особи.

Токен JWT складається з трьох частин: заголовок (header), корисне навантаження (payload) і підпис або дані шифрування. Перші два елементи - це JSON об'єкти певної структури. Третій елемент обчислюється на підставі перших і залежить від обраного алгоритму (у випадку використання не підписаного JWT може бути опущений). Токени можуть бути перекодовані в компактне представлення (JWS / JWE Compact Serialization): до заголовку і

корисного навантаження застосовується алгоритм кодування Base64-URL, після чого додається підпис і всі три елементи розділяються крапками («.»).

Як правило, при використанні JSON токенов в клієнт-серверних застосунках реалізована наступна схема:

1. Клієнт проходить авторизацію у застосунку (наприклад, з використанням логіна і пароля).
2. У разі успішної авторизації сервер відправляє клієнту access- і refresh-токени.
3. При подальшому зверненні до сервера, клієнт використовує access-токен. Сервер перевіряє токен на валідність і надає клієнту доступ до ресурсів.
4. У разі, якщо access-токен стає дійсним, клієнт відправляє refresh-токен, у відповідь на який сервер надає два оновлених токени.
5. У разі, якщо refresh-токен стає недійсним, клієнт знову повинен пройти процес авторизації.

При використанні кукі сервер повинен зберігати інформацію про видані сесіях, в той час як використання JWT не вимагає зберігання додаткових даних про видані токенах: все, що повинен зробити сервер – це перевірити підпис.

Сервер може не займатися створенням токенів, а надати таку можливість зовнішнім сервісам.

У JSON токенах можна зберігати додаткову корисну інформацію про користувачів. Як наслідок - більш висока продуктивність. У випадку з кукі іноді необхідно здійснювати запити для отримання додаткової інформації. При використанні JWT ця інформація може бути передана в самому токені.

РОЗДІЛ 3.
ПРИЗНАЧЕННЯ ТА ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ЗАСТОСУНКУ

3.1. Призначення застосунку

Програмний продукт «Evopost» може застосовуватися користувачами, у яких є потреба доставити посилку недорого і у стислі терміни, або мандрівниками, що прагнуть зекономити на подорожі перевозячи посилки замовників.

Застосунок призначений для використання широкою цільовою аудиторією, починаючи зі студентів, що навчаються закордоном, закінчуючи робітниками, які прямують у відрядження. Детальну “Target Customers” та інші діаграми можна знайти у додатках А-В.

Кожен зареєстрований користувач має можливість опублікувати маршрут своєї подорожі і запропонувати доставку іншим користувачам або ж оприлюднити свою посилку і замовити її перевезення. Функція «Купи для мене» дозволяє розмістити публікацію з посиланням на продукт, який потрібно придбати і доставити.

Оплата перевезення повинна здійснюватися через застосунок «Evopost». Кошти не переводяться миттєво, а будуть утримані сервісом до тих пір, поки товар не буде доставлено до отримувача, який повинен підтвердити факт надходження посилки. За кожну оброблену транзакцію «Evopost» отримує 7% від винагороди за перевезення.

Комунікація між користувачами відбувається у вбудованому сервісі обміну повідомленнями. Користувачі можуть обмінюватися повідомленнями тільки після того, як запит на доставку було прийнято перевізником або відправником, залежно від сторони надходження інтересу.

Після оплати та успішної доставки можна залишити відгук та оцінити свого перевізника/відправника. Система рейтингів допоможе підняти рівень довіри до сервісу «Evopost» та його користувачів.

3.2 Процес розробки застосунку

Весь код застосунку зберігається у приватному GitHub [29] репозиторії, що підключена до Continuous Integration сервісу Travis CI [30], який у свою чергу зв’язаний з платформою Heroku [31].

Бек-енд розробка відбується у 3 етапи (див. рис. 1). Спочатку новий функціонал або правки коду завантажуються на GitHub. Потім проєкт повинен бути зібраний та протестований на Travis CI. Завдяки цьому кроку знижується ризик падіння сервісу через помилки у коді. Після успішного проходження усіх тестів сервіс автоматично розгортається на Heroku. Сервіс стає публічно доступним і клієнти (Android та iOS) можуть комунікувати з ним через REST протокол.

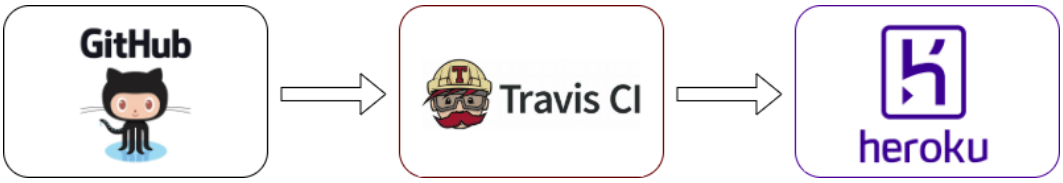


Рисунок 2. Процес розробки

Для тестування API (див. рис. 2) серверу використовується бібліотека Swagger-UI [32], що дає можливість зручного перегляду у браузері доступних ендпоінтів та моделей, що приймають та відправляють контролери.

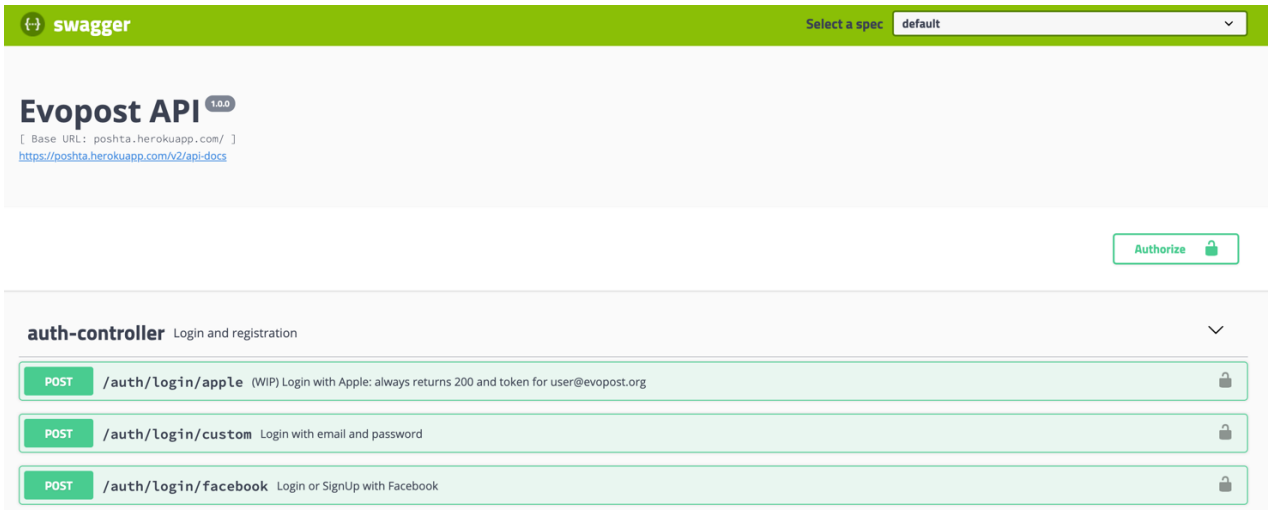


Рисунок 3. Swagger-UI

3.3 Архітектура застосунку

Клієнт застосунку реалізовано на мобільних платформах iOS та Android. Серверна частина розроблена мовою Java з використанням фреймворку Spring Boot. Для написання інтеграційних тестів було обрано мову Python. У ролі сховища даних виступає об'єктно-реляційна СКБД PostgreSQL.

Щоб забезпечити проведення платежів у застосунок було інтегровано платіжну систему Stripe. Для підтвердження телефонного номера користувача було обрано сервіс відправки SMS повідомлень Vonage. Хмарне сховище медіа файлів Cloudinary використовується для збереження фотографій користувачів, а також публікацій їх оголошень. Для обміну повідомленнями між користувачами застосунку було застосовано протокол Web-Sockets [33]. Спрощену модель архітектури реалізованої програми наведено на рис. 3.

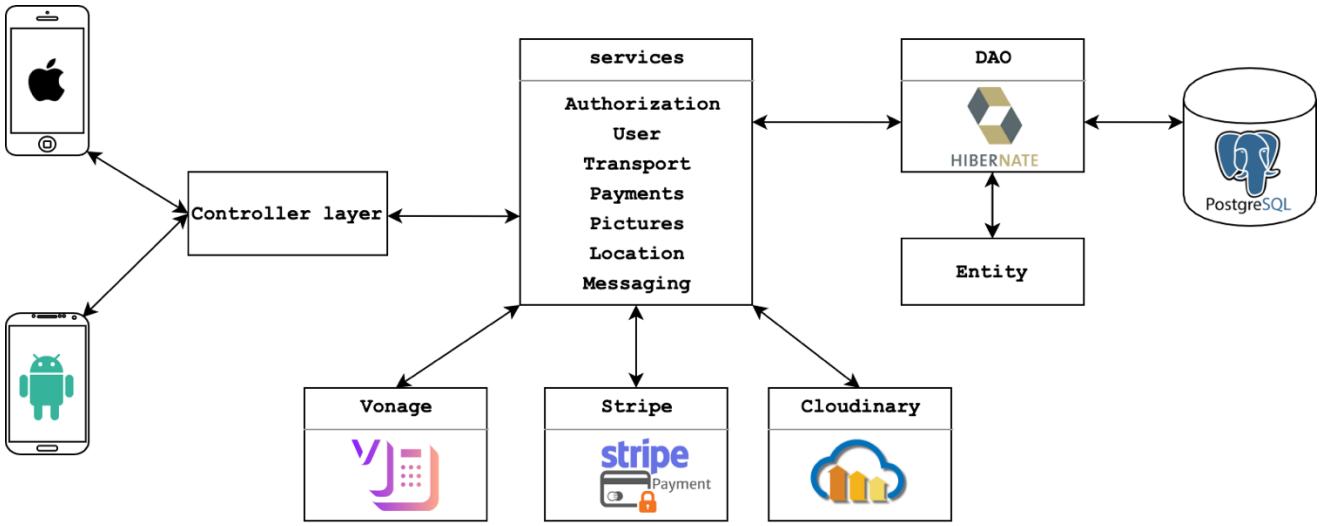


Рисунок 4. Архітектура застосунку

3.4 Особливості реалізації застосунку

Робота над застосунком відбувалася у команді з дев’яти участників. Зокрема, у проєкті приймали участь два дизайнери, два розробники серверної частини додатку, два мобільні розробники, юрист-консультант, проєктний менеджер, та автор роботи у ролі розробника і бізнес девелопера.

Серед усіх членів команди були чітко розділені зони відповідальності та їхні забов’язання. За зручність використання застосунку відповідав UX-дизайнер. В свою чергу, за гарний інтерфейс додатку відповідав UI-дизайнер. Мобільні розробники поділили між собою роботу над iOS та Android версіями аплікації. За серверну частину і архітектуру та деплоймент відповідали 2 Java розробники, а також автор роботи на початку проекту. При наближенні до релізу проекту - автор роботи сконцентрувався на бізнес та маркетинговій частині, відповідав за залучення інвестицій і вів перемовини з інвесторами. Для того, щоб розібратися у деталях перевезень товарів закордон було залучено юриста-консультанта, який також взяв на себе створення правил користування додатком.

Проєктний менеджер відповідав за своєчасне виконання завдань кожним членом команди, проведення мітингів, планування задач і синхронізацію між серверною та мобільною командами розробки. В його обов’язки також входило розуміння бізнес вимог та донесення їх до розробників і дизайнерів, а також контроль якості виконання.

РОЗДІЛ 4.
ІНСТРУКЦІЯ КОРИСТУВАЧА

Фронт-енд проєкту було реалізовано у вигляді мобільної версії для Android та iOS. На прикладі другої реалізації клієнтського застосунку проілюструємо інструкцію використання програми. Android-версія має майже ідентичний інтерфейс і буде інтуїтивно зрозумілою після уважного прочитання даної інструкції.

Після запуску програми перед користувачем з'являється вікно входу у застосунок. Вхід може бути здійснено як за допомогою електронної пошти, так і за через соціальні мережі (див. рис. 5).

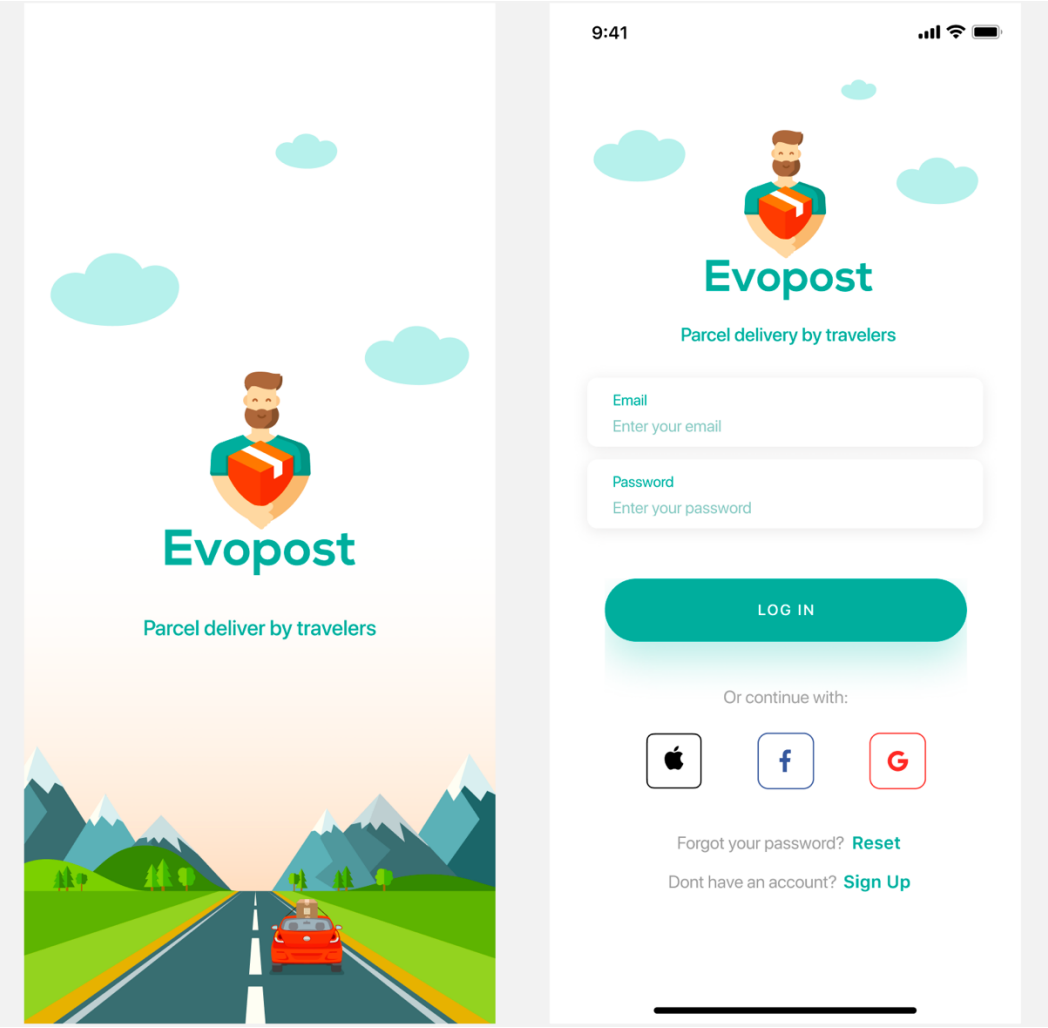


Рисунок 5. Розділ застосунку «Вхід»

Для реєстрації за допомогою електронної пошти потрібно ввести своє ім’я, по-батькові, а також придумати пароль не менше, ніж на 8 символів (див. рис. 6).

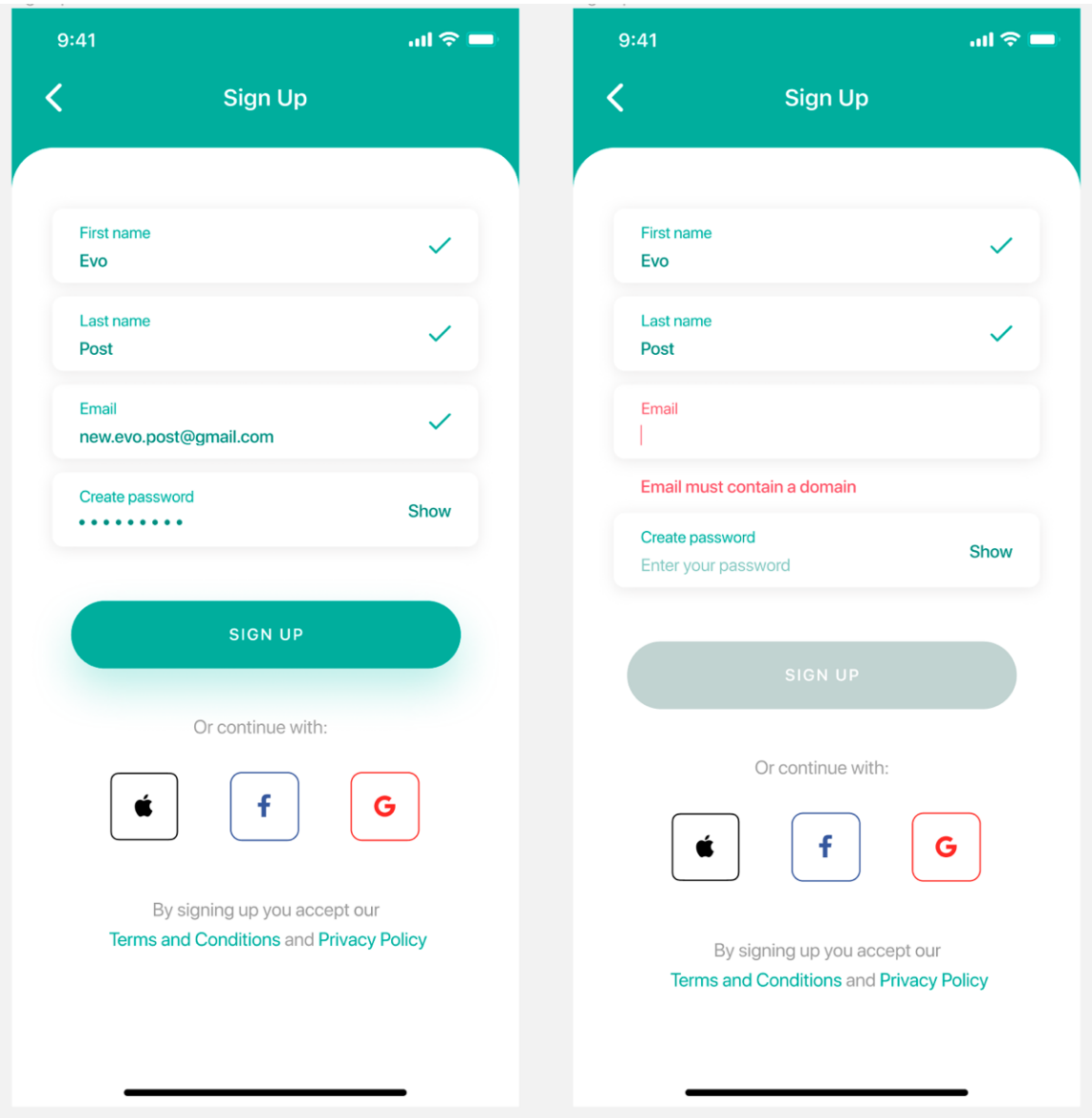


Рисунок 6. Розділ застосунку «Реєстрація»

У випадку, якщо користувач попередньо зареєструвався за допомогою електронної пошти і забув свій пароль – він може скинути його ввівши адресу поштової скриньки, на яку буде надіслано новий випадково згенерований пароль (див. рис. 7).

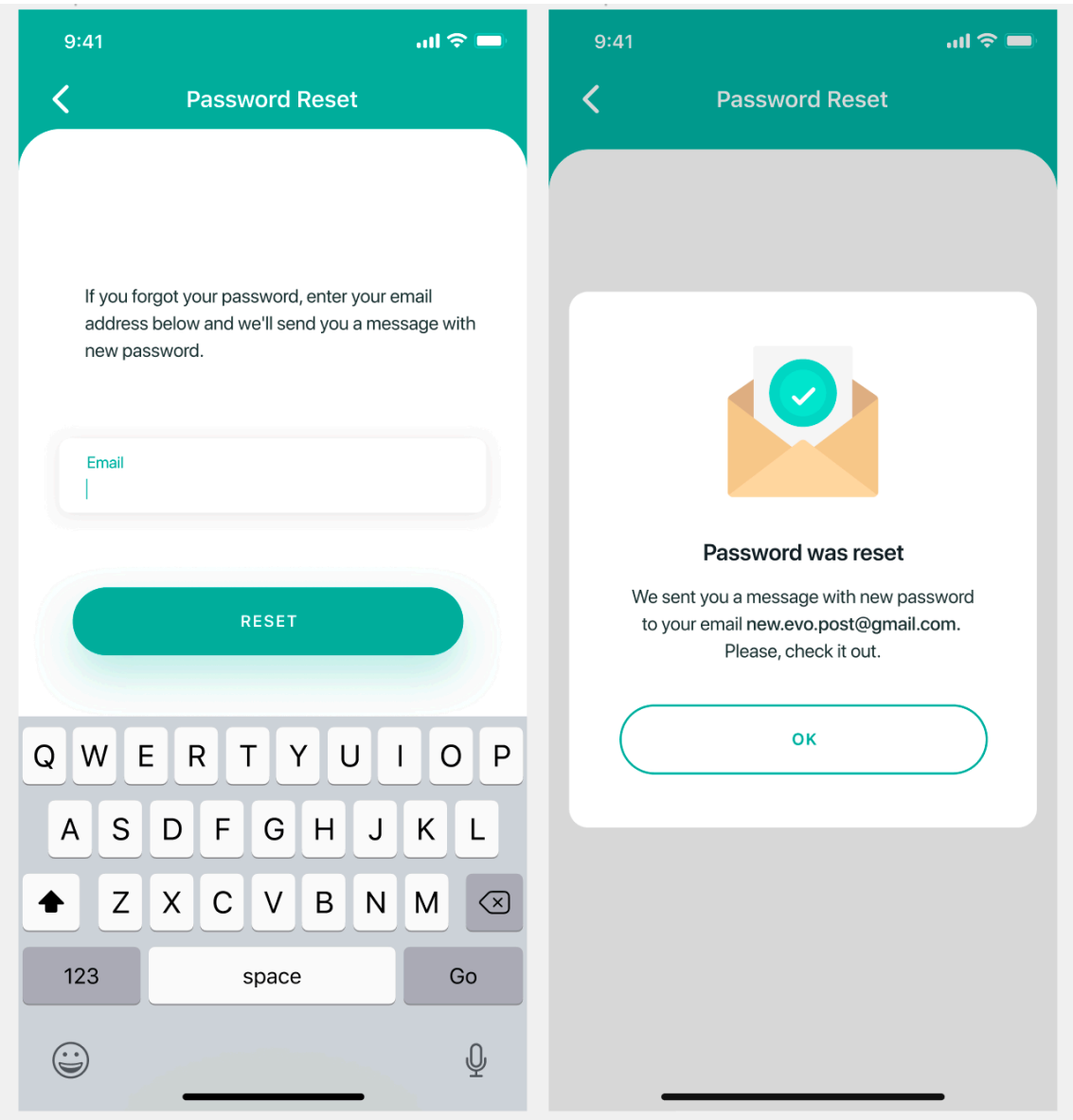


Рисунок 7. Розділ застосунку «Скидання паролю»

Після успішного входу у застосунок перед користувачем з’являється вікно застосунку, яке має декілька підрозділів: «Головна», «Мої пости», «Зробити пост», «Повідомлення» та «Профіль». В останньому (див. рис. 8) користувач може змінювати інформацію про себе, додавати платіжні методи, змінювати пароль, вимикати нотифікації, звертатися у підтримку, переглянути умови користування та політику конфіденційності, а також вийти з системи.

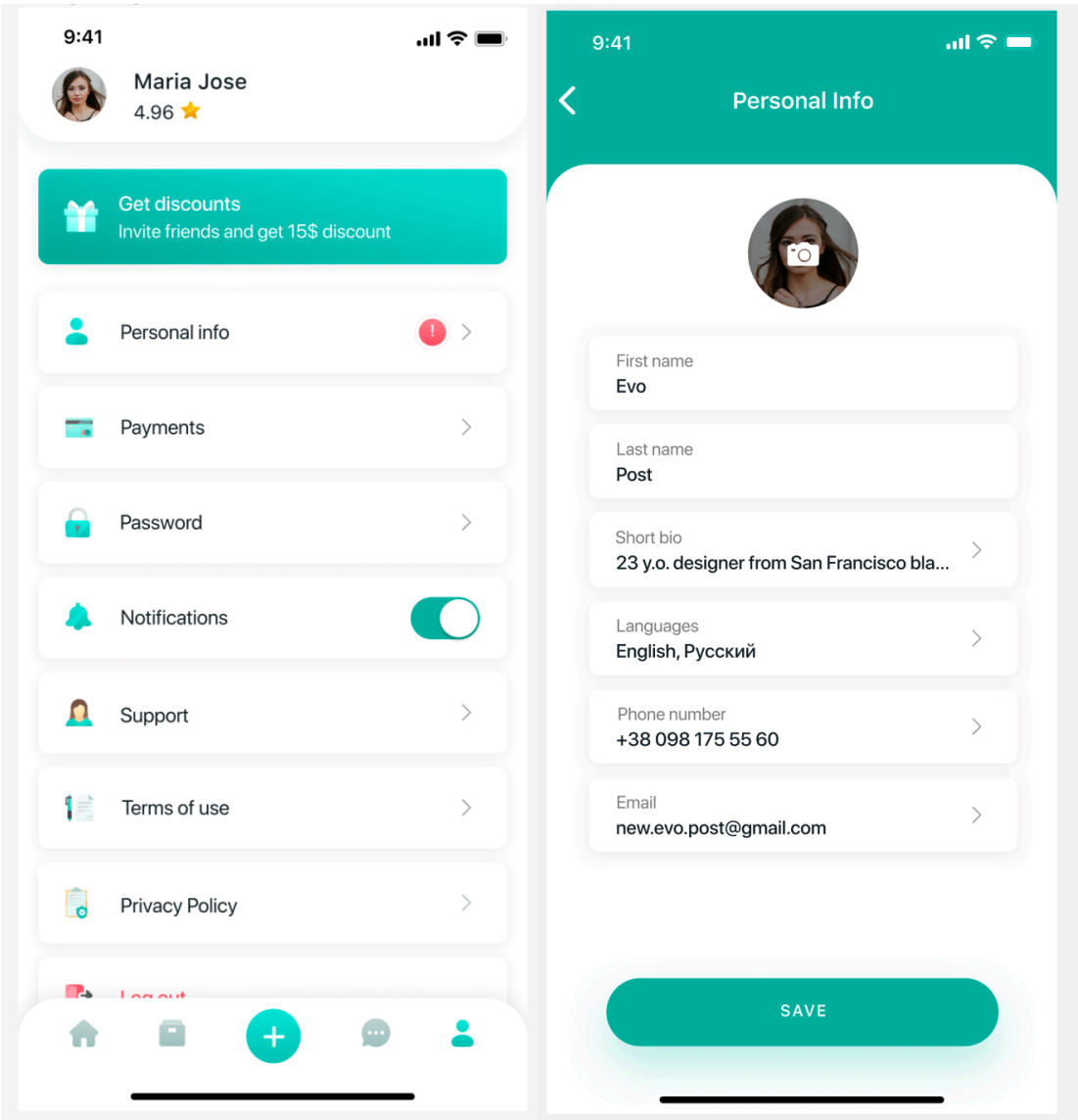


Рисунок 8. Розділ застосунку «Профіль»

Натиснувши на символ «плюс» посередині нижньої панелі користувач переходить у розділ «Зробити пост», де можна створити нову посилку. Під час її створення можна обрати опцію «Купи для мене», де буде вказане посилання на продукт, ціна та кількість товару, який потрібно доставити (див. рис. 9).

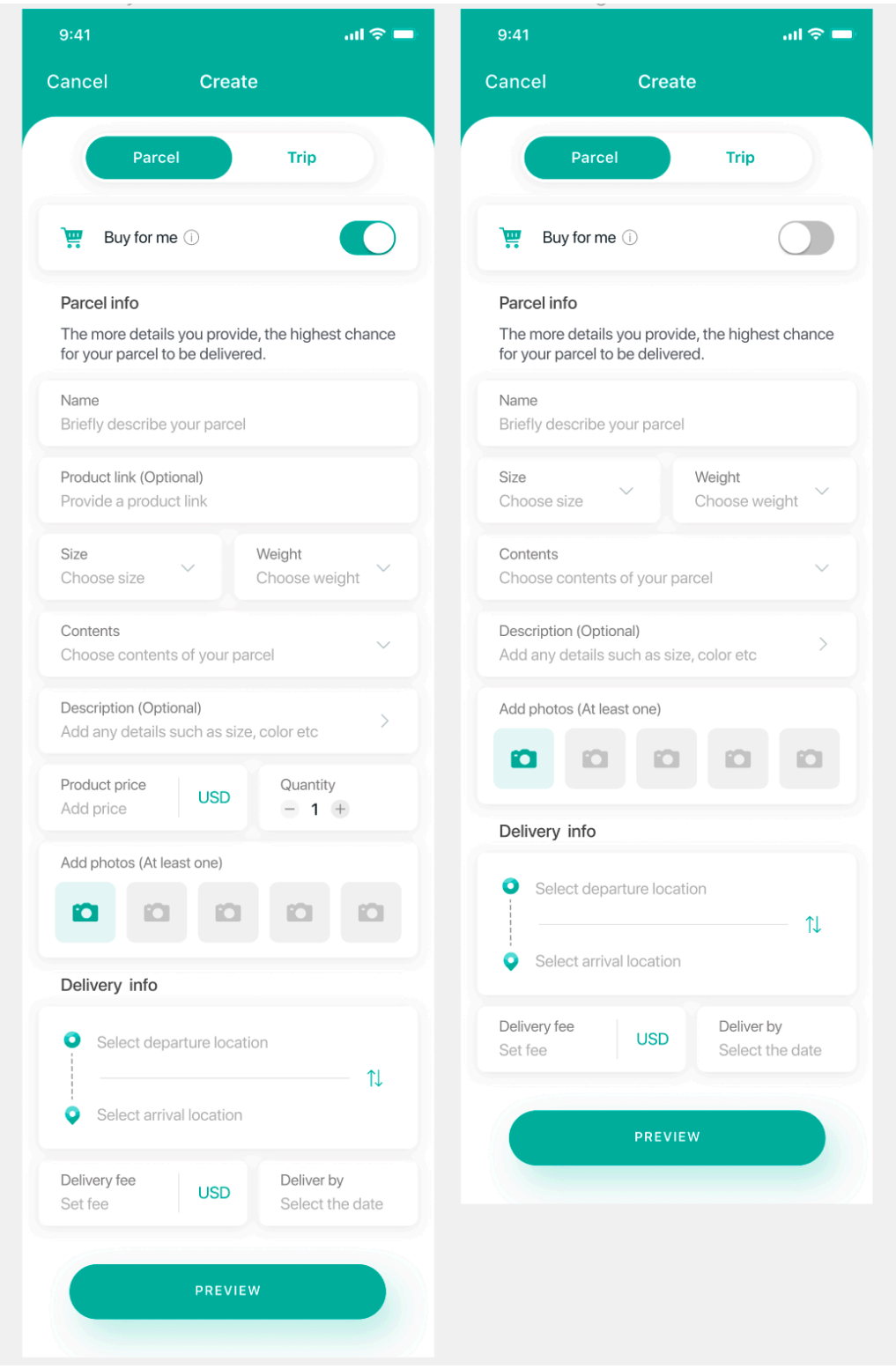


Рисунок 9. Розділ застосунку «Створити посилку»

При створенні подорожі необхідно вказати свій маршрут, дати, короткий опис, максимальний об’єм та вагу, яку ви згодні перевести, а також зазначити які предмети не є бажаними для перевезення (див. рис. 10).

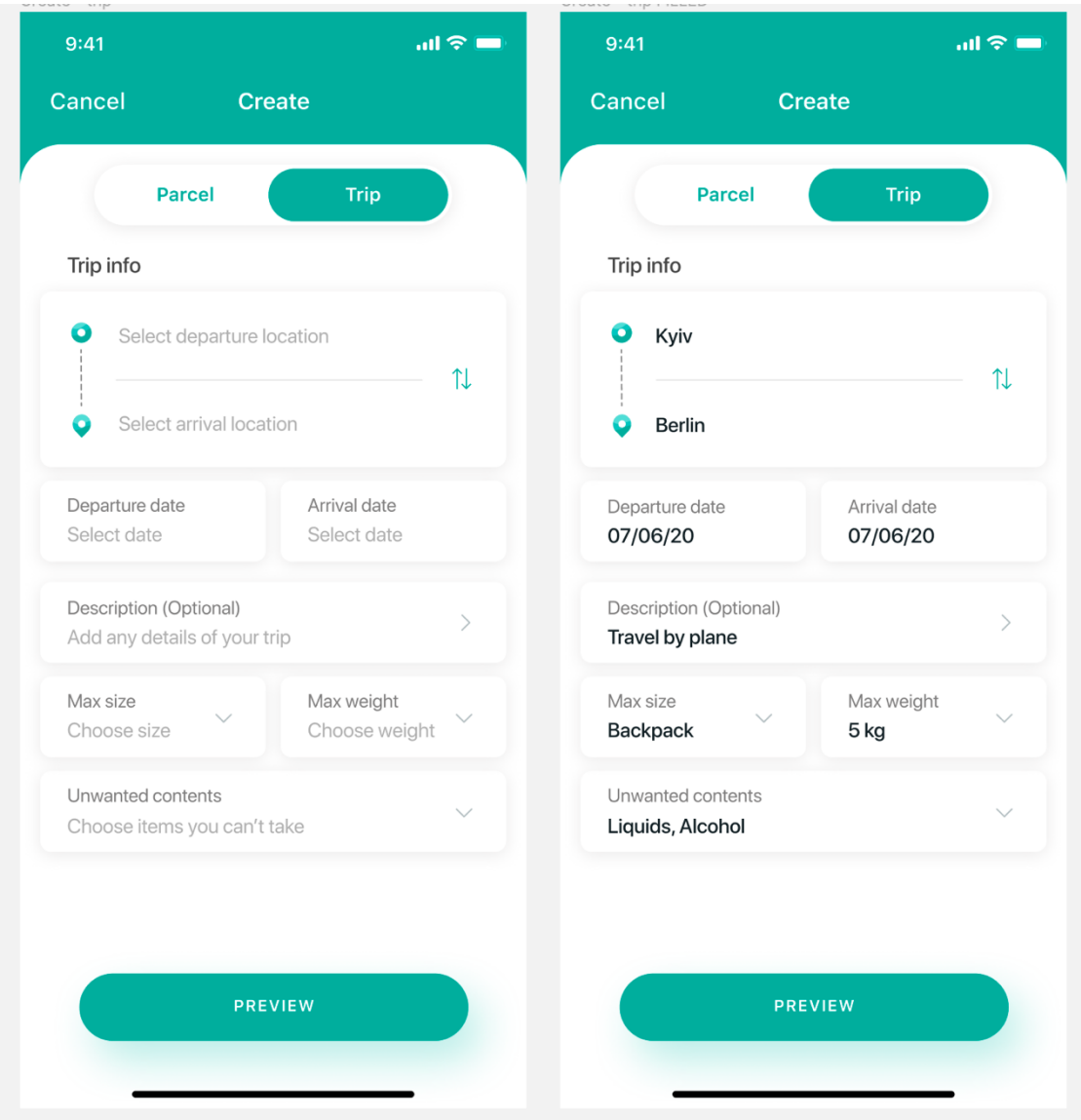


Рисунок 10. Розділ застосунку «Створити подорож»

Перед тим як опублікувати свою подорож чи посилку, користувач має можливість попереднього перегляду (див. рис. 11).

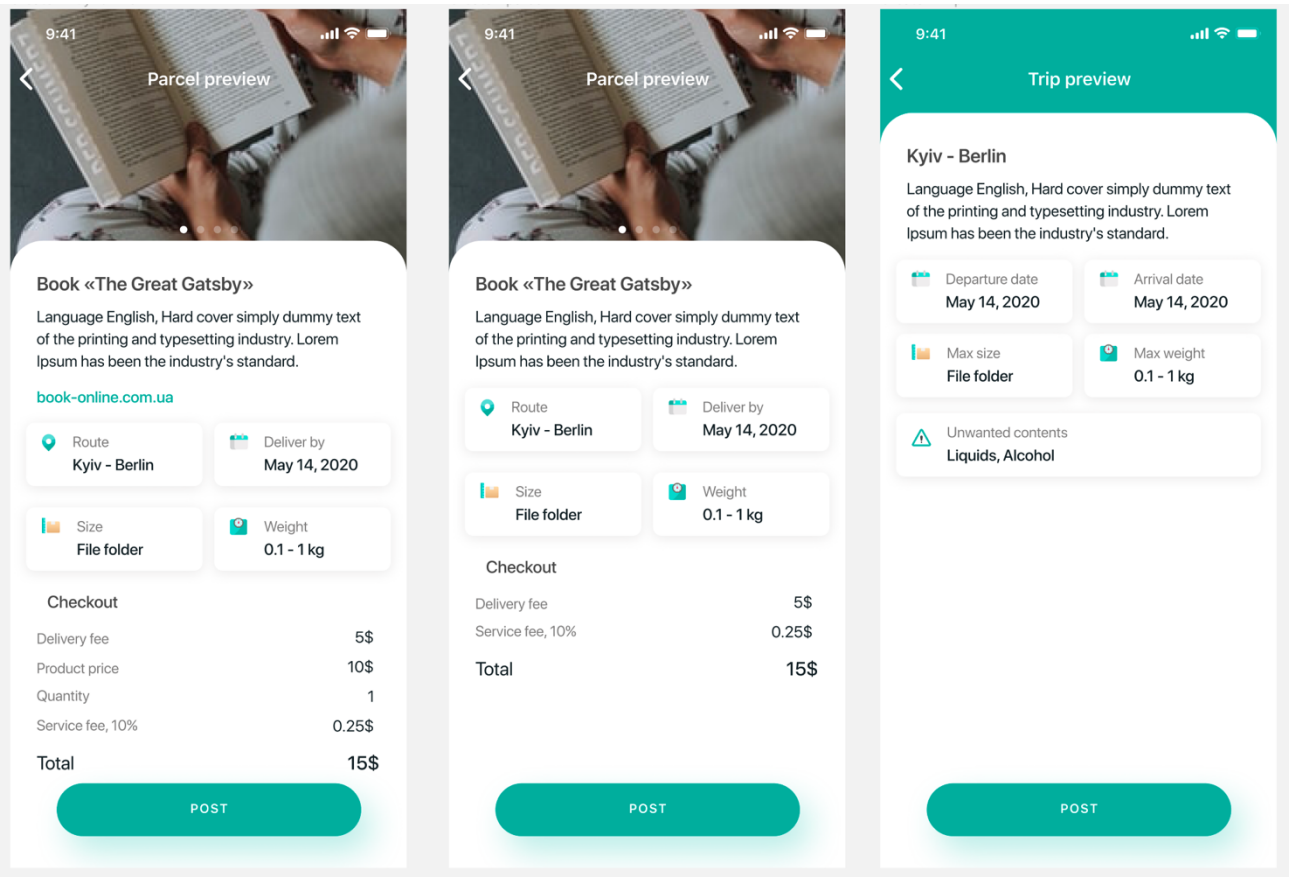


Рисунок 11. Розділ застосунку «Попередній перегляд посилок та подорожей»

Для того, щоб переглядати, редагувати або видаляти свої публікації, потрібно перейти у розділ «Мої пости» (див. рис. 12).

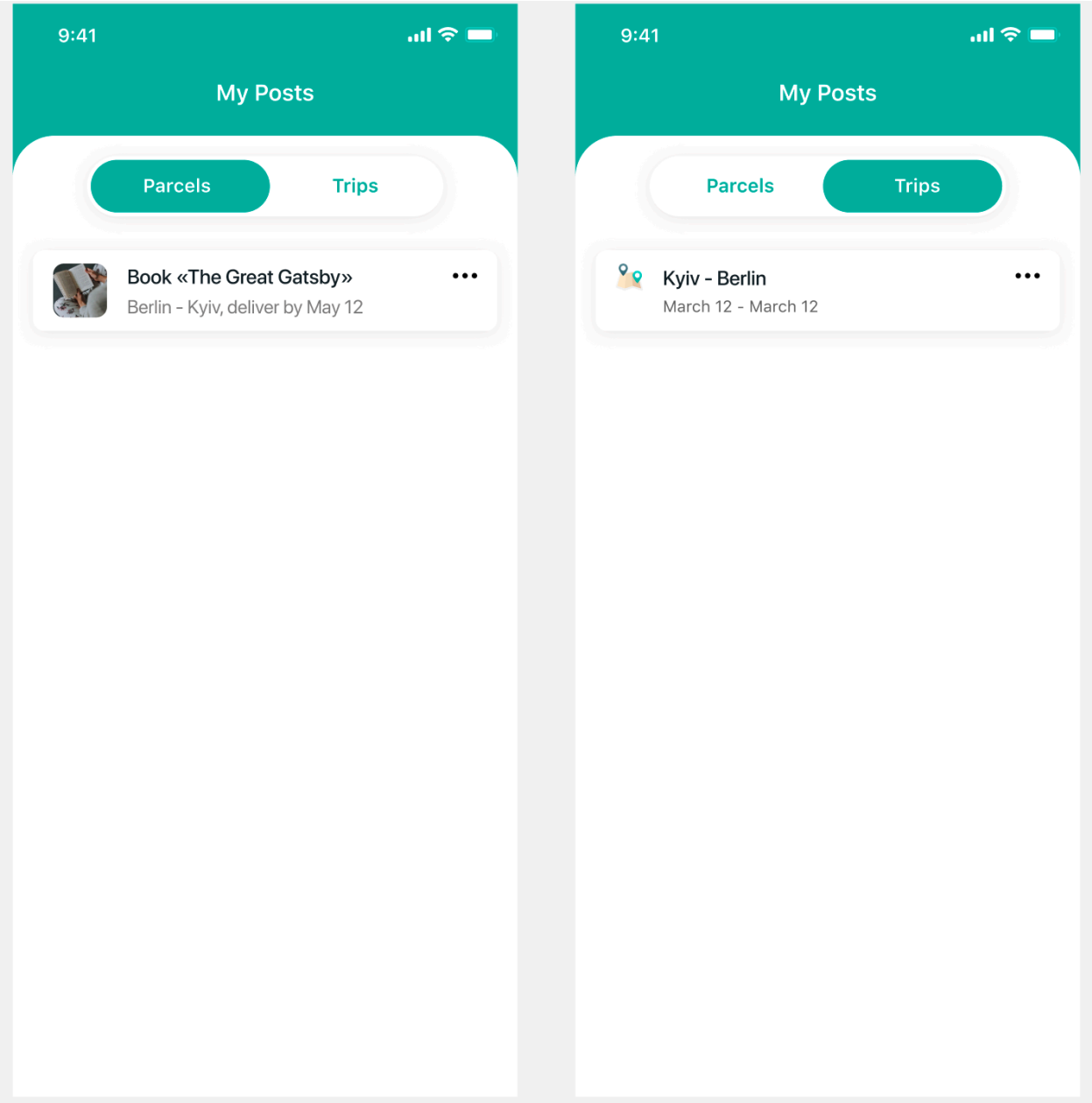


Рисунок 12. Розділ застосунку «Мої пости»

Щоб здійснити пошук посилок або подорожей необхідно перейти у розділ «Головна». Користувач може заповнити критерії пошуку власноруч, або шукати за своїми публікаціями (див. рис. 13).

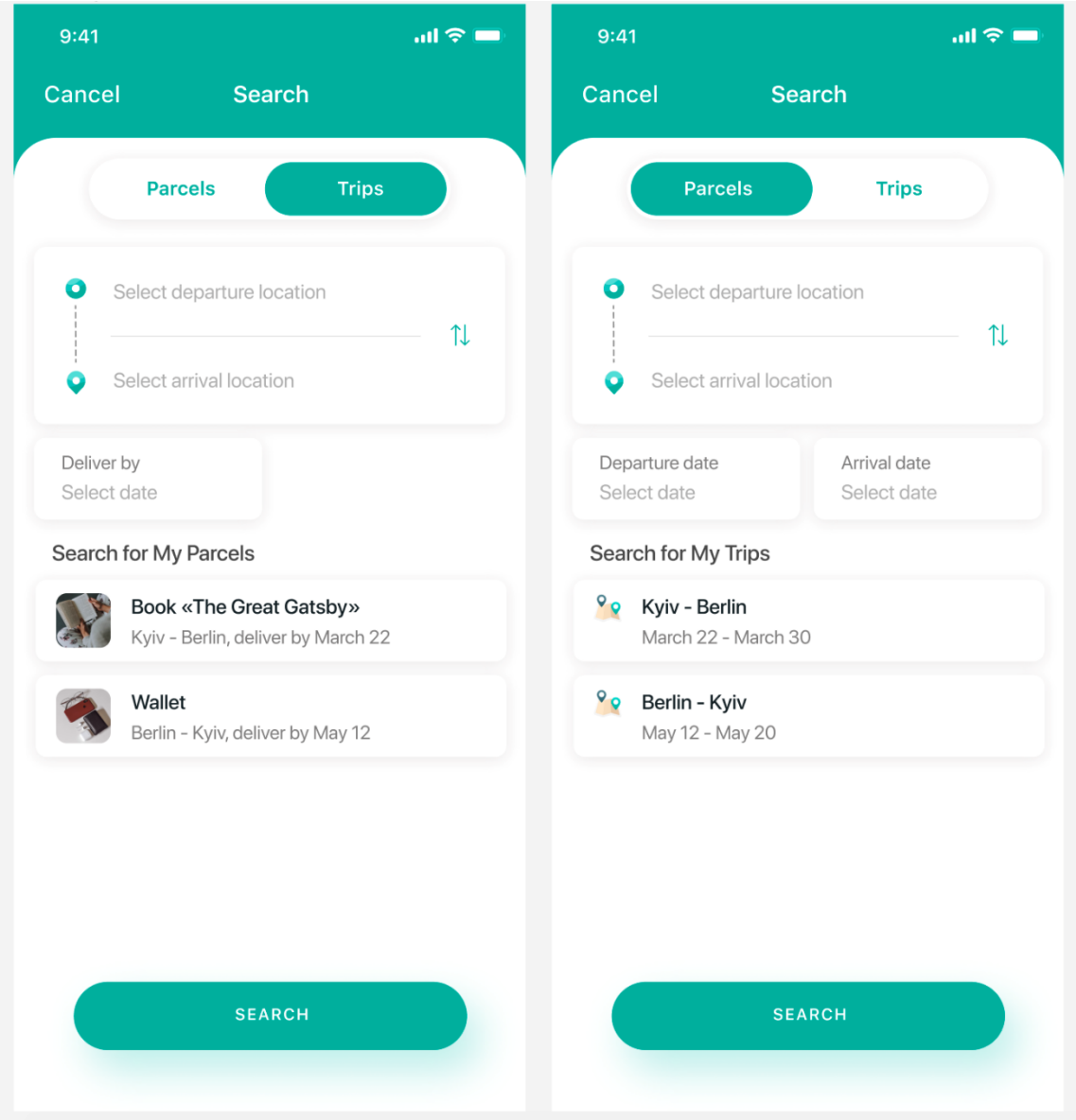


Рисунок 13. Розділ застосунку «Головна/Пошук»

При відображенні результатів пошуку посилок користувач буде бачити автора публікації, його рейтинг, короткий опис та фотографію товару, індикатор «Купи для мене» та винагороду за перевезення. Пост також можна додати в обране натиснувши на символ «сердечко» (див. рис. 14).

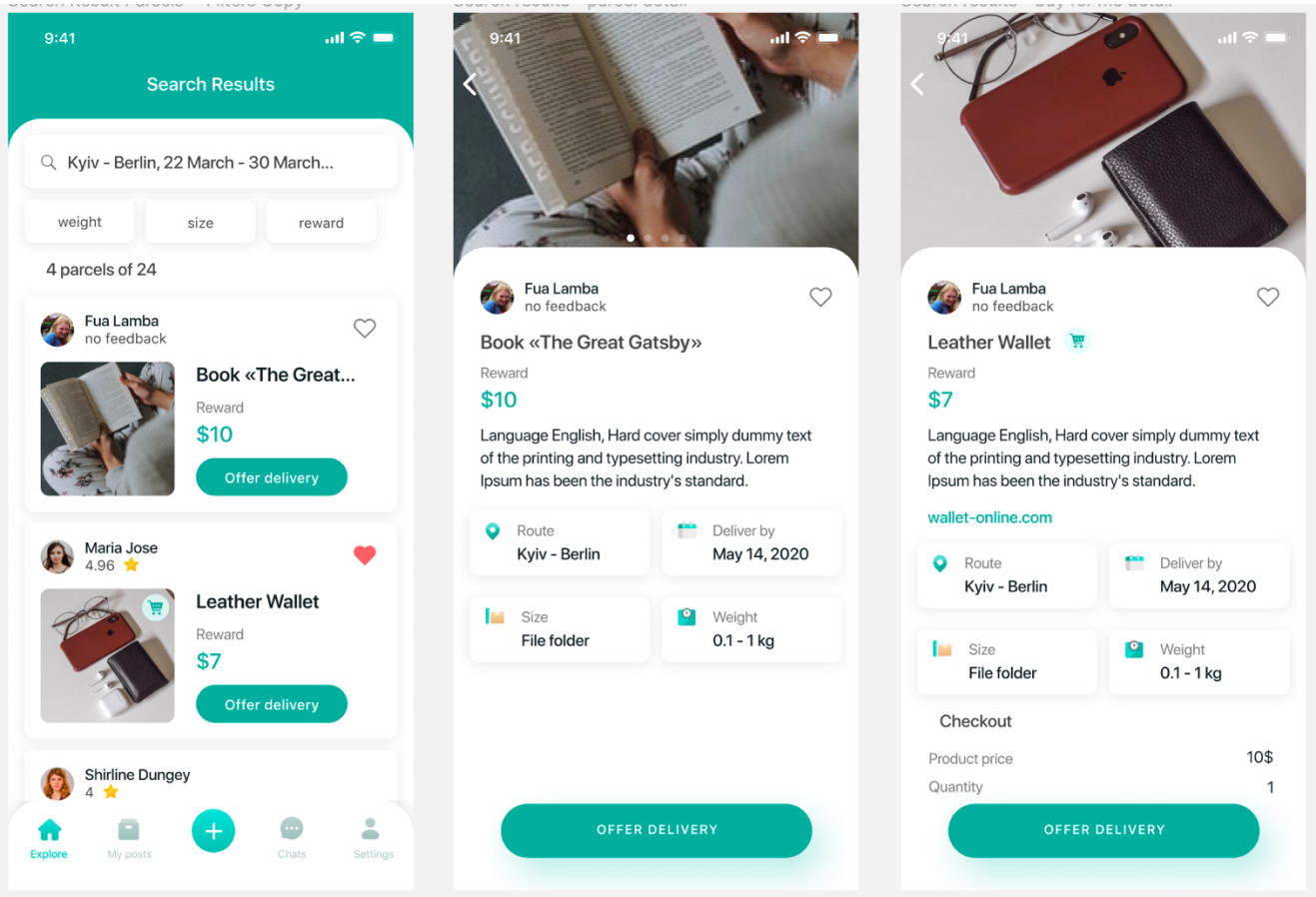


Рисунок 14. Розділ застосунку «Результати пошуку посилок»

При відображенні результатів пошуку подорожей користувач буде бачити автора публікації, його рейтинг, маршрут, дати а також обмеження на вагу і розмір перевезення (див. рис. 15).

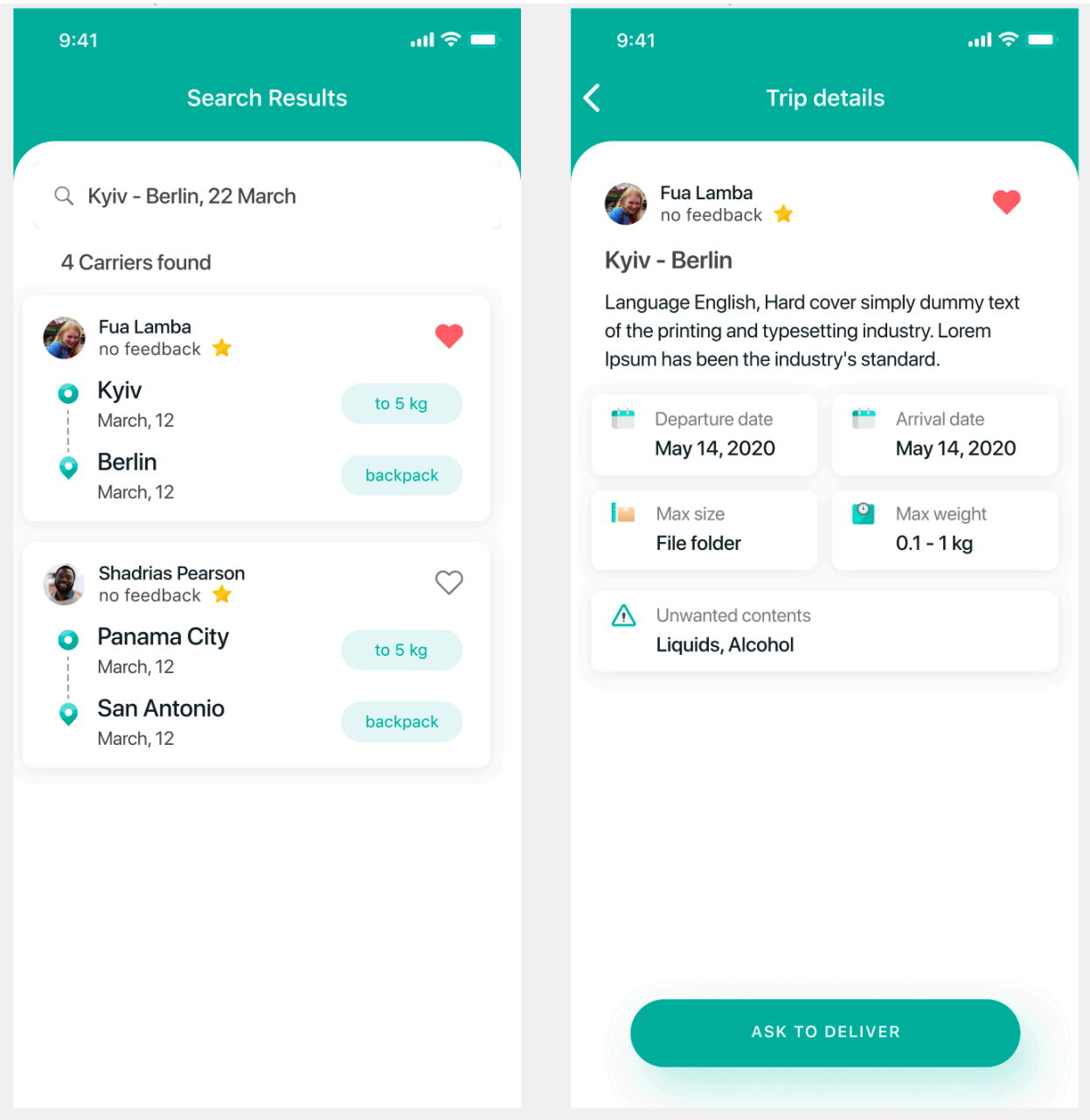


Рисунок 15. Розділ застосунку «Результати пошуку подорожей»

Якщо отримані результати задовольняють критерії пошуку користувача – він може запропонувати або запросити перевезення посилки (див. рис. 16).

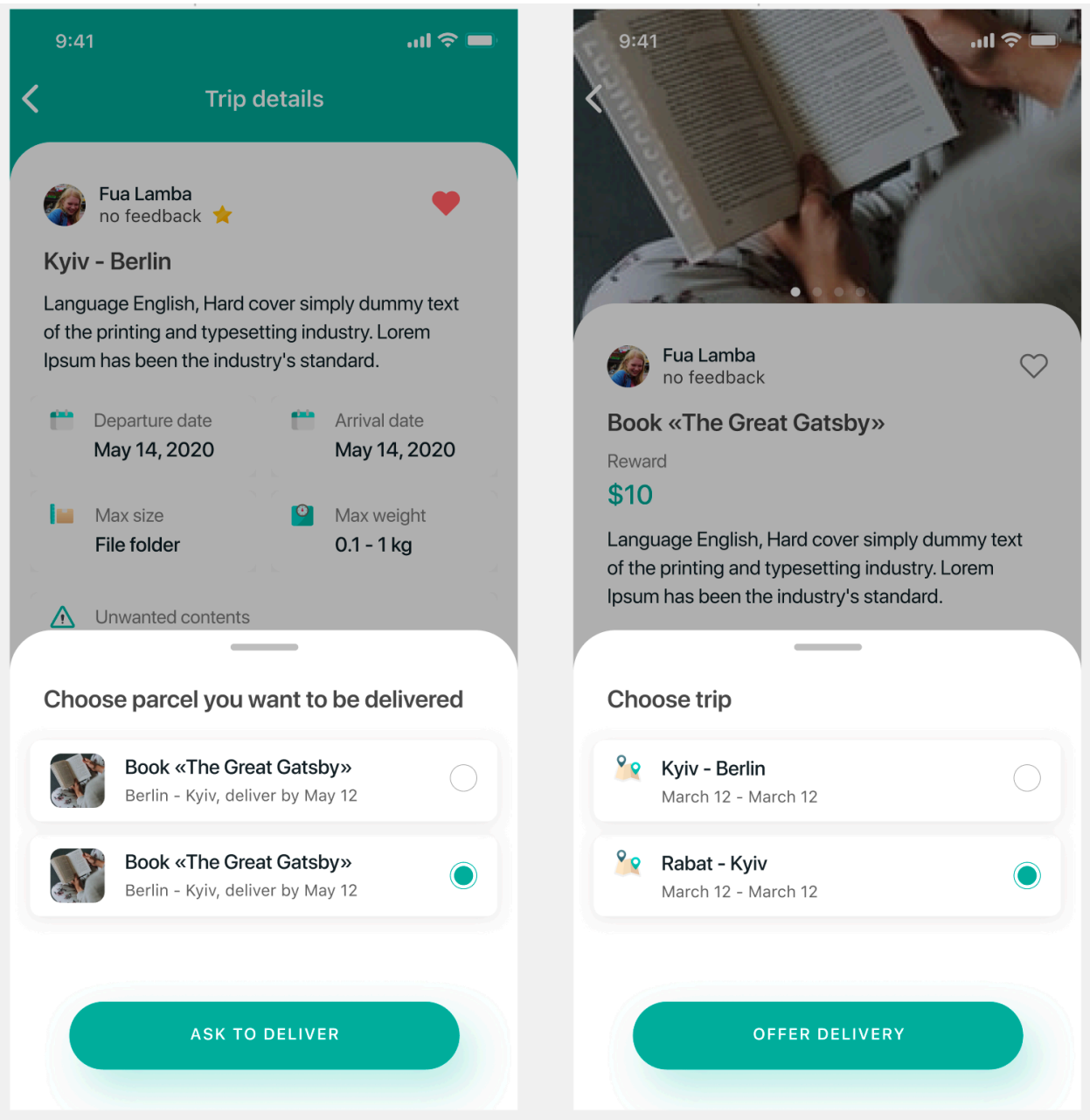


Рисунок 16. Розділ застосунку «Запросити/запропонувати перевезення посилки»

Після чого користувач і перевізник можуть перейти до вбудованого сервісу обміну повідомленнями (див. рис. 17). Детальну діаграму чатів схеми роботі чатів можна знайти у додатках Г-Д.

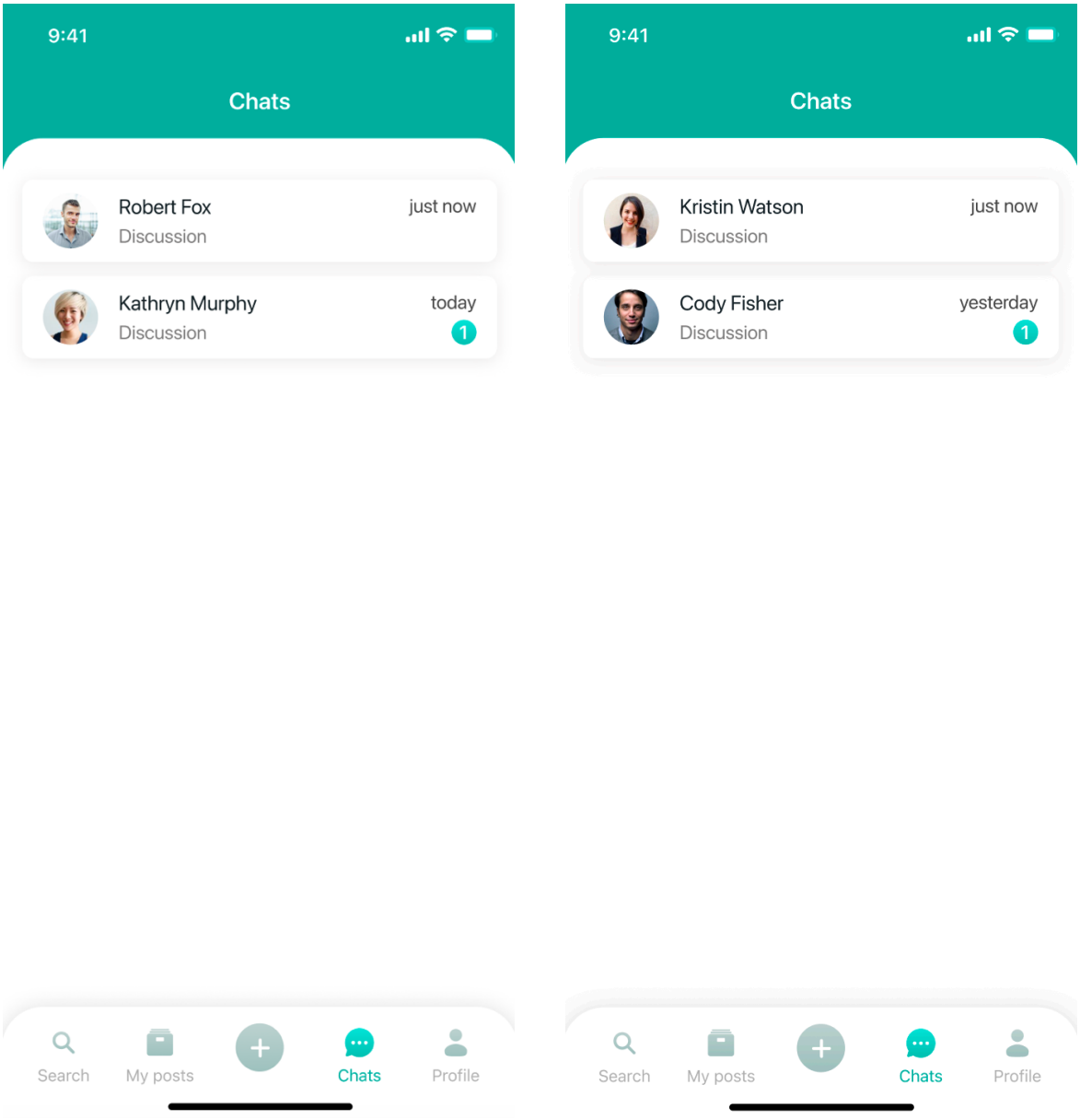


Рисунок 17. Розділ застосунку «Повідомлення»

Користувачі можуть обговорити деталі перевезення, обрати місце зустрічі для передачі посилки тощо (див. рис. 18).

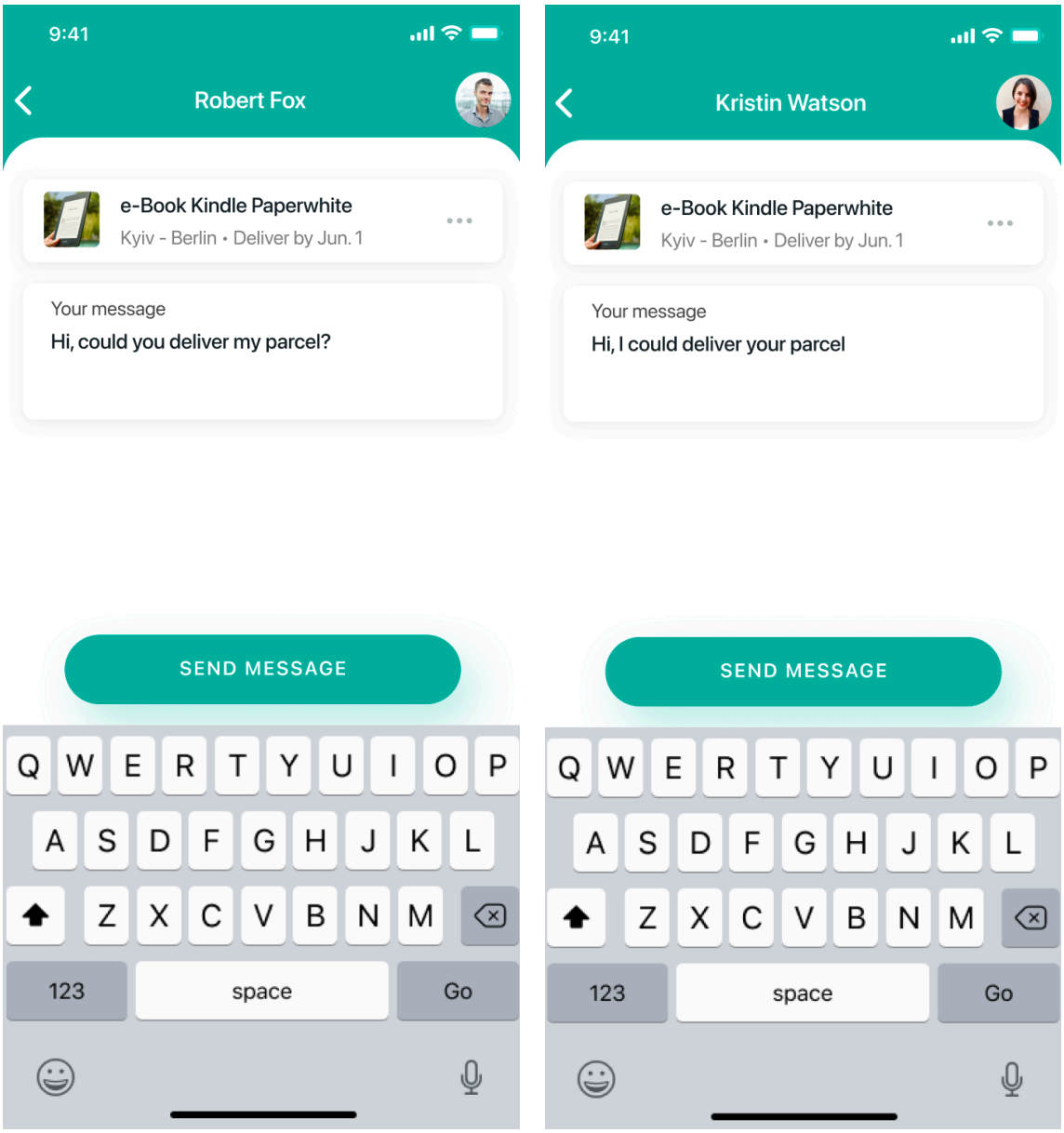


Рисунок 18. Розділ застосунку «Повідомлення»

Далі користувачі можуть перейти до оплати перевезення, не виходячи з чату. В сервісі обміну повідомлення знаходиться центр керування і відстеження посилки (див. рис. 19). Детальну діаграму станів посилки та подорожі можна знайти у додатках Е-Є.

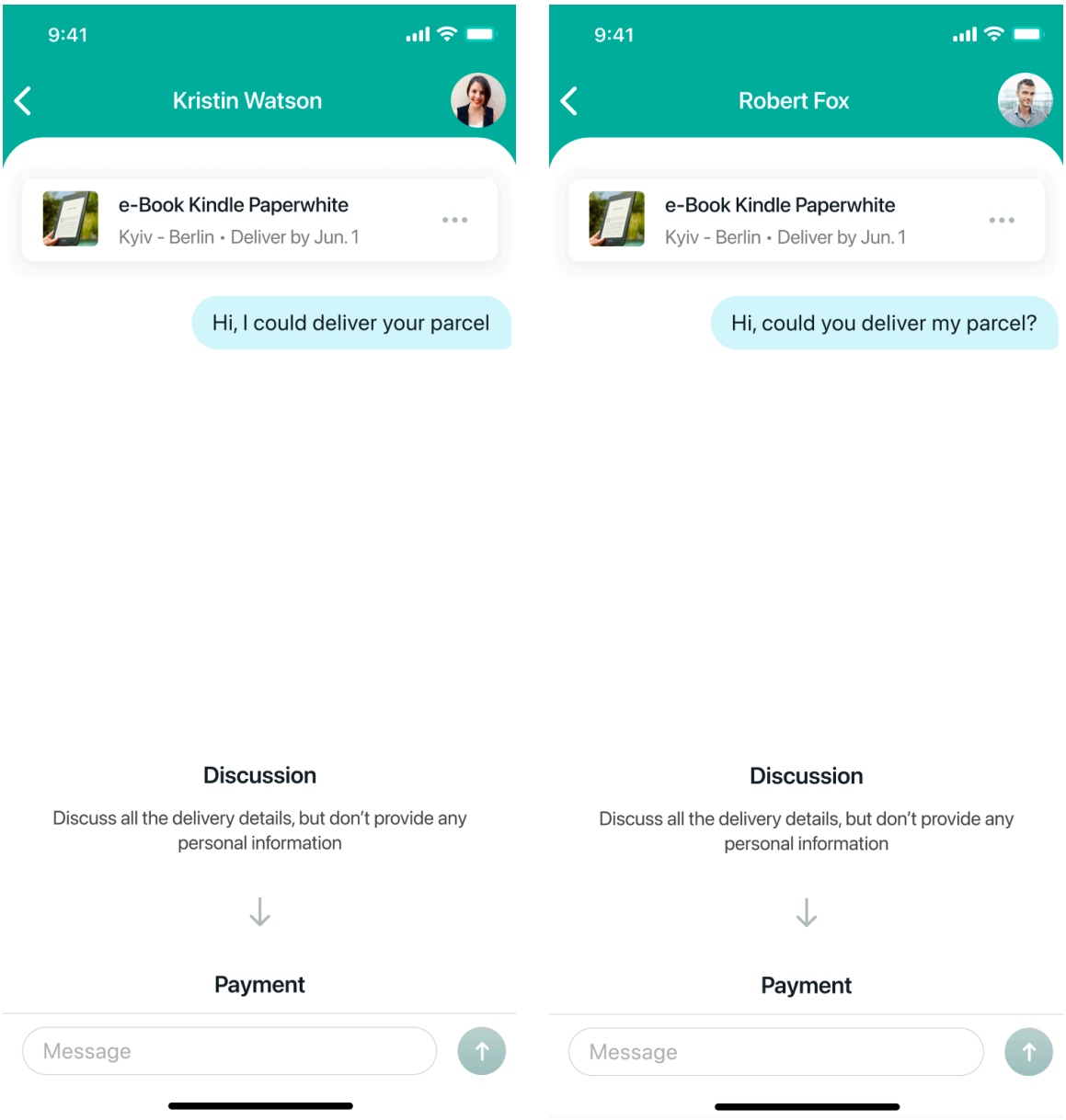


Рисунок 19. Розділ застосунку «Повідомлення»

Важливо відмітити те, що перш ніж здійснити оплату, користувачі повинні прийняти правила застосунку та погодитись з тим, що вони передають лише ті речі, які легально дозволено ввозити в країну призначення. (див. рис. 20). Після того як мандрівка завершена і посилка була успішно доставлена, користувачі можуть залишити відгуки один про одного. Детальніше у додатку Ж.

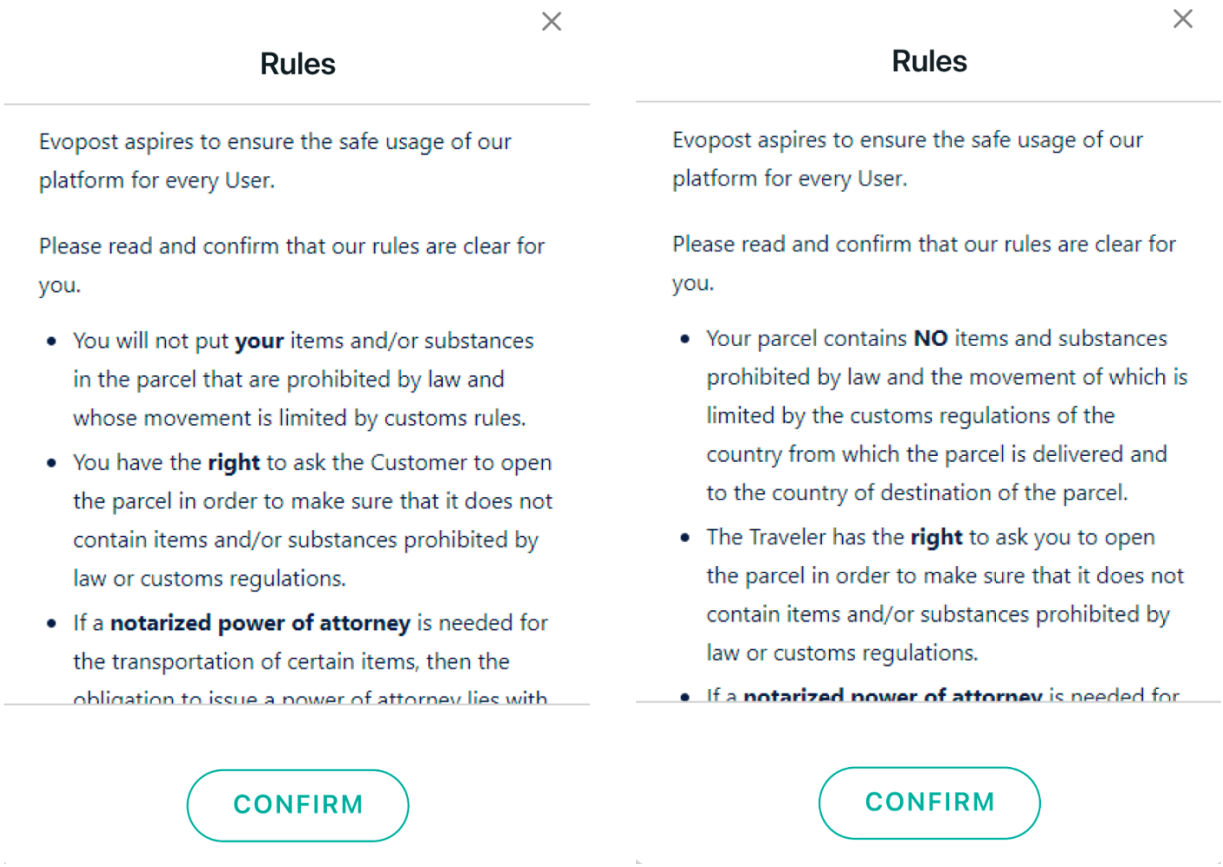


Рисунок 20. Розділ застосунку «Правила застосунку»

ВИСНОВКИ

В ході виконання дипломної роботи було досліджено сучасні веб-технології проаналізовано ринок сервісів міжнародної доставки та створено застосунок, який дозволяє мандрівникам доставляти посилки подорожуючи. При цьому було систематизовано, закріплено і розширено теоретичні та практичні знання інформаційних технологій та застосовано їх при розв’язанні задачі розробки застосунку з використанням технології Spring.

Вивчено та досліджено основні концепції та філософію фреймворку Spring. Використано сховище медіа файлів Cloudinary для збереження фотографій користувачів та посилок, платіжну систему Stripe для безпечних онлайн переказів та сервіс відправки СМС-повідомлень Vonage для верифікації користувачів. Реалізовано мобільну версію застосунку для зручного та інтуїтивного користування.

Під час написання дипломної роботи були покращені практичні навички програмування, застосовано теоретичні знання на практиці і розширено стек опанованих автором технологій. Також було здобуто досвід роботи у команді та проджект менеджменті.

Було здобуто навички у сфері бізнес девелопменту, маркетингу і залучення інвестицій у проєкт.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Офіційний сайт DHL [Електронний ресурс] // DHL. – 2020. – Режим доступу до ресурсу https://www.dhl.com/.
2. Офіційний сайт Hermes [Електронний ресурс] // Hermes. – 2020. – Режим доступу до ресурсу https://www.hermesworld.com/.
3. Офіційний сайт FedEx [Електронний ресурс] // FedEx. – 2020. – Режим доступу до ресурсу https://www.fedex.com/.
4. Pro Spring 4 / Кріс Шеффер, Кларенс Хо, Роб Харроп - К .: «Вільямс», 2017. - 752 с
5. Офіційний сайт проєкту Spring [Електронний ресурс] // Spring. - 2020 – Режим доступу до ресурсу: https://spring.io/.
6. Java Persistence with Hibernate (Second Edition of Hibernate in Action) / Christian Bauer and Gavin King – 2016. - 880 с.
7. Hibernate [Електронний ресурс] // Hibernate. – 2021. – Режим доступу до ресурсу: https://hibernate.org/.
8. Офіційний сайт PostgreSQL [Електронний ресурс] // PostgreSQL. – 2020. – Режим доступу до ресурсу: https://www.postgresql.org/.
9. PostgreSQL Up & Running // O'Reilly Media, Inc. – 2020. – 630 с.
10. Офіційний сайт Cloudinary [Електронний ресурс] // Cloudinary. – 2020. – Режим доступу до ресурсу https://cloudinary.com/.
11. Cloudinary [Електронний ресурс] // GitHub. – 2021. – Режим доступу до ресурсу: https://github.com/Cloudinary.
12. Офіційний сайт Stripe [Електронний ресурс] // Stripe. – 2020. – Режим доступу до ресурсу https://stripe.com/.
13. Stripe [Електронний ресурс] // GitHub. – 2021. – Режим доступу до ресурсу: https://github.com/Stripe/.
14. Офіційний сайт Vonage [Електронний ресурс] // Vonage. – 2020. – Режим доступу до ресурсу https://www.vonage.com/.

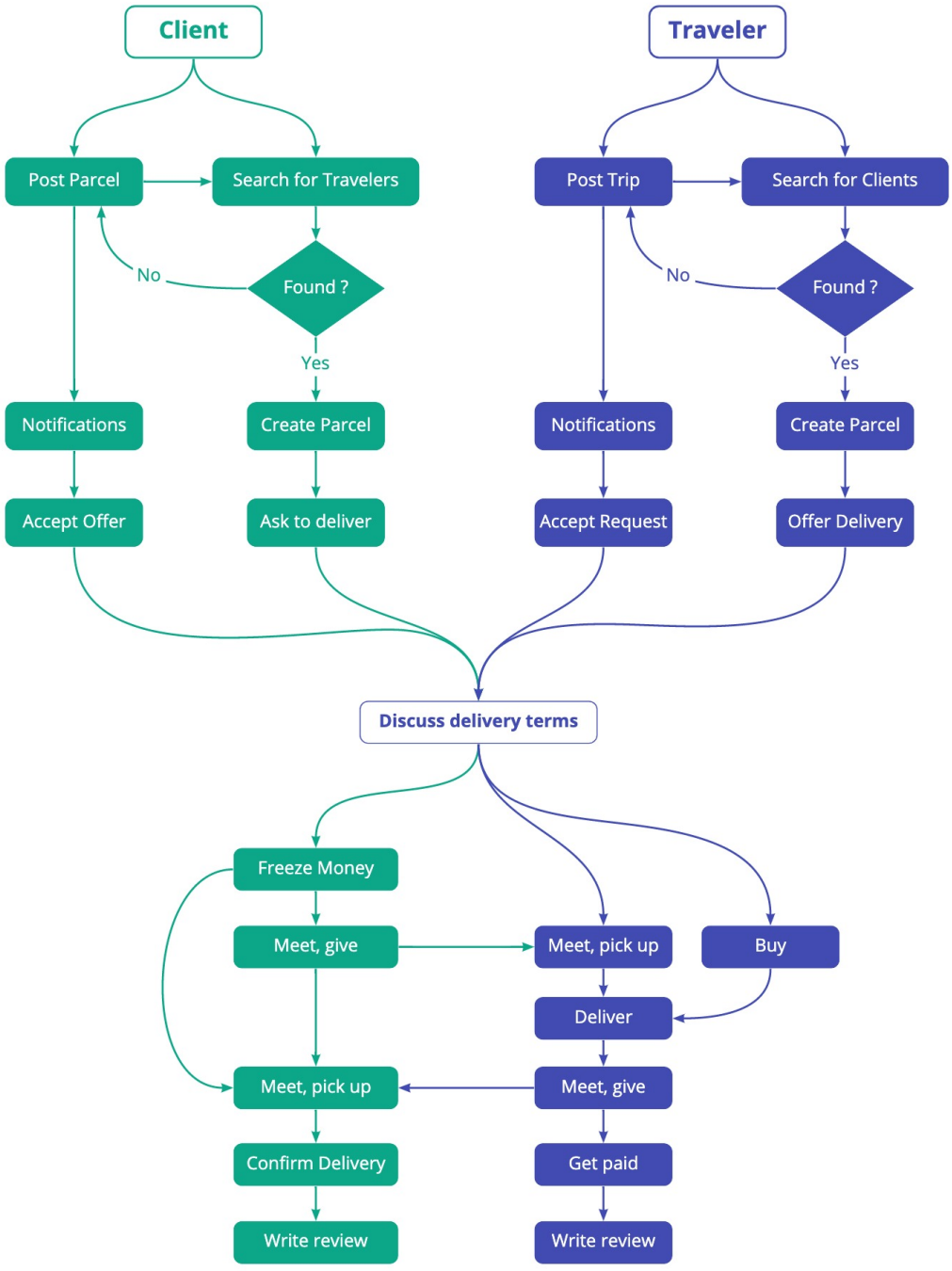
15. Vonage [Електронний ресурс] // GitHub. – 2021. – Режим доступу до ресурсу: https://github.com/Vonage
16. Mastering OAuth 2.0 [Електронний ресурс] // C. Bihis. – 2021. – 540 с.
17. Офіційний сайт PiggyBee [Електронний ресурс] // PiggyBee. – 2020. – Режим доступу до ресурсу: https://www.piggybee.com/en/.
18. Офіційний сайт BuddyExpress [Електронний ресурс] // BuddyExpress. – 2020. – Режим доступу до ресурсу: https://www.buddyexpress.com/.
19. Офіційний сайт HeyParcel [Електронний ресурс] // HeyParcel. – 2020. – Режим доступу до ресурсу: https://heyparcel.com/.
20. Офіційний сайт Zaldee [Електронний ресурс] // Zaldee. – 2020. – Режим доступу до ресурсу: http://www.zaldee.com/Home/.
21. Офіційний сайт Travoney [Електронний ресурс] // Travoney. – 2020. – Режим доступу до ресурсу: http://travoney.com/.
22. Офіційний сайт Koorier [Електронний ресурс] // Koorier. – 2020. – Режим доступу до ресурсу: https://apps.apple.com/us/app/koorier/id1240460103
23. Офіційний сайт AirWillBill [Електронний ресурс] // AirWillBill. – 2020. – Режим доступу до ресурсу: https://www.airwaybill.com/.
24. Офіційний сайт Nimber [Електронний ресурс] // Nimber. – 2020. – Режим доступу до ресурсу: https://www.nimber.com/.
25. Офіційний сайт TravelPost [Електронний ресурс] // TravelPost. – 2020. – Режим доступу до ресурсу: https://travelpost.io/en/.
26. Офіційний сайт Grabr [Електронний ресурс] // Grabr. – 2020. – Режим доступу до ресурсу: https://grabr.io/en/.
27. Журнал Inc: Cloudinary [Електронний ресурс] // Inc. – 2020. – Режим доступу до ресурсу: https://www.inc.com/travis-wright/how-image-optimization-can-manage-1000s-of-images-with-multiple-versions-across-.html/.
28. Офіційний сайт Cloudinary [Електронний ресурс] // Cloudinary. – 2020. – Режим доступу до ресурсу:

https://cloudinary.com/blog/cloudinary_s_jquery_library_for_embedding_and_transforming_images/.

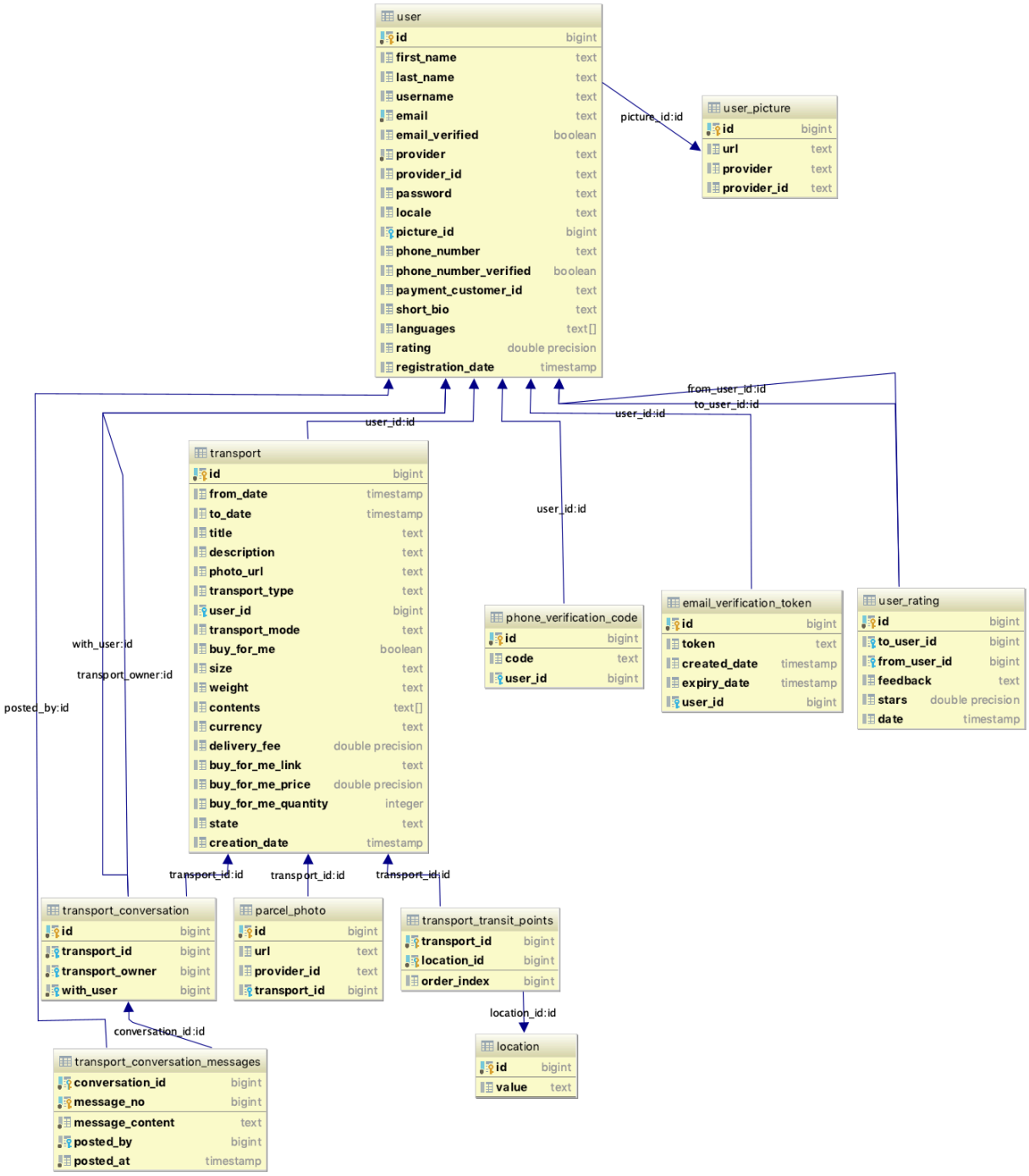
29. Офіційний сайт Github [Електронний ресурс] // Github. – 2020. – Режим доступу до ресурсу: https://github.com/.
30. Офіційний сайт Travis CI [Електронний ресурс] // Travis CI. – 2020. – Режим доступу до ресурсу: https://travis-ci.com/.
31. Офіційний сайт Swagger [Електронний ресурс] // Swagger. – 2020. – Режим доступу до ресурсу: https://swagger.io/tools/swagger-ui/.
32. WebSockets [Електронний ресурс] // Spring IO. – 2021. – Режим доступу до ресурсу: https://spring.io/guides/gs/messaging-stomp-websocket/.

ДОДАТКИ

Додаток А. Діаграма “User Flow”

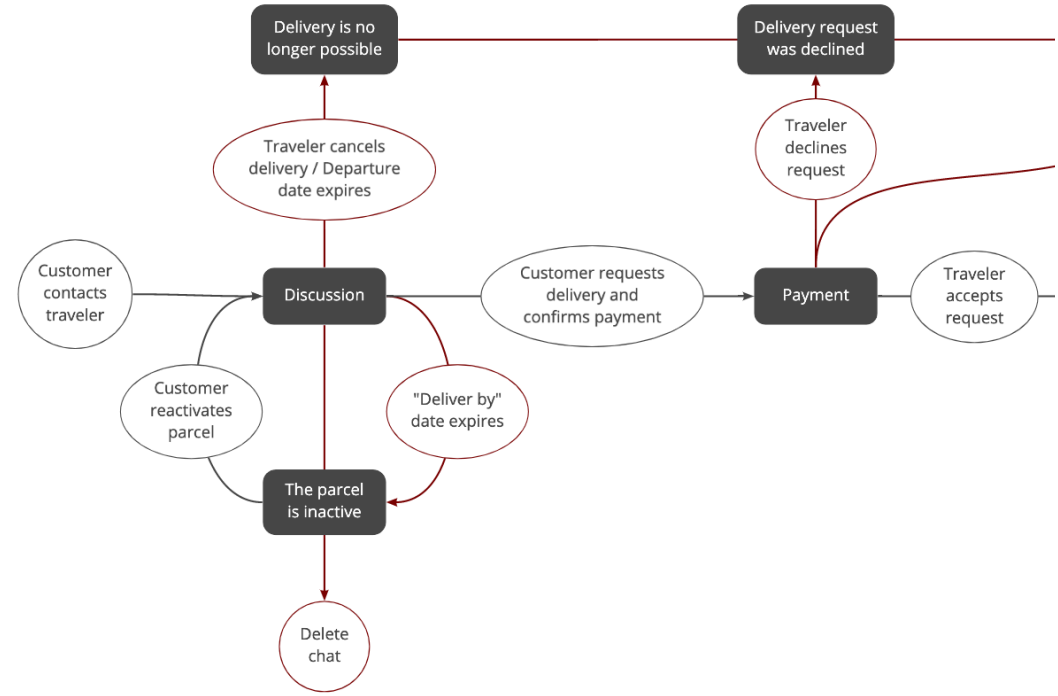


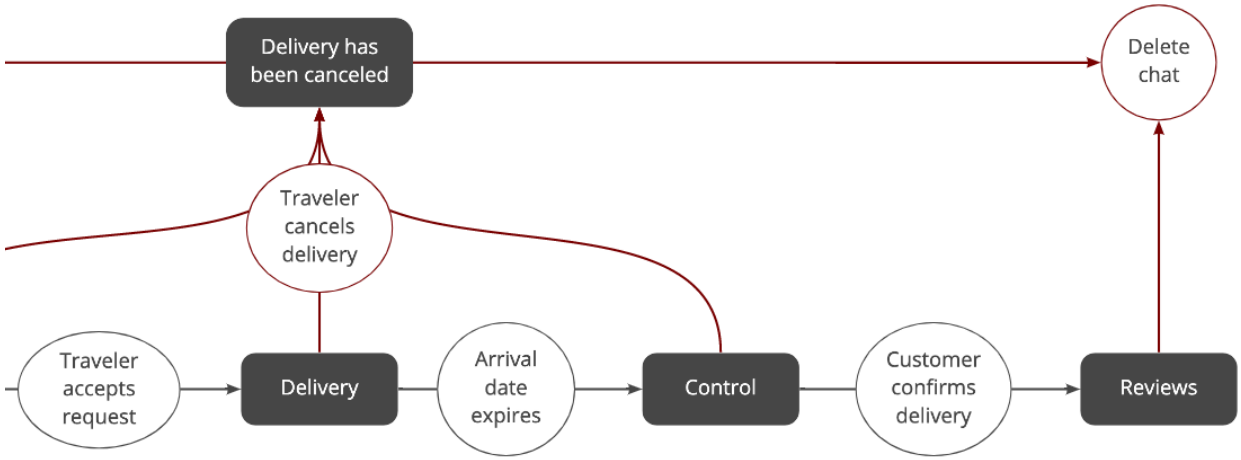
Додаток Б. Діаграма ERD



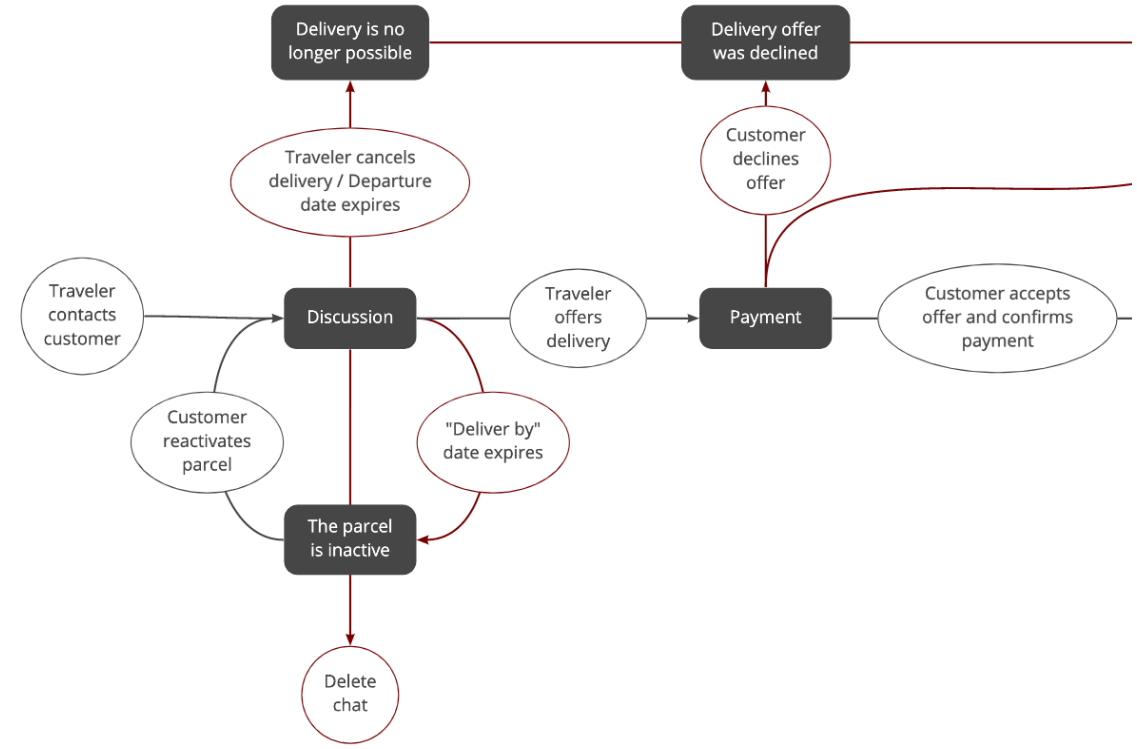
Додаток В. Діаграма “Target Customers”

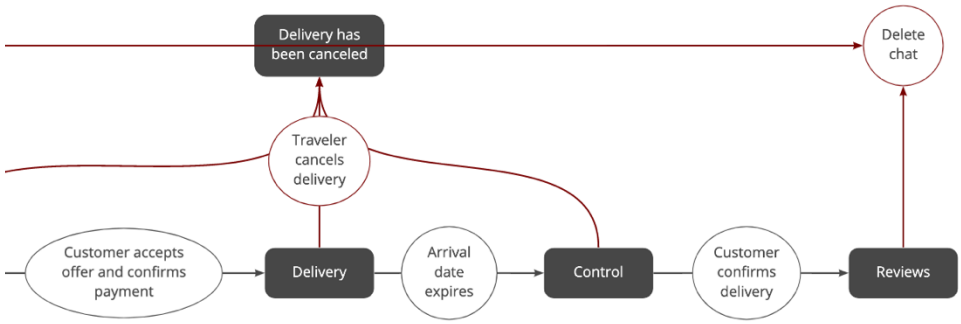
Додаток Г. Діаграма “Схема чату, що розпочав замовник”



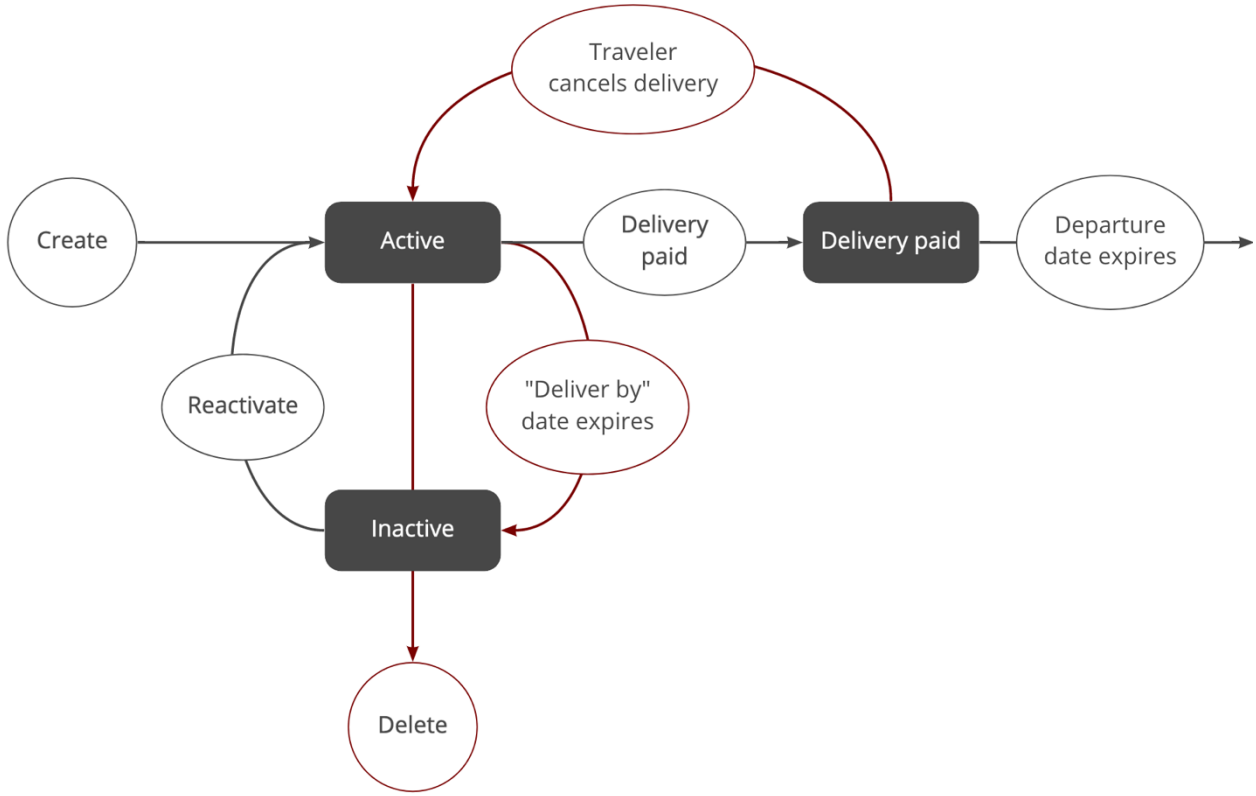


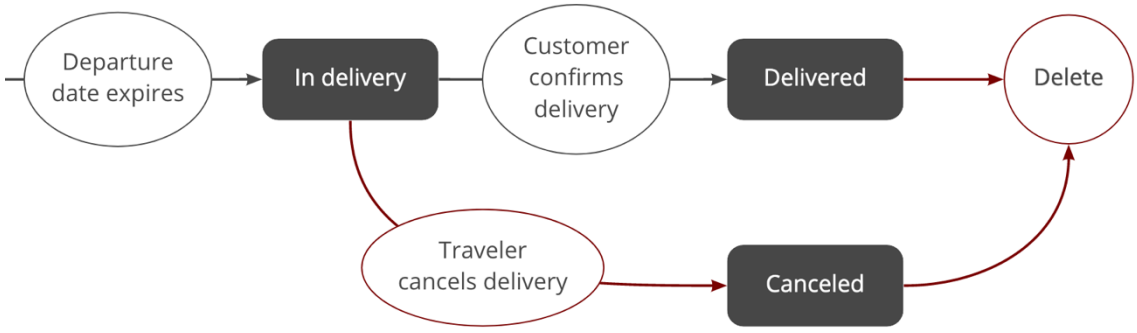
Додаток Д. Діаграма “Схема чату, що розпочав перевізник”



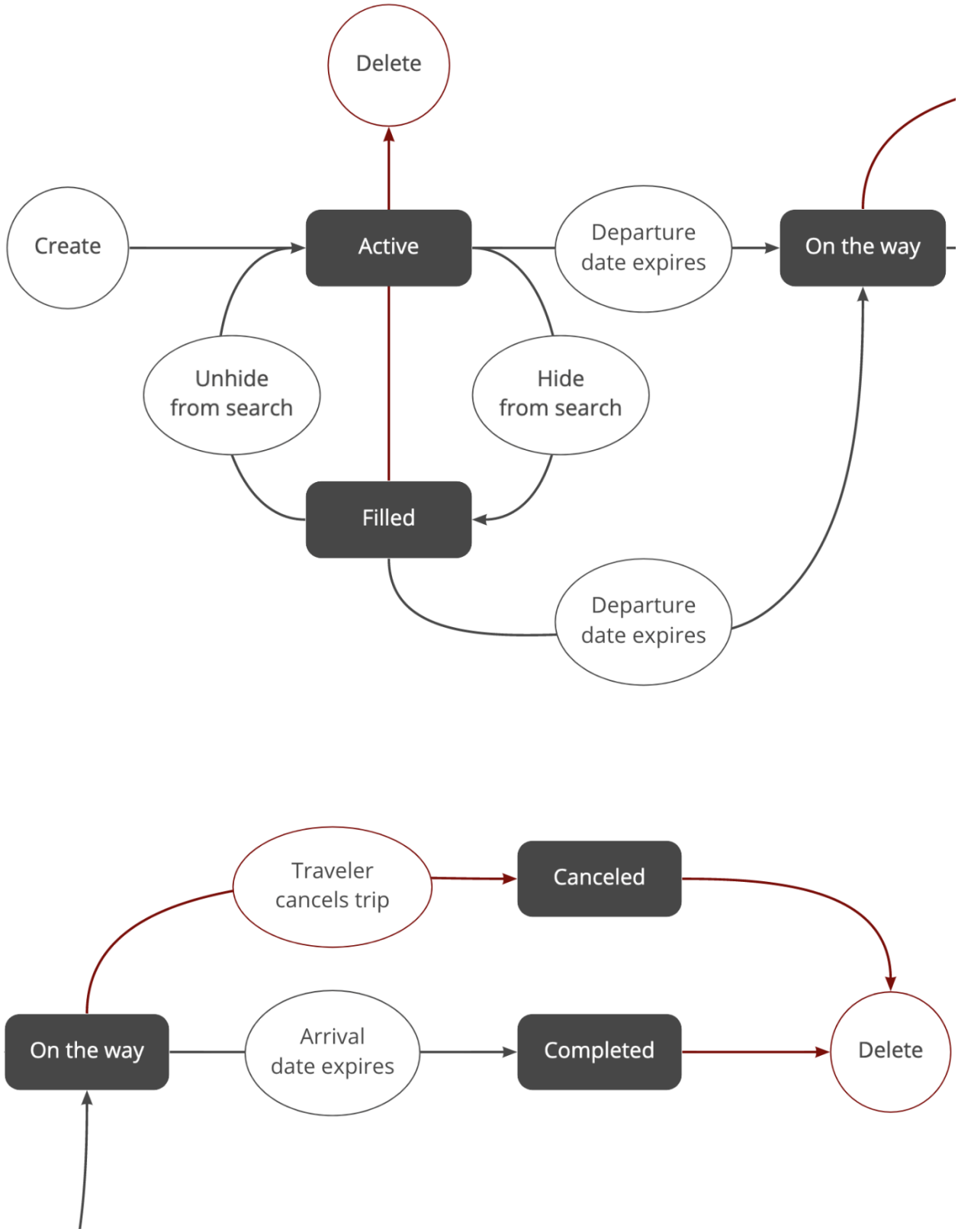


Додаток Е. Діаграма “Зміна станів посилки”





Додаток Є. Діаграма “Зміна станів подорожі”



Додаток Ж. Діаграма “Завершення подорожі”