

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки
Кафедра програмування та математики

ПОЯСНЮВАЛЬНА ЗАПИСКА
до кваліфікаційної випускної роботи

освітній ступінь бакалавр
спеціальність 121 „Інженерія програмного забезпечення”
(шифр і назва спеціальності)
спеціалізація „Інженерія програмного забезпечення”

на тему „Розробка додатку для побудови маршруту пересування залізничним транспортом”

Виконав: студент групи ІІЗ-16д

(підпис)

О.В. Сітченко

(ініціали і прізвище)

Керівник

(підпис)

Д.М. Марченко

(ініціали і прізвище)

Завідувач кафедри

(підпис)

В.О. Лифар

(ініціали і прізвище)

Рецензент _____

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра програмування та математики

Освітній ступінь бакалавр

спеціальність 121 «Інженерія програмного забезпечення»
(шифр і назва спеціальності)

спеціалізація «Інженерія програмного забезпечення»
(назва спеціалізації)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,

д.т.н., доцент

_____ Лифар В.О.

«__» _____ 2020 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ ВИПУСКНУ РОБОТУ СТУДЕНТУ

_____ Сітченко Олександр Валерійович _____

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка додатку для побудови маршрутів пересування залізничним транспортом.

керівник роботи професор, д.т.н. Марченко Дмитро Миколайович _____,
(вчене звання, науковий ступінь, прізвище, ім'я, по батькові)

затверджений наказом університету від «__» _____ 20__ року №__

2. Строк подання студентом роботи 20 травня 2020 р.

3. Вихідні дані до роботи Об'єктом даної роботи є процес циркуляції інформації.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Вступ

4.2 Аналітичний огляд галузі (огляд публічних джерел інформації), з висвітленням наступних питань:

1.Історичний розвиток.

2.Сучасні моделі.

4.3 Основна частина, в якій висвітлити:

1. Модель розробки програмного забезпечення

2. Основні вимоги до проєктованого програмного забезпечення.

3. Ризики проєкту

4.4 Практична частина – огляд технологій, які використовуються під час реалізації проєкту.

4.4 Висновки

4.5 Перелік використаних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників) _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 30 березня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	30.03.2020	
2	Укладання і погодження з керівником плану і етапів виконання роботи	06.04.2020	
3	Узагальнення даних літературних джерел	13.04.2020	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху виконання завдання	17.04.2020	
5	Вибір і укладання моделі розробки програмного забезпечення	20.04.2020	
6	Укладання та тестування програмного продукту	27.04.2020	
7	Укладання, оформлення та погодження пояснювальної записки з керівником	04.05.2020	
8	Задача пояснювальної записки на кафедрі	12.05.2020	
9	Підготовка доповіді та презентації	30.05.2020	

Студент

_____ (підпис)

О.В. Сітченко

(ініціали і прізвище)

Керівник роботи

_____ (підпис)

Д.М. Марченко

(ініціали і прізвище)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
дипломної роботи студента гр. ПЗ-16д Сітченко О.В.

Науковий керівник

Професор, д.т.н.

Марченко Д.М.

Оцінка наукового керівника: _____

Рецензент

_____ (ПБ, місто роботи, посада)

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК

(підпис)

Лифар В.О.

РЕФЕРАТ

Робота містить: 46 сторінок основного тексту, 5 схем, 14 рисунків, 8 таблиць, 21 використане джерело.

Метою випускної кваліфікаційної роботи є вивчення та аналіз особливостей створення систем обробки даних, підходів до їх реалізації та дослідження на прикладі створення інформаційної технології маршрутизації шляхів пересування залізничним транспортом

Був проведений детальний аналіз питання та шляхів його вирішення, створена інформаційна модель проєкту. Багато часу було приділено технічній частині, а саме пошуку шляхів отримання інформації для аналізу, її збереження, аналіз та розробка інтерфейсу для взаємодії з користувачем.

В результаті виконаної роботи був створений та реалізований алгоритм з побудови додатку з побудови маршрутів пересування залізничним транспортом.

Проєкт має великі можливості та декілька шляхів для подальшого розвитку.

Система реалізована відповідно всім вимогам технічного завдання.

Зроблено детальний опис алгоритму розробки системи, а також, продемонстрована робота android додатку.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД	7
1.1. ІСТОРИЧНИЙ РОЗВИТОК	7
1.2. СУЧАСНІ МОДЕЛІ	12
РОЗДІЛ 2. ІНФОРМАЦІЙНА МОДЕЛЬ ПРЕДМЕТНОЇ ГАЛУЗІ	19
2.1. МОДЕЛЬ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	19
2.2. ЗБІР ВИМОГ	21
2.3. АНАЛІЗ	22
2.4. ПРОЄКТУВАННЯ.....	25
2.5. РИЗИКИ ПРОЄКТУ	26
РОЗДІЛ 3. ВИБІР МЕТОДІВ ТА РЕАЛІЗАЦІЇ ПОСТАВЛЕНИХ ЗАДАЧ.....	27
3.1. WEB SCRAPING	27
3.2. МАШИННЕ НАВЧАННЯ.....	32
3.3. БАЗА ДАНИХ	36
3.4. АЛГОРИТМ РОЗРАХУНКУ МАРШРУТУ	42
3.5. ANDROID STUDIO	44
3.6. ОПИС ІНТЕРФЕЙСУ ДОДАТКУ	46
ВИСНОВКИ.....	49
СПИСОК ЛІТЕРАТУРИ.....	50

ВСТУП

Актуальність даної теми обумовлена тим, що наявні інтернет-ресурси не дозволяють побудувати маршрут пересування залізничним транспортом на території України з використанням приміського залізничного транспорту. Наприклад, на сайті Укрзалізниці ми маємо можливість переглянути тільки визначені маршрути пересування пасажирських поїздів, а в випадку відсутності квитків наявна можливість знаходження квитка з пересадками. Інші сайти надають ту ж саму інформацію, беручи інформацію з сайту Укрзалізниці. На деяких з них за відсутності квитків пропонується альтернатива (BlaBlaCar).

Однак, середньостатистичній людині набагато зручніше зробити декілька пересадок використовуючи один вид транспорту, якщо є така можливість, ніж використовуючи декілька видів транспорту. Але через те, що наявні системи не охоплюють увесь спектр залізничного транспорту людина, у випадку відсутності квитка, змушена використовувати або декілька видів транспорту або відмовитися від поїздки.

Об'єктом дослідження є процеси циркуляції інформації.

Предметом дослідження є модель та алгоритм інформаційної технології маршрутизації шляхів пересування залізничним транспортом

Метою дослідження є створення моделі та реалізація алгоритму, що дозволить побудувати додаток з побудови маршрутів пересування залізничним транспортом.

Відповідно до мети роботи поставлені такі завдання дослідження:

- Провести аналіз сучасних підходів, методів та моделей до побудови додатків з маршрутизації шляхів пересування транспортом
- Провести аналіз системи з управління проектом для побудови маршрутів пересування приміським залізничним транспортом

- Сформувати підхід до побудови додатку для побудови маршрутів пересування залізничним транспортом

Проведені дослідження базуються на загальній теорії управління складними системами.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД

1.1. ІСТОРИЧНИЙ РОЗВИТОК

Розробка маршрутів пересування людей є одним з завдань логістики.

Логістика (англ. logistics, від грец. λογιστική - облік)[**Error! Reference source not found.**] є наукою про планування, управління та контроль за матеріальними, інформаційними та фінансовими потоками в економіці. Також логістикою називають будь-які процеси пов'язані з життєвим циклом будь-яких предметів, а саме їх транспортування, обробка та зберігання.

Зараз у переважній більшості наукових робіт поява і розвиток логістики пов'язується з розвитком військового мистецтва. Однак сучасна розшифровка глиняних табличок, знайдених археологами в 1928 р. в руїнах міста Угаріт, датованих XIV століттям до н.е., свідчать про існування в той час науки про переміщення товарів усередині самого міста й між містами суміжних держав по суші та морськими шляхами [1].

Розвиток логістики також невід'ємно пов'язаний з розвитком інфраструктури, торгівлі та появленням нових видів транспорту.

В історії людина завжди намагається полегшити своє життя. Так ще у древній Месопотамії у IV тис. до н. е. з'явився перший тягловий транспорт. Це призвело до розвитку будівництва доріг, що вилилося в появу "Великого шовкового шляху" та будівництво в 312 році римських доріг. Для доріг обиралися найкоротші відстані, попри різні перешкоди. В населених пунктах дороги мали тротуари, що були завширшки 4 - 5 м та мали на меті розмежувати рух між пішоходами й повозками. Узбіччях всіх доріг мали дорожні стовпи - герми, на яких було вказано рік встановлення, ім'я правлячого імператора та відстань до найближчого міста. В Римській імперії була створена перша централізована мережа доріг, що об'єднала між собою Піренейський, Апеннінський та Балканський півострови, Малу Азію й Близький Схід, Британію та Францію, Африканське узбережжя Середземного моря. Центром

всієї мережі доріг виступав Рим. Паралельно цьому розвивалося мореплавство, особливий вклад в який внесла Фінікія. Фінікійці відкрили велику кількість морських шляхів та впливали на розвиток мореплавства в інших країн стародавнього світу [3].

Попри загальний занепад Середньовіччя в порівнянні зі Стародавнім світом, розвиток інфраструктури продовжувався. Завдяки збільшенню міст та початку спеціалізації промисловості в них, розвивалося торгівля, поява постійних дворів та розвиток освіти призвели до збільшення кількості мандрівників, а велика кількість військових походів призвела до появи логістичних проблем. Саме в цей час в мореплаванні були закладені підґрунтя, що призвело надалі до Великих географічних відкриттів, а саме блокування арабами вільних торговельних шляхів до Індії [3].

Завдяки Великим географічним відкриттям, в епоху Відродження з'явилося світова торгівля. Було відкрита величезна кількість морських шляхів, подорожі стали носити більш організований характер, вдосконалювалися транспортні засоби та розвивалась інфраструктура для мандрівників.

Попри свій розвиток мореплавство не мало можливості пов'язувати міста, які не мали вихід до моря, а наявні тяглові види транспорту вже не відповідали наявним потребам. Це призвело до появи у XVII столітті Воллатонської вагонної дорозі, що вважається першою залізницею у світі. У XIX столітті в містах за принципом Воллатонської вагонної дороги була влаштована конка, що вважається попередницею трамваю. У 1825 році Джеймс Ватт і Джордж Стефенсон побудували першу залізницю, що з'єднала Стоктон та Дарлінгтон. Після цього розпочинається інтенсивне будівництво залізниць, як більш заощадливого і швидкого виду транспорту та зв'язку на той час. Після Другої світової війни паротяги були замінені електротягами і теплотягами[4].

Ще в кінці 18 століття були спроби створення механізованого автомобільного транспорту - паровий візок Жозефа Кюньо 1770 рік, але лише

під кінець 19 століття автомобільний транспорт розпочинає поширюватися. В 1900 році в усьому світі нараховувалось всього 11 тисяч автомобілів, але вже через 57 років ця цифра зросла більш ніж в 9 тисяч разів та становила 102 мільйона автомобілів. На бурхливий розвиток автомобільного транспорту добре сприяла його висока прохідність і маневровість, порівняно висока швидкість перевезення та можливість доставлення вантажів без перевантаження, відносно невелика вартість перевезення, можливість спеціалізації автомобілів для перевезення однотипних вантажів, простота керування і догляду за автомобілями[**Error! Reference source not found.**].

Одночасно з розвитком автотранспорту розпочинає свій розвиток і авіатранспорт. Такий вид транспорту стає найшвидшим, але в той же час і найдорожчим. Люди вперше почали літати, використовуючи повітряні кулі та відкрили добу польотів, 5 червня 1783 року, а дещо пізніше були створені дирижаблі з двигунами. Винахід бензинового двигуна надає можливість братам Райт у 1903 здійснили перший керований політ. Ця подія мала велику значимість та підштовхнула розвиток літакобудування. Саме після неї найбільш широко для перевезення людей, а також і в військових цілях, почали використовуватися літаки. Перший військовий реактивний літак Мессершмітт було створено в 1939 році. По закінченню Другої світової війни жваво починається розвиватися цивільне літакобудування, і вже в 1969 році було створено перший надзвуковий пасажирський літак Конкорд [5].

Зважаючи на все це є можливим виділити такі етапи розвитку й становлення логістики[1]:

1. Перший етап, період з XIV ст. до н.е. – до I ст. н.е.. Його можливо охарактеризувати використанням логістичного підходу для управління рухом товарів усередині держав (між містами) та між державами, тобто відбувався розвиток логістики на макро- і мікрорівнях. У цей період зародилися перші принципи логістики та були розроблені елементарні методи з розрахунку маршрутів доставлення товарів сухопутними та морськими шляхами; під час керуванні переміщенням армій, розпочинається використання логістичних

підходів до ведення військових операцій та їх матеріально-технічного постачання.

2. Другий етап, I тисячоріччя н.е. – кінець XIX ст. У цей період відбувається становлення логістики як військової науки. Принципи логістики розвиваються завдяки використанню під час керування військами та їх матеріально-технічного постачання.

3. Третій етап у розвитку логістики розпочинається з початку 1900 р. і продовжується до середини XX століття. Відбувається етап запозичення та застосування всього здобутого теоретичного та практичного досвіду, який було отримано у військовій справі, у всі галузі економіки, вивчаються та трансформуються операції та процеси, пов'язані із переміщенням товарно-матеріальних ресурсів в просторі та часі. Розвиток логістики, в цей період, пов'язують з ланцюгом економічних криз, які послідовно виникали на світовому ринку в період з 1900 р. до 1903 р., та в 1907 р. і 1920 р. Також саме в цей період відбувається поява перших логістичних спілок. Однією з таких спілок є Національна асоціація проблем управління запасами, що була заснована у США в 1915 р., а потім в 1967 р. була перетворена в Національну асоціацію агентів постачання. Хвиля економічних криз змусила уряди країн Західної Європи і США шукати рішення проблеми щодо стабілізації економіки. Саме тоді учені та фахівці зосередили свою увагу на розвитку логістики.

4. Четвертий етап обмежується 50-ми й початком 80-х рр. XX століття. Початок другої половини XX ст. ознаменовано створенням у США Національної ради з менеджменту фізичного розподілу, яка пізніше була перетворена у Раду логістичного менеджменту. У СРСР, а саме в Ленінграді наприкінці 1970-х років була розроблена унікальна логістична технологія. Суть її зводилася до того, що робота багатьох видів транспорту організовувалася по методу транспортного вузла, де і здійснювалася їхня взаємодія. На даний період часу починають щорічно проводитися міжнародні симпозиуми, конференції й конгреси, що були зосереджені на обговоренні

наукових та практичних проблем логістики. Так, нині всім відомі: представницький форум з логістики, що відбувся в жовтні 1989 року в Парижі та семінар з логістики, який було проведено у Москві водночас. Саме під кінець четвертого етапу з'явилися такі логістичні системи й технології, як:

- Materials Requirements Planning - система планування потреби в матеріалах, деталях і вузлах;
- Distribution Requirements Planning - система планування розподілу ресурсів;
- Optimized Production Technologies - оптимізовані виробничі технології;
- Just-in-time - система точно вчасно.

5. П'ятий етап – сучасний період розвитку логістики – почався із середини 80-х рр. ХХ ст. та триває по нині. Саме зараз досить інтенсивний розвиток одержали методи керування матеріальними потоками. Крім цього, масовий розвиток і використання інформаційних технологій практично всіма учасниками ринку та створення сучасних комунікаційних технологій підсилюють розуміння того, що високої ефективності в керуванні матеріальними потоками в ланцюзі "закупівля – виробництво – дистрибуція – продажі" можна досягти внаслідок керування потоками інформації. П'ятий етап розвитку логістики вважають всесвітнім через те, що логістизація економічних процесів стає проблемою США, Європи й Азії. Країни, що розвиваються, і країни з перехідною економікою активно включилися в процес логістизації.

1.2. СУЧАСНІ МОДЕЛІ

З появою перших електронних обчислювальних систем починається епоха автоматизації. Автоматизація торкнулася не тільки промисловості, но і транспортні системи. Так виникли поняття ITS та ІТС.

Інтелектуальна транспортна система (ITS, англ Intelligent transportation system)[6] — це транспортна система, в якій використовуються інноваційні розробки в моделюванні і регулюванні транспортних потоків. Це дає кінцевим споживачам більшу інформативність і безпеку, а також якісно підвищує рівень взаємодії між учасниками руху в порівнянні зі звичайними транспортними системами.

Інтелектуальні транспортні системи (абр. ІТС)[6]— це системна інтеграція сучасних інформаційних, комунікаційних технологій і засобів автоматизації з транспортною інфраструктурою, транспортними засобами та користувачами, яка орієнтована на підвищення безпеки й ефективності транспортного процесу, комфортності для водіїв та користувачів транспорту.

Розвиток ITS розпочався в США в 1986 році групою вчених, що зібралась для обговорення подальшого напрямку розвитку сухопутних транспортних засобів. Необхідність вдосконалення викликали серйозні проблеми з затримками транспорту в годину пік. Особливо критична ситуація була у містах, де час пік закінчувався пізно вночі. Крім цього, побоювання викликала безпека руху на магістралях, та екологічний збиток, який наноситься навколишньому середовищу. Останньою причиною є фактор міжнародної конкурентоспроможності – основні конкуренти США вже впроваджували сучасні системи управління пасажиропотоком, які будувалися на основі принципу тотальної інформатизації. Ці причини й стали основою переходу від політики екстенсивного збільшення кількості доріг в бік підвищення якості та пропускну здатності. Результатом став вихід закону ISTEA (Intermodal Surface Transportation Efficiency Act) та впровадження проєкту Intelligent Vehicle Highway Systems вартістю 660 млн доларів в 1991 році. У цьому акті, зокрема, говорилося, що його метою є «... розвиток

економічно ефективної транспортної системи, що послужить опорою для успішної конкуренції країни на світовому ринку і буде обслуговувати споживачів і перевозити товари з максимальним енергоощадження» [6].

Зараз в багатьох країнах світу, в тому числі державах Євросоюзу та Україні, зростає розуміння важливості розв'язання глобальних проблем транспортних комплексів. Це, перш за все, пов'язано з вимогами підвищення безпеки та ефективності перевезень, з ростом мобільності суспільства, необхідністю зменшення впливу транспорту на навколишнє середовище та інших. У розв'язанні цих проблем найважливіше місце займає створення і використання інтелектуальних транспортних систем. Вони акумулюють і інтегрують передові досягнення сучасних досліджень і розробок в областях телекомунікацій, інформаційних технологій, методів інтелектуальних систем, супутникових технологій позиціонування, географічних інформаційних систем. В цілому розробка ІТС відповідає тенденціям розвитку інформатизації суспільства, підвищення рівня інтелектуальності систем управління. Зауважимо, що зараз загальнодержавна транспортна політика багатьох розвинених країн базується на розробці та просуванні інтелектуальних транспортних систем. Створення і застосування ІТС служить важливим стимулом для розвитку інноваційних технологій ряду галузей промисловості.

Сучасний залізничний транспорт є надзвичайно складною організаційно-технічною системою, управління якою зараз практично неможливо в рамках раніше сформованих традиційних підходів. Складність транспортної інфраструктури та її об'єктів (залізничні вузли, станції, транспортні коридори і т.д.) принципово виключає можливість роботи в повністю автоматичному режимі. Ефективне управління такою системою із залученням тільки класичних методів вирішення складних завдань математичного моделювання неможливо, і тут великі надії покладаються на інтелектуальні системи, які поряд з точними математичними моделями використовують дані та знання, накопичені в процесі їх діяльності. Стосовно до залізничного транспорту метою розробки інтелектуальних транспортних

систем є наступне: неухильне виконання вимог щодо забезпечення безпеки перевезень, скорочення рівня впливу на навколишнє середовище, істотне підвищення ефективності виробничої діяльності в цілому[8].

Негативні наслідки впливу транспорту, масштаби та важливість яких дають підставу оцінювати їх як стратегічний виклик національного і майже континентального масштабу. До них можна віднести неприпустимий рівень людських втрат, негативний вплив на навколишнє середовище та ріст споживання невідновлюваних джерел енергії, затримки людей і вантажу на всіх видах транспорту, що пов'язано як з об'єктивною нестачею потужностей транспортної інфраструктури, так і з низьким рівнем керування транспортними потоками. Світовою транспортною спільнотою вирішення цих проблем було знайдено у переході від створення систем управління транспортом до створення транспортних систем, в яких засоби зв'язку, управління і контролю спочатку вбудовані в транспортні засоби й об'єкти інфраструктури, а можливості управління (прийняття рішення), на основі інформації, отриманої в реальному масштабі часу, доступні не тільки транспортним операторам, але й всім користувачам транспортними засобами, які є учасниками цих систем. Вирішення завдання знайшли в побудові інтегрованої системи, яку можливо уявити в вигляді зв'язку між людиною та транспортним засобом, з максимальним використанням новітніх інформаційно-керівних технологій. Такі системи й стали називати інтелектуальними. Термін «інтелектуальні транспортні системи» був запропонований японською стороною на конференції в Йокогамі. Зараз поняття ІТС є досить широким і недостатньо чітко визначеним[6].

Понад 20 років тому багато країн активізували свої зусилля з перекладу організації дорожнього руху та керування транспортними потоками на новий рівень з максимальним використанням сучасних інформаційних і комунікаційних технологій. З середини 80-х років ХХ століття в Японії, США, Європі почалися великомасштабні роботи по створенню і розвитку ІТС або систем транспортної телематики. У 1973 році Японія одна з перших країн у

світі приступила до науково-дослідних і конструкторських розробок по ІТС. Міністерство міжнародної торгівлі цієї країни прорахував необхідність розвитку САС (Комплексна система управління автомобільним транспортом), були виділені величезні кошти, залучені кращі фахівці. За 40 років країна стала світовим лідером в області керування транспортними потоками, і сьогодні проблем з транспортними «пробками» в Токіо не існує, але є унікальна транспортна інфраструктура технологій.

Світова практика показує, що впровадження ІТС дозволяє знизити кількість дорожньо-транспортних пригод до 50%, збільшити пропускну здатність доріг на 25-30%, знизити витрату пального на 20%, витрати часу в дорозі на 30%, підвищити зайнятість населення на 5%. Наприклад, в США, які зараз значно відстали в цьому плані від Японії та Південної Кореї, системи управління автострадами збільшують швидкості від 13 до 48% проти раніше наявного режиму перевантаження. У Південній Кореї загальна економія від реалізації ІТС-технологій оцінюється в \$ 1,5 млрд на рік. Протягом наступних 20 років очікується створення промислових потужностей на 20 мільярдів доларів і зменшення економічних втрат через затори на дорогах на 26 млрд доларів.

З середини 80-х років ХХ століття в Японії, США та Європі починаються великомасштабні роботи по створенню і розвитку ІТС, або систем транспортної телематики схема 1.2.1, схема. 1.2.2. Незабаром на ринці з'явилися супутникові навігаційні системи GPS (США), GLONASS (Росія) і європейська система Galileo, що знаходиться в завершальній стадії введення її в роботу. [8].



Схема 1.2.1. Місце інформаційно-комунікаційних технологій у розвитку засобів транспорту

Транспортна телематика інтенсивно розвивається для всіх видів транспорту: наземного, авіаційного, залізничного, водного. Основні функції ІТС зображені на схемі 1.2.3. Особливо велика роль ІТС в розв'язанні задач інтермодальності. Однак найбільш комплексні та масштабні дослідження в галузі транспортної телематики проводяться для наземного транспорту. Прикладні напрямки європейських програм розвитку телематики наведені на схемі 1.2.2. У сучасному світі просування ІТС вже не є проблемою тільки окремо взятої країни. Майбутнє національних економік спирається на глобальні схеми транспортування та інтеграцію транспортних послуг у світовий ринок за допомогою механізмів Світової організації торгівлі (СОТ). З цією метою з 1993 року ведуться роботи з міжнародної стандартизації технологій ІТС. У 2008 році Комісія Європейських Спілок прийняла План дій

прискороного розгортання інтелектуальних транспортних систем в Європі та координації ІТС на автомобільному транспорті з іншими видами транспорту. У своїй доповіді Комісія констатувала, що в цій сфері послуги зараз розгорнуті на фрагментарній основі. Це призвело до клаптиків ковдри з національних, регіональних і місцевих рішень без чіткого узгодження, ставлячи під загрозу цілісність єдиного ринка.

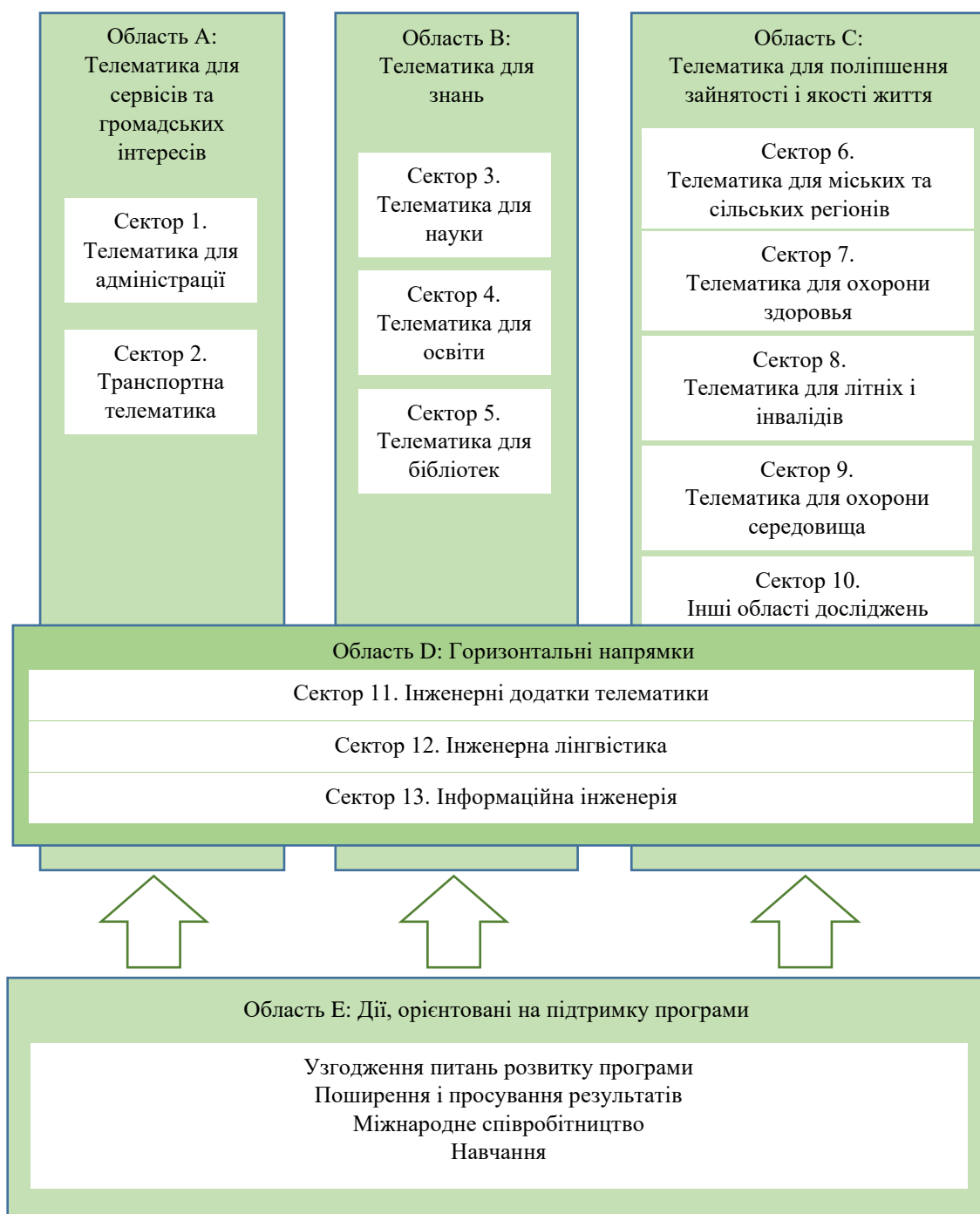


Схема 1.2.2. Прикладні європейські програми в області телематики

У світі існує тільки одна всеосяжна архітектура ІТС, запропонована транспортним департаментом США. Створення єдиної архітектури ІТС дозволяє контролювати три основних напрямки[8]:

1. Безпека (основна мета - зниження аварійності, моніторинг природних і техногенних катаклізмів).
2. Мобільність (збір інформації про пробки від рухомих в потоці автомобілів, інформування учасників руху та ін.).
3. Захист навколишнього середовища (зниження шкоди навколишньому середовищу від транспорту за допомогою моніторингу ситуації в реальному часі і своєчасного прийняття рішень).



Схема 1.2.3. Функції ІТС

РОЗДІЛ 2. ІНФОРМАЦІЙНА МОДЕЛЬ ПРЕДМЕТНОЇ ГАЛУЗІ

2.1. МОДЕЛЬ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для успішного керування проєктами розробки програмного забезпечення потрібно дотримуватися двох принципів. Перший це строгість – для постійного контролю стану проєкту. Другий – гнучкість – потрібно для успішної адаптації до неминучих несподіванок. В цьому розділі висвітлюється модель процесу розробки MSF. Модель MSF являє собою структурний каркас, на базі якого може створюватися конкретний спосіб вирішення завдань. Модель процесу розробки – є складовою частиною всієї структури, де описується життєвий цикл проєкту розробки програмного забезпечення. Вона дозволяє проєктній групі під час створення продукту підтримувати зв'язок із замовником для адаптування процесу розробки відповідно до його побажань. Крім цього, завдяки методу можливо забезпечити прискорення реалізації головних складових проєкту. Модель процесу розробки додатків являє собою гнучкий компонент загальної моделі процесу MSF, який успішно застосовують в індустрії розробки програмного забезпечення для кращої керованості проєктів, мінімізації ризиків, підвищення якості продукції й прискорення розробки[9].

Будь-який проєкт розробки програмного забезпечення під час свого розвитку має певний життєвий цикл, що можливо уявити в вигляді послідовність етапів і сукупність дій, які призводять до створення першої версії продукту. Побудова реалістичної моделі життєвого циклу, де буде описуватися на деякому рівні абстракції складові проєкту та порядок їх визначення, реалізації, тестування та виконання, спрощує виконання проєкту й гарантує, що в проєкті з кожним наступним етапом реалізується усе більше запланованих завдань.[9]. Наразі всі методи розробки додатків являють із себе

квінтесенцію досвіду та практики, що були напрацьовані при використанні традиційних моделей, як модель водоспаду й спіральна модель.

При розробці проєкту за основу буде взятий універсальна модель розробки програмного забезпечення. Універсальний процес (Unified Process, UP)[9] – широко розповсюджений метод аналізу, проєктування й розробки додатків масштабу підприємства. Основними характеристиками цього процесу є:

1. сценарій використання є основою проєкту;
2. пріоритет архітектури;
3. ітераційний і інкрементний процес розробки;
4. ретельне прогнозування ризиків проєкту;
5. об'єктно- і сервісно-орієнтоване проєктування;
6. активне використання об'єктно-орієнтованих шаблонів і об'єктів.

Основу універсального процесу складають п'ять основних етапів, що постійно виконуються на чотирьох фазах процесу розробки до створення додатка. Кожне завершення цих етапів є ітерацією, і кожна з ітерацій завершується проміжним створення прототипу продукту до його повної готовності. Назви етапів є умовними. На кожній ітерації цикл розпочинається зі збору вимог.

1. Вимоги – це збір технічних, прикладних і бізнес-вимог до проєкту.
2. Аналіз – бізнес- і прикладне моделювання на основі зібраних вимог.
3. Проєктування – створення архітектури з використанням об'єктно-орієнтованого підходу.
4. Реалізація – це розробка спроєктованого додатка (на ранніх стадіях – розробка прототипів).
5. Тестування – перевірка зробленої роботи.

Так як універсальний процес базується на основі спіральної моделі, він має такі ж самі фази – дослідження, пророблення, створення й перехідний період. Кожна з яких переслідує свої цілі [9]:

1. в фазі «Дослідження» відбувається створення моделі предметної області;
2. під час ітерації фази «Пророблення» створюється базова архітектура всього проєкту;
3. фаза «Створення» має на меті, з кожною ітерацією послідовного випуску версій з функціональними можливостями, що поступово розширюються, створення продукту;
4. перехідний період в який відбуваються останні тестування готовності продукту до експлуатації.

На першій і другій фазі основна увага приділяється збору вимог, їх аналізу та проєктуванням і реалізацією архітектури згідно до них. По завершенні фаз додаток із все зростаючою деталізацією описується за допомогою сукупності моделей універсального процесу. Перехід до кожної наступної фази відбувається лише при виконанні завдань поточної фази. А при закінченні кожної фази відбувається аналіз результатів її виконання та вирішується питання з продовженням роботи над проєктом та схвалення бюджету з графіком наступної фази у випадку продовження праці над проєктом. Завершальні етапи кожної фази є точкою синхронізації технічної та ділової сторони проєкту.

2.2. ЗБІР ВИМОГ

1. Інтуїтивно-зрозумілий інтерфейс та дизайн. У нового користувача не повинні виникати проблеми з тим як користуватися додатком.
2. Швидкість розрахунку маршрутів. Після сформування запиту з боку користувача, вхідні дані повинні бути швидко опрацьовані, та на основі них розрахований маршрут.
3. Актуальність наданої інформації. При розрахунку маршрутів важливу роль займає наскільки данні про чинний розклад руху залізничного

транспорту є актуальним, бо будь-які зміни, інформація про які не була оновлена можуть зробити розроблений маршрут невірним або неактуальним.

4. Широкий вибір маршрутів. При розрахунку маршруту та його видачі повинен підбиратися не єдиний варіант найкоротшого маршруту, бо розрахований маршрут може не підходити користувачу.

5. Збереження результату пошуку. Додаток повинен дозволяти користувачу зберігати знайдений маршрут.

6. Підтримка попередніх версій android.

7. Підтримка користувача. У кожного користувача повинна бути можливість звернення до підтримки в випадку знаходження або виникнення якоїсь помилки.

8. Коректність роботи алгоритму розрахунку маршруту. Алгоритм пошуку маршруту повинен видавати правильний коректний розроблений маршрут.

9. Оптимізація додатку. При роботі з додатком у користувача не повинні виникати проблеми з роботою додатку.

10. Відповідність стандартам розробки програмного забезпечення.

2.3. АНАЛІЗ

Згідно з даним проєктом ми маємо 3 головні завдання:

1. Отримання актуальної інформації про рух пасажирського та приміського залізничного транспорту.

2. Вибір алгоритму для розрахунку маршрутів та в випадку необхідності його модернізація під заданий проєкт.

3. Розробка самого android додатку.

При детальному розгляді першого завдання стало відомо, що Укрзалізниця не має публічного API для доступу до бази даних, а наявні компанії, які надають API, обмежені лише інформацією про наявність квитків пасажирського залізничного транспорту та не є надійними постачальниками

доступу до API. У зв'язку з цим для отримання інформації треба застосовувати Web scraping та синтаксичний аналіз (парсинг). Через те що інформація про пасажирський залізничний транспорт є залежною від наявності квитків та знаходиться на окремому сайті треба написати два різних скрипта для отримання інформації рис 2.3.1, рис. 2.3.2, при чому у випадку з інформацією про наявність квитків пасажирського транспорту треба організувати обхід захисту від ботів рис 2.3.3.

The screenshot displays the website of Ukrzaliznytsia (Ukrainian Railways), specifically the section for regional train schedules. The header features the company logo and navigation links. The main content area includes a search form for regional train schedules, with options to select the direction, date, and view type. Below the search form, there are sections for regional train schedules by direction, between stations, and by station. The interface is clean and user-friendly, with clear labels and input fields.

УКРЗАЛІЗНИЦЯ
 «ПІВДЕННО-ЗАХІДНА ЗАЛІЗНИЦЯ»

Про нас ▾ НОВИНИ та ІНТЕРВ'Ю Перевезення вантажів ▾ Послуги ▾ РОЗКЛАД РУХУ ▾ Schedule Працевлаштування

• Головна ▸ РОЗКЛАД РУХУ ▸ РОЗКЛАД РУХУ Приміських поїздів

Пошукові форми

Розклад на сайті повинен щоденно поновлюватись відповідальними працівниками по кожній регіональній дирекції. Поки що в розкладі можлива відсутність відображення невеликих тимчасових змін, або випадкові розбіжності. Для надійності погоджуйте розклад з поточною інформацією на станціях!

До уваги! ▸ Розклад руху міської електрички залізничним кільцем (Київ)

До уваги! ▸ Розклад руху **Kyiv Boryspil Express**

Розклад руху приміських поїздів за напрямком

Укрзалізниця (Україна)

Виберіть напрямок ▾ Обидва рейси ▾ Показати

На дату 2019-12-08 Всі дні Скорочений вигляд

Перелік напрямків може бути неповним. Дивіться також розклад між станціями

Розклад руху приміських поїздів між станціями

Відправлення: Прибуття: ↻ Показати

На дату 2019-12-08 Всі дні

Розклад руху по станції

Назва станції: Показати

Алфавітний показник станцій та платформ за географією

До уваги! ▸ Інтерактивна карта-схема Південно-Західної залізниці

довідка: Тарифи | Карта Південно-Західної залізниці [інформація]

© 2005-2019 ПАОУ ВП КВ

E!Train v3.0

запрошуємо на роботу

3 224 514
 9 455
 +8 765

Рис. 2.3.1. Інтерфейс сайту Укрзалізниці з розкладом руху приміського залізничного транспорту

Технічна підтримка: +380 (44) 591-1988 Skype: uz_booking Email: booking@uz.gov.ua UA

Довідкова служба: +380 (44) 309-7005 Контакт центр "Якість та сервіс": 0 (800) 503-111

Замовлення квитків Договір оферти Приклади бланків Порядок надання послуг Перевірка ЕПД Повернення квитків

Про УЗ Пасажирам Вантажні перевезення Прес-центр Контакти Авторизуватись

Подорожуй популярними напрямками

Київ ↔ Львів Київ ↔ Харків Київ ↔ Дніпро-Головний Київ ↔ Вінниця Київ ↔ Одеса

Звідки Куди

Київ Львів Харків Одеса Дніпро-Головний Київ Львів Харків Одеса Дніпро-Головний

Дата відправлення

08.12.2019 Час відправлення від 00:00

Сьогодні Завтра Післязавтра

Пошук поїздів на 08.12.2019

Рис 2.3.2. Інтерфейс сайту Укрзалізниці з продажу квитків з розкладом руху пасажирського залізничного транспорту

Подорожуй популярними напрямками

Київ ↔ Львів Київ ↔ Харків Київ ↔ Дніпро-Головний Київ ↔ Вінниця Київ ↔ Одеса

Звідки Куди

Харків Одеса

Київ Львів Харків Одеса Дніпро-Головний Київ Львів Харків Одеса Дніпро-Головний

Дата відправлення

08.12.2019

Сьогодні Завтра Післязавтра

Перевірочний код

Введіть цифри з картинки

4835

Відправити

Харків Одеса

Пошук прямих поїздів

Рис 2.3.3. Приклад дії захисту сайту Укрзалізниці з продажу квитків

Для розрахунку маршруту був обраний алгоритм пошуку по ширині через те, що він є одним зі швидких алгоритмів пошуку шляху. Також він потребує деякої модернізації для отримання декількох варіантів маршруту та створення матриці суміжності залізничних станцій.

При розробці android додатку було обране інтегроване середовище розробки для платформи Android – Android Studio.

2.4. ПРОЄКТУВАННЯ

Проектування — це процес створення проекту, прототипу або прообразу майбутнього об'єкта, з визначенням стану та способів його реалізації. Проектування відбувається з застосуванням системного підходу, що полягає у визначенні та створенні структури системи, типу зв'язків, визначених атрибутів, аналізуванні впливів зовнішнього середовища.

Додаток:

1. Розробка інтерфейсу з використанням можливостей IDE Android Studio.
2. Організація взаємодії з серверною частиною.
3. Тестування розробленого додатка та його оптимізація.

Серверна частина:

1. Розробка бази даних з використанням СКБД MySQL.
2. Розробка Web scraping і його реалізація мовою програмування Python для отримання інформації про розклад приміського транспорту.
3. Розробка Web scraping та використання машинного навчання реалізованого мовою програмування Python для отримання інформації про розклад пасажирського транспорту.
4. Використання хостингу з характеристиками, необхідними для стабільного функціонування додатку.

2.5. РИЗИКИ ПРОЄКТУ

Ризик [10] – це міра непевності в досягненні поставленою системою заданої мети при використанні обраних способів досягнення цієї мети. Метою аналізу проєктних ризиків є надання оцінки всім видам ризиків проєкту, а також знаходження:

1. можливих шляхів до зниження ризиків;
2. ступеня доцільності від реалізації проєкту за наявного рівня ризику та способів його зниження.

Можливі ризики проєкту:

1. Блокування доступу до сайту Укрзалізниці через велику кількість запитів. Вирішенням є постійна зміна параметрів User-Agent та використання проксі-серверів.
2. Оновлення структури сайту. Вирішенням є постійна підтримка проєкту та оперативна зміна коду скриптів web scraping.
3. Зміна captcha. Розв'язанням цієї проблеми є проведення заново машинного навчання згідно з новими вимогами.
4. Відсутність доступу до сайту Укрзалізниці. Через те, що пасажирський залізничний транспорт залежить від наявності квитків, то в випадку відсутності доступу до сайту буде використовуватися пошук маршрутів тільки приміським залізничним транспортом.
5. Проблеми в роботі додатку або алгоритму. Зменшення ймовірності виникнення ризику досягається ретельними тестуваннями та постійним покращенням коду.

РОЗДІЛ 3. ВИБІР МЕТОДІВ ТА РЕАЛІЗАЦІЇ ПОСТАВЛЕНИХ ЗАДАЧ

3.1. WEB SCRAPING

Веб-скрапінг (веб-збирання або витягнення веб-даних)[**Error! Reference source not found.**] — перетворення у структуровані дані інформації з вебсторінок, які призначені для перегляду людиною за допомогою браузера.

Як правило, реалізується за допомогою комп'ютерної програми, що імітує поведінку людини в інтернеті, або з'єднуючись з веб-сервером з використанням протоколу HTTP, або керуючи повноцінним веб-браузером. Також буває і скрапінг за допомогою копіювання даних людиною, яка збирає та копіює конкретні дані з інтернету в базу даних або електронну таблицю для їх подальшого використання.

Веб-скрапінг охоплює завантаження та вилучення. Спочатку завантажується сторінка, після цього здійснюється добування потрібної інформації. При цьому зміст сторінки може аналізуватися, переведені в інший формат та скопійований в електронну таблицю тощо. Веб-скрапери беруть щось зі сторінки, щоб використати це для інших цілей. Прикладом може бути контактне сканування, що полягає в пошуку і копіюванні імен та телефонних номерів або назв компаній та їх URL-адрес до бази.

Кожна з вебсторінок побудована за допомогою текстових мов розмітки (HTML або XHTML) та часто містить корисні дані у текстовій формі. Але більшість вебсторінок призначені лише для кінцевого користувача, а не для зручного автоматичного використання. Через це й виникла необхідність у створенні набору інструментів, що «збирають» веб-вміст. Тому веб-скрапери є прикладним програмним інтерфейсом для вилучення даних з веб-сайтів.[12]

Так як при використанні веб скрапінгу відбувається навантаження на веб-сайт, виникли методи для його запобігання. Наприклад, пошук та

блокування ботів, що виконували сканування (перегляд) сторінок веб-сайту. У відповідь на це є деякі веб-скрапінгові системи, в яких використовують методи аналізу об'єктної моделі документа, комп'ютерного зору та обробки тексту за допомоги природної мови, для імітації пошуку людиною, щоб дозволити збирати вміст веб-сторінок для подальшого її синтаксичного аналізу.

Наразі питання легальність веб-скрапінгу у світі різниться. Так є веб-сайти що забороняють скрапінг у правилах використання, але з юридичної токи зору наслідки від заборони не є чіткими.

Веб-скрапінг буде проводитися за допомогою мови програмування Python, при використанні фреймворку Scrapy та бібліотеки SQLAlchemy. Scrapy це швидкий веб-фреймворк з відкритим кодом, написаний на Python, який використовується для отримання даних з вебсторінки за допомогою селекторів на основі XPath. Основними перевагами цього фреймворку є[13]:

- Вбудована підтримка для пошуку та вилучення даних з HTML/XML сторінок, з використанням розширених селекторів CSS і виразів XPath та можливістю використання допоміжних методів, які використовують регулярні вирази.
- Інтерактивна консоль оболонки, що дозволяє краще писати та налагоджувати павука.
- Вбудована підтримка генерації файлів у форматах JSON, CSV, XML для збереження результатів синтаксичного аналізу сторінок задля їх подальшого використання.
- Широкий спектр вбудованих розширень для обробки даних та можливість їх подальшої модифікації.
- Наявність автоматичного регулювання швидкості сканування з використанням механізму автоматичного дроселювання.
- Обробка запитів виконується асинхронно та є запланованою.

Бібліотека SQLAlchemy[14] – це об'єктно-реляційне відображення інструментарію SQL для мови програмування Python випущене під ліцензією MIT. Вона діє на основі шаблону Data mapper, що забезпечує двосторонню

конвертацію даних з об'єкта (в оперативній пам'яті) до бази даних та навпаки з бази даних знов у об'єкт. Такий підхід дозволяє виконувати зміну структури бази даних або об'єкта в певних межах, змінюючи при цьому лише файл взаємодії з базою даних.

Алгоритм роботи веб-скрапінгу для отримання даних про розклад руху приміського залізничного транспорту:

1. Отримання HTML-сторінок для аналізу.
 - a. Підключення до початкової сторінки, з якої буде починатися веб-скрапінг (рис 3.1.1).
 - b. Отримання масиву значень, які відповідають наявним напрямкам руху приміських поїздів.
 - c. Виконання запитів до сервера сайту, для отримання розкладу (рис. 3.1.2).
 - d. Завантаження отриманої сторінки, та перехід до наступної.
2. Синтаксичний аналіз HTML-сторінки.
 - a. Відкриття отриманої сторінки з розкладом.
 - b. Використання попередньо підготовлених виразів XPath для знаходження інформації.
 - c. Збереження упорядкованої отриманої інформації в форматі CSV
3. Внесення отриманої інформації в базу даних.
 - a. Підключення до бази даних MySQL.
 - b. Зчитування даних з CSV.
 - c. Внесення даних до бази даних.
4. Закінчення роботи веб-скрапінгу.

The screenshot shows the website of the Ukrainian Railways (UZ) Southern-Western Railway branch. The header includes the logo and navigation menu. The main content area features a search form for regional train schedules. The form includes a dropdown for the railway branch (set to 'Україна'), a 'Виберіть напрямок' (Select direction) dropdown, and a 'Показати' (Show) button. Below the form, there are radio buttons for 'На дату' (On date) and 'Всі дні' (All days), with the date '2019-12-08' selected. A 'Скорочений вигляд' (Short view) checkbox is also present. A warning message in red text states: 'До уваги! Розклад руху міської електрички залізничним кільцем (Київ)' and 'До уваги! Розклад руху Kyiv Boryspil Express'. The form is divided into two sections: 'Розклад руху приміських поїздів за напрямком' and 'Розклад руху приміських поїздів між станціями', each with its own search fields and options.

Рис. 3.1.1. Сторінка сайту філії Укрзалізниці з розкладом руху приміського залізничного транспорту.

Алгоритм роботи веб-скрапінгу для отримання даних про рух пасажирського залізничного транспорту має іншу структуру через те, що залежить від наявності квитків:

1. Отримання запиту від android-додатка з даними для пошуку інформації про рух пасажирського залізничного транспорту.
2. Підключення до сторінки з продажу квитків Укрзалізниці, для проведення веб-скрапінгу.
3. Виконання запиту до сайту з метою отримання інформації про наявність потягу по певному маршруту. У випадку появи captcha, виконується завантаження зображення та надсилання його до скрипта з обходу captcha, отримання результату та внесення його в необхідне поле (або оновлення captcha в випадку, якщо не вдалося її розпізнати).
4. Отримання сторінки з результатами пошуку та її завантаження.
5. Синтаксичний аналіз сторінки та відправлення отриманого результату до додатка

Розклад руху приміських поїздів за напрямком

Укрзалізниця (Україна)

Івано-Франківськ - Моршин Туди Показати

На дату Всі дні Скорочений вигляд

Перелік напрямків може бути неповним. Дивіться також розклад між станціями

Івано-Франківськ - Моршин			Івано-Франківськ – Болахів	
Пункт / Номер та обіг	Σ км		приб.	відпр.
0	Івано-Франківськ	0	6425,	щоденно *
1	з.п. Угринів	3.4	–	20:20
2	з.п. Павлівка	8.6	20:26	20:27
3	Ценжів	16.5	20:35	20:36
4	з.п. Майдан	22.5	20:50	20:52
5	Боднарів	28.3	21:01	21:02
6	з.п. Вістова	33.2	21:10	21:12
7	Калуш	42.3	21:20	21:21
8	з.п. Голинь	50.3	21:32	21:34
9	з.п. Брошнів	53.8	21:43	21:44
10	Рожнятів	56.6	21:49	21:50
11	з.п. Раків	60.8	21:56	21:58
12	з.п. Надіїв	63.5	22:05	22:06
13	з.п. Рахія	66.5	22:12	22:13
14	Долина	70.9	22:21	22:22
15	Яворів	75.3	22:31	22:33
16	з.п. Свіча	78.5	22:41	22:42
17	Болахів	82.7	22:47	22:48
18	з.п. Лісовичі	89.5	22:53	–
19	Моршин	92.7	–	–

Рис. 3.1.2. Приклад розкладу отриманого в результаті запитів.

Звідки Куди

Київ Львів Харків Одеса Дніпро-Головний Київ Львів Харків Одеса Дніпро-Головний

Дата відправлення Час відправлення від

08.12.2019 00:00

Сьогодні Завтра Післязавтра

Пошук поїздів на 08.12.2019

Рис. 3.1.3. Сторінка сайту Укрзалізниці для покупки квитків.

3.2. МАШИННЕ НАВЧАННЯ

Машинне навчання (МО, Machine Learning, ML)[15] - це великий підрозділ штучного інтелекту, в якому вивчаються методи побудови алгоритмів, які були б здатні до навчання; це дослідження методів, за допомогою яких комп'ютер може покращувати свої характеристики завдяки отриманому досвіду.

В 1952 році Артур Самуель розробив першу програму на основі алгоритмів, здатних самонавчатися, яка була призначена для гри в шашки. Артур Самуель також дав перше визначення терміну «машинне навчання», в якому він розглядався як область дослідження розробки машин, які не треба заздалегідь запрограмувати. Більш точне визначення терміну «навчання» дав набагато пізніше Т. М. Мітчелл[14]: «кажуть, що комп'ютерна програма навчається на основі досвіду E стосовно деякого класу задач T і заходи якості P , якщо якість вирішення завдань з T , вимірний на основі P , поліпшується з набуттям досвіду E ». Наразі відбувається розробка різноманітних систем машинного навчання, призначених для використання в таких технологіях, як Інтернет Речі, при створенні безпілотного транспорту, Промислові Інтернет Речі, в концепції «розумне» місто, та в багатьох інших.

Розрізняють два типи машинного навчання: індуктивне навчання, або навчання по прецедентах та дедуктивне навчання. Своєю чергою індуктивне навчання, ділять на три основних типи: контрольоване навчання (навчання з учителем), неконтрольоване навчання (навчання без учителя), та навчання з підкріпленням, тобто навчання що використовує деякі принципи та методи як з контрольованого так і неконтрольованого типу навчання.

Для машинного навчання застосовуються різні технології та алгоритми. Наприклад, може використовуватися дискримінантний аналіз та байєсовські класифікатори й інші математичні методи. Але з кінця ХХ століття більше уваги почали приділяти реалізації за допомогою штучних нейронних мереж (ANN).

ANN є системою з'єднаних і скоординованих між собою штучних нейронів, які виконані на основі порівняно простих процесорів. Так кожен процесор ANN періодично отримує сигнали від одних процесорів, сенсорів, або від інших джерел сигналів та періодично посилає сигнали до інших процесорів. Всі ці прості процесори, з'єднані в мережу, здатні знаходити рішення з досить складних завдань. [16]

Реалізація машинного навчання для обходу captcha (рис. 3.2.1) буде складатися з двох частин: підготовка даних до аналізу та самої програми розпізнавання цифр.



Рис 3.2.1. Приклад captcha сайту Укрзалізниці з продажу квитків з розкладом руху пасажирського залізничного транспорту

Етап підготовки даних необхідний для того щоб привести початкове зображення у вигляд зручний для його аналізу. Для цього буде використовуватися витяг кольору за допомогою алгоритму багатозначного розкладання зображення. Тобто ми спочатку будемо гістограму кольорів зображення рис 3.2.1 шляхом отримання всіх пікселів в зображення з угрупованням за кольором, після чого проводиться підрахунок по кожній групі (рис 3.2.2). Проаналізував отриманні данні ми отримаємо, в даному випадку 4 групи кольорів які зустрічаються найчастіше, при чому той, що зустрічається найбільше разів є фоновим.

```
from PIL import Image
from operator import itemgetter

im = Image.open("captcha.gif")
im = im.convert("P")
his = im.histogram()

values = {}

for i in range(256):
    values[i] = his[i]

for j,k in sorted(values.items(), key=itemgetter(1), reverse=True)[:10]:
    print j,k
```

Рис 3.2.2. Приклад коду на Python для отримання гістограми кольорів

Після того як було отримано цю інформацію, створюється нове зображення, засноване на цих колірних групах. Для кожного з найбільш поширених кольорів ми створюємо нове бінарне зображення (з 2 кольорів), де пікселі цих кольорів заповнюється чорним, а все інше білим. Тобто ми створюємо нове зображення такого ж розміру з білим фоном, а потім обходимо початкове зображення в пошуках потрібного нам кольору. Якщо він знаходить піксель з потрібним нам кольором, то він зазначає цей же піксель на другому зображенні чорним кольором. Перед завершенням роботи друге зображення зберігається рис 3.2.3.

```
from PIL import Image

im = Image.open("captcha.gif")
im = im.convert("P")
im2 = Image.new("P",im.size,255)

im = im.convert("P")

temp = {}

for x in range(im.size[1]):
    for y in range(im.size[0]):
        pix = im.getpixel((y,x))
        temp[pix] = pix
        if pix == 220 or pix == 227: # these are the numbers to get
            im2.putpixel((y,x),0)

im2.save("output.gif")
```

Рис 3.2.3. Приклад коду на Python для отримання бінарного зображення

Наступним кроком для полегшення та прискорення завдання аналізу captcha виконується розділ зображення на n-кількість фрагментів відповідно до кількості символів в captcha. Розподіл на фрагменти виконується за допомогою знаходження контурів символів. Отримані фрагменти зберігаються окремими зображеннями.

Після отримання окремих зображень символів captcha готова до аналізу. Але для того щоб наша нейронна мережа мала можливість для аналізу нам потрібно спочатку її навчити. Для цього потрібно створити колекцію з різних варіантів зображень символів які зустрічаються у captcha. І чим більша кількість навчальних даних тим вище ймовірність правильного результату. Після проведення навчання, нейронна мережа буде готова до приймання вхідних даних, їх аналізу та повернення відповіді.

3.3. БАЗА ДАНИХ

Для зберігання даних, а саме розкладу руху приміського залізничного транспорту, отриманих при веб-скрапінгу на стороні серверу було прийнято рішення використати систему керування базами даних (СКБД) MySQL. При зберіганні даних розкладу в android додатку буде використовуватися СКБД SQLite.

MySQL – є вільною системою керування реляційними базами даних. Ця СКБД була створена як альтернатива комерційним системам тому вона має відкритий код. Зараз MySQL є одною з найпоширеніших систем керування БД та використовується, в першу чергу, для створення динамічних вебсторінок, завдяки великій підтримці з боку різноманітних мов програмування. Вона має багатий набір функціональних можливостей, що забезпечують безпечне середовище для отримання, обслуговування та зберігання даних.

Можливості сервера MySQL [17]:

- 1) простота у розгортанні та під час використання;
- 2) підтримка необмеженої кількості користувачів для одночасної роботи з базою даних;
- 3) кількість записів у таблицях може сягати 50 млн;
- 4) швидке виконання команд;
- 5) вбудована проста й ефективна система безпеки.

SQLite [18] – це система керування базами даних, яка легко вбудовується в додатки. На відміну від більшості інших баз даних SQL, вона не має окремого процесу серверу, тобто SQLite виконує операції зчитування та запису безпосередньо до звичайних файлів дисків. Вся база даних SQL з декількома таблицями, індексами, активаторами міститься в одному файлі диску, який є крос-платформним та може вільно копіюватися. Завдяки базуванню на файлах

вона має широкий набір інструментів в порівнянні з мережевими базами даних. Саме тому SQLite є дуже швидкою. В порівнянні з іншими СКБД в SQLite має всього 5 типів даних, які також називаються класами зберігання та є більш загальними ніж в інших базах даних:

- NULL – має значення NULL.
- INTEGER – знакове цілочисельне значення використовуючи 1, 2, 3, 4, 6 або 8 байтів в залежності від порядку числа.
- REAL – число з плаваючою крапкою, займає 8 байтів для зберігання числа в форматі IEEE.
- TEXT – текстовий рядок, при зберіганні використовуються кодування UTF-8, UTF-16BE або UTF-16LE.
- BLOB – тип даних BLOB, масив двійкових даних (призначений для зберігання зображень, аудіо та відео).

При розробці бази даних одним з головних етапів є створення інформаційної та даталогічної моделі предметної області, в цьому випадку розкладу руху приміського залізничного транспорту.[19]. При огляді предметної області можливо виділити 4 сутності:

1. Напрямок – Кожен потяг має свою початкову й кінцеву станцію.
2. Потяг – за одним і тим же маршрутом може пересуватися декілька різних потягів в різний час.
3. Маршрут – потяг прямує від початкової до кінцевої станції зупиняючись на якийсь час на станціях.
4. Станція – залізничний вузол через які їздять потяги, зупиняючись згідно з розкладом.

У кожній з зазначених сутностей є атрибути (таблиці 3.3.1 – 3.3.4), які є поіменовано логічно неподільними властивостями (характеристиками) сутності.

Таблиця 3.3.1. Список атрибутів сутності «Напря́м»

№ п.п.	Найменування атрибута	Короткий опис
1	Id	Ідентифікатор
2	Назва	Назва маршруту, яким рухається потяг
3	Начальна станція	Станція, з якої почав рухатися потяг
4	Кінцева станція	Станція, до якої буде рухатися потяг

Таблиця 3.3.2. Список атрибутів сутності «Потяг»

№ п.п.	Найменування атрибута	Короткий опис
1	Id	Ідентифікатор
2	Id_напря́м	Ідентифікатор відповідного напрямку яким рухається потяг
3	Id_номер	Кожен потяг який рухається певним маршрутом в певний час має свій ідентифікатор

Таблиця 3.3.3. Список атрибутів сутності «Маршрут»

№ п.п.	Найменування атрибута	Короткий опис
1	Id	Ідентифікатор
2	Id_станція	Ідентифікатор станції, на якій знаходиться потяг в цей момент часу
3	Час прибуття	Час прибуття потягу
4	Час відправлення	Час відправлення потягу

Таблиця 3.3.4. Список атрибутів сутності «Станція»

№ п.п.	Найменування атрибута	Короткий опис
1	Id	Ідентифікатор
2	Назва	Назва станції

Для кращого розуміння предметної області використовується модель «сутність-зв'язок» (ER-модель), що дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. Головна ідея даної моделі полягає в тому, що між двома сутностями в предметній області є деяка асоціація, тобто зв'язок, за допомогою якого стає можливим опис та визначення предметної області.[19] На схемі 3.3.1 зображені зв'язки, що існують між сутностями даної предметної області та як вони виражаються.

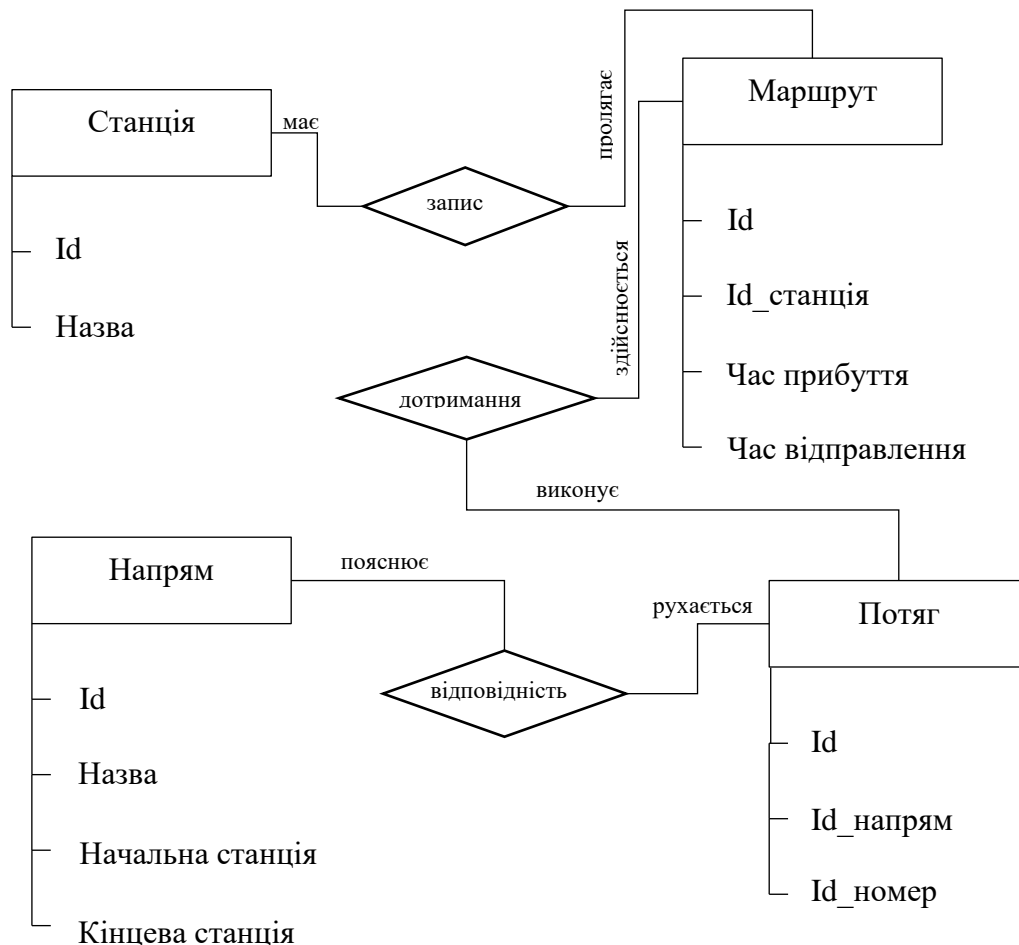


Схема 3.3.1. ER-діаграма зв'язків між сутностями предметної області

В той же час на схемі 3.3.2 зображена логічна структура зв'язків, з зображенням їх типу та модальності. Модальність зв'язків позначає чи є зв'язок між сутностями обов'язковим чи можливий варіант при якому один примірник першої сутності може бути не пов'язаний з примірниками другої сутності. Типи зв'язку бувають[19]:

- 1:Б - один примірник однієї сутності (лівої) пов'язаний з декількома екземплярами другої сутності (правої)
- 1:1 - один примірник першої сутності (лівої) пов'язаний з одним примірником другої сутності (правої)
- Б:Б - кожному екземпляру першої сутності може відповідати декілька екземплярів другої сутності та навпаки.

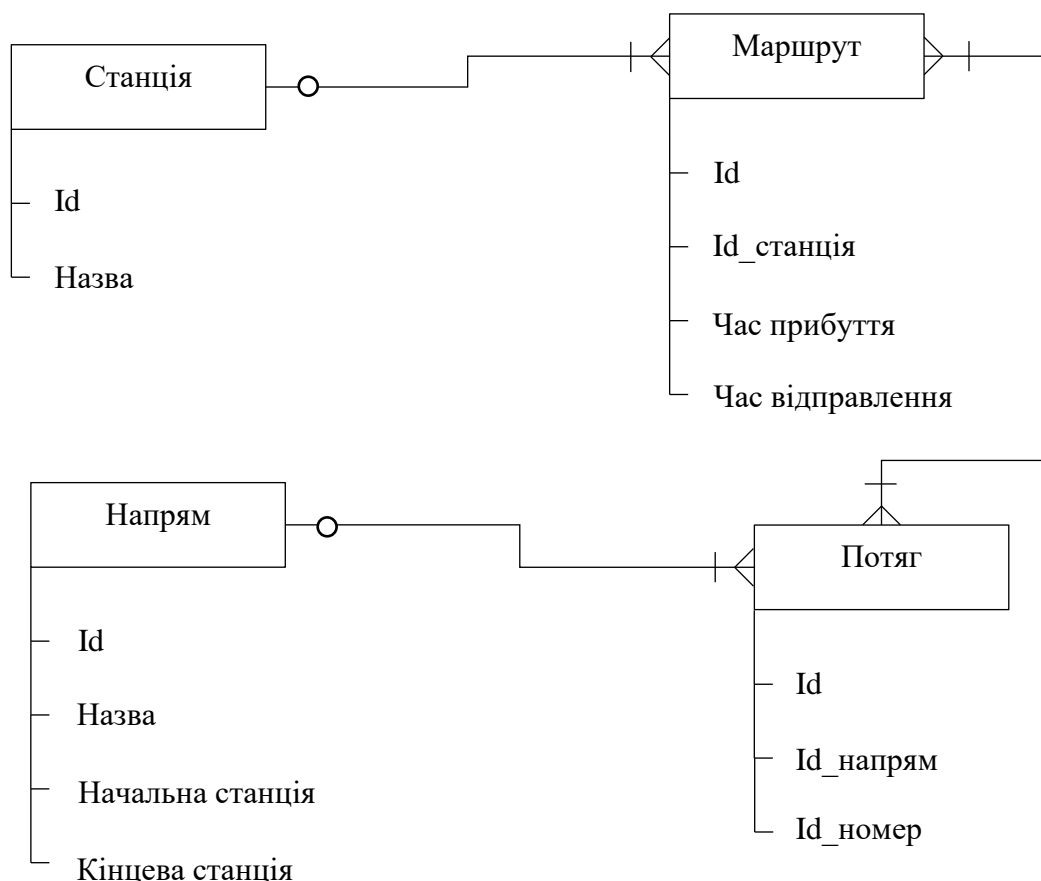


Схема 3.3.2. ER-діаграма логічної структури зв'язків між сутностями

Після інформаційної моделі, створюється даталогічна модель, що являє собою відображення логічного зв'язку між елементами даних безвідносно до їхнього

змісту та середовища зберігання. Вона будується в термінах інформаційних одиниць припустимих в СКБД. В таблицях 3.3.5 – 3.3.8 знаходиться опис даталогічних моделей сутностей предметної області відповідно до MySQL.

Таблиця 3.3.5. Даталогічна модель сутності «Напря́м»

№ п.п.	Найменування реквізиту	Ідентифікатор	Тип	Довжина	Обмеження та коментарі
1	Id	id	Int	10	UNSIGNED
2	Назва	title	Varchar	120	
3	Начальна станція	from_station	Varchar	50	
4	Кінцева станція	to_station	Varchar	50	

Таблиця 3.3.6. Даталогічна модель сутності «Потяг»

№ п.п.	Найменування реквізиту	Ідентифікатор	Тип	Довжина	Обмеження та коментарі
1	Id	id	Int	10	UNSIGNED
2	Id_напря́м	id_direction	Int	10	UNSIGNED
3	Id_номер	id_trainroute	Int	10	UNSIGNED

Таблиця 3.3.7. Даталогічна модель сутності «Маршру́т»

№ п.п.	Найменування реквізиту	Ідентифікатор	Тип	Довжина	Обмеження та коментарі
1	Id	id	Int	10	UNSIGNED
2	Id_станція	id_station	Int	10	UNSIGNED
3	Час прибуття	arr_time	Time		
4	Час відправлення	dep_time	Time		

Таблиця 3.3.8. Даталогічна модель сутності «Станція»

№ п.п.	Найменування реквізиту	Ідентифікатор	Тип	Довжина	Обмеження та коментарі
1	Id	id	Int	10	UNSIGNED
2	Назва	stantion	Varchar	50	

3.4. АЛГОРИТМ РОЗРАХУНКУ МАРШРУТУ

Пошук у ширину — це алгоритм пошуку наявності шляху від однієї вершини до іншої на графі.

Якщо задано довільний граф $G = (V, E)$ та початкову вершину s , алгоритм пошуку в ширину систематично перевіряє всі досяжні із s вершини (порядок обходу вершин зображено на рис. 3.4.1.). Перший крок полягає в тому, що вершина s позначається, як пройдена, а всі вершини, досяжні з s , при цьому без відвідування проміжних вершин, додаються до списку. Кожен наступний крок всі поточні вершини списку відмічаються, як пройдені та формується новий список із вершин, котрі були ще не пройденими сусідами поточних вершин списку. Під час реалізації списку вершин найчастіше застосовують чергу та принцип FIFO. Виконання алгоритму відбувається до знаходження шуканої вершини або до тої пори, коли після певного кроку в список не буде включена жодна з вершин. Другий випадок означає, що було перевірено всі вершини, які доступні з початкової, та відмічені, як пройдені, а шляху до шуканої вершини не знайдено. [20]

Алгоритм отримав назву пошук в ширину через те, що пошук (між пройденими та непройденими вершинами) одноманітно розширюється вздовж всієї своєї ширини. Тобто, спочатку проходить всі вершини на відстані k , а вже після їх обходу вершини на відстані $k+1$.

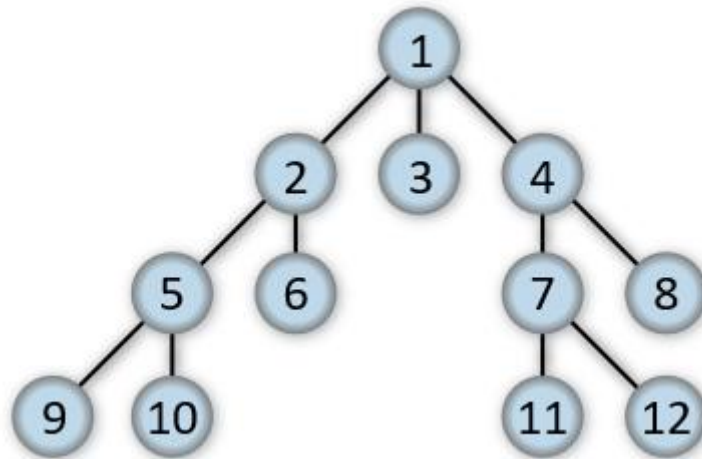


Рис. 3.4.1. Порядок обходу вершин

Кроки алгоритму пошуку по ширині[20]:

1. Розпочати з довільної вершини v та виконати $BFS(v):=1$. Включення вершини v у початок черги.
2. Розгляд вершини, що перебуває на початку черги; нехай вона буде позначатися як вершина x . Якщо всі вершини, суміжні із вершиною x , уже отримали BFS-номер, то перейти до кроку 4, якщо ні то до кроку 3.
3. Нехай $\{x, y\}$ – ребро, номер $BFS(y)$ якої не визначено. Позначити це ребро та визначити $BFS(y)$ як черговий BFS-номер, включити вершину y до кінця черги та перейти до кроку 2.
4. Виключити вершину x із черги. Якщо в черзі відсутні вершини, то зупинитись, інакше - перейти до кроку 2.

Якщо довжини ребер розглянутого графа рівні між собою то пошук в ширину є оптимальним, тобто завжди буде знайдено найкоротший шлях від однієї вершини до іншої. В разі зваженого графа пошук в ширину знаходить не обов'язково найкоротший шлях, а той що містить мінімальну кількість ребер.

На зваженому графі з невід'ємними вагами ребер використовують пошук за критерієм вартості, який є узагальненням пошуку в ширину. Він відвідує

кожен вузол графа згідно з зростанням вартості шляху від початкового вузла та зазвичай використовує чергу за пріоритетами.[20]

Кроки алгоритму пошуку маршруту пересування залізничним транспортом:

1. Перевірка наявності станції відправлення та прибуття і перевірка наявності розкладу за станціями, тобто чи зупиняються потяги на цій станції.
2. Позначаємо станцію відправлення як першу вершину графу та вносимо її в чергу.
3. Цикл пошуку - перевірка черги, якщо вона не порожня, то береться перша вершина з черги та відбувається перевірка всіх станцій з якими вона суміжна, з метою знаходження станції прибуття. В цьому випадку в ролі ребра будуть використовуватися потяги, які курсують між станціями. Якщо всі суміжні станції вже внесені в чергу, то вершина виключається з черги та відбувається перехід до наступної. Якщо станції, ще не були внесені, то вони вносяться в кінець черги. Пошук відбувається доки не буде знайдена станція прибуття або черга не стане порожня, що позначає неможливість побудування маршруту.
4. Виведення результату пошуку користувачу.

3.5. ANDROID STUDIO

Android Studio[21] — це інтегроване середовище розробки (IDE) для розробки додатків для платформи Android, презентація якої відбулась на конференції Google I/O 16 травня 2013 року, а вже 8 грудня 2014 року було випущено перший стабільний реліз IDE Android Studio. Android Studio, побудоване на базі вихідного коду продукту IntelliJ, прийшло на зміну плагіну ADT для платформи Eclipse, та представляла собою спеціалізоване середовище для розробки під операційну систему Android. Середовище розвивається відповідно до відкритої моделі розробки та поширюється згідно ліцензії Apache 2.0.

В середовищі є засоби для розробки додатків не тільки для смартфонів і планшетів, але й для носимих пристроїв на базі Android Wear, окулярів Google Glass, телевізорів (Android TV) та автомобільних систем (Android Auto).

Середовище розробки пристосоване до вирішення типових завдань, що виникають під час процесу розробки додатків для платформи Android. Так у середовищі вбудовані засоби для спрощення процесу тестування програм на сумісність з різними версіями платформи та інструменти для проєктування додатків, які працюють на пристроях з екранами різної роздільності. Також середовище має два редактора для створення інтерфейсу додатку. Перший редактор, використовуючи колекцію типових елементів інтерфейсу, надає можливість швидко створювати інтерфейс додатку та попередній перегляд різних станів елементів інтерфейсу. Другий редактор використовується для створення нестандартних інтерфейсів за допомогою власних елементів оформлення. У середовище також є функція завантаження прикладів коду з GitHub.

Деякі особливості передбачені в IDE Android Studio [21]:

- Особливий підхід до створення інтерфейсу додатку.
- Консоль розробника, що надає широкі можливості при розробці та підтримується з боку Google.
- Можливість планування та роботи з релізами.
- Базування структури додатку на Gradle.
- Розширені інструменти зміни внутрішньої структури програмного забезпечення, що були пристосовані до розробки на платформі Android.
- Утиліти для дослідження продуктивності, сумісності версій та інших проблем додатку.
- Використання можливостей оптимізатору коду ProGuard та підписів до програм.
- Наявність шаблонів різноманітних Android дизайнів та їх компонентів.

3.6. ОПИС ІНТЕРФЕЙСУ ДОДАТКУ

При розробці дизайну інтерфейсу додатку основним принципом була простота: простота рішення, навігації, інтуїтивність інтерфейсу. Макет додатку створювався за допомогою програми InVisio, що дозволяє виконати попереднє тестування навігації по додатку користувачем. А завдяки вбудованим можливостям в Android Studio інтерфейс було створено з використанням інструментів, що підтримуються з версії Android 5.0 Lollipop, котра згідно з статистичними даними, які надає компанія Google дозволяє охопити близько 94% всіх користувачів Android.[21]. Інтерфейс будується за допомогою мови розмітки XML.

На рис. 3.6.1 зображена головна сторінка додатку, за допомогою якої користувач може розпочати пошук необхідного йому маршруту, вказавши станцію відправлення і прибуття, та дату з якої треба розраховувати маршрут. Або відкрити меню для навігації по додатку (рис 3.6.2).

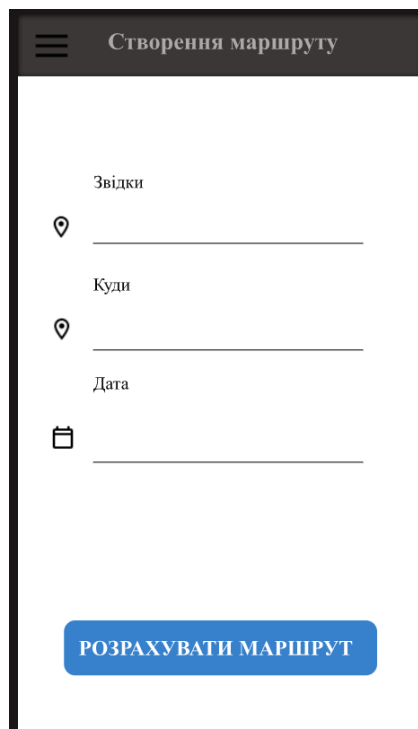


Рис 3.6.1. Інтерфейс пошуку маршрутів в android додатку

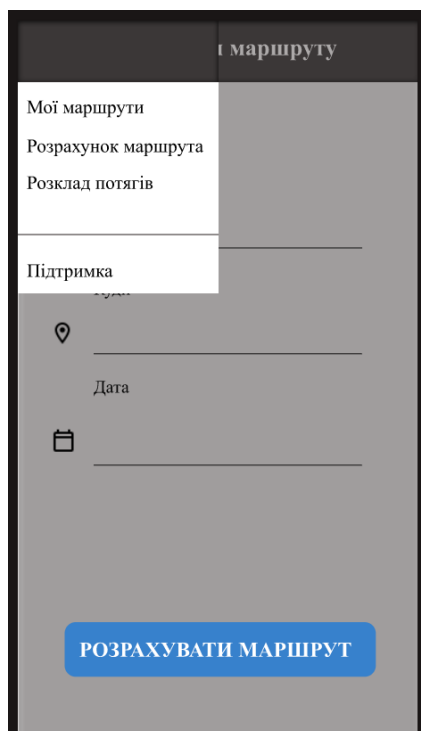


Рис 3.6.2. Інтерфейс меню android додатку

На рис. 3.6.3 зображений інтерфейс перегляду розкладу по станції, для цього необхідно користувачу вказати станцію, з розкладом якої він хоче ознайомитися, та дату.

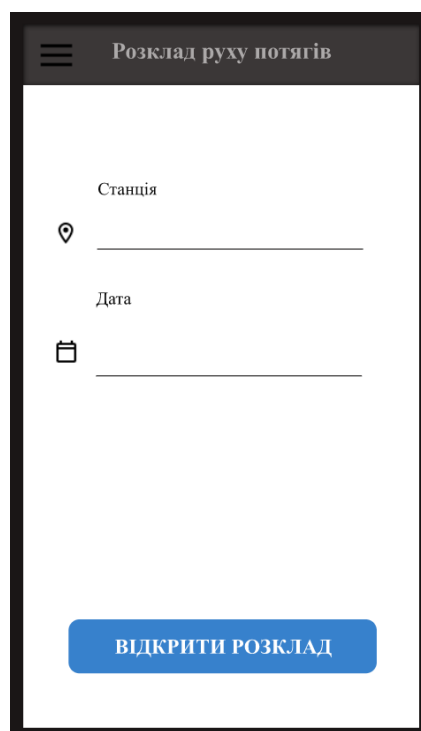


Рис 3.6.3. Інтерфейс розкладу руху потягів за станцією android додатку

На рис. 3.6.4 зображено інтерфейс керування збереженими маршрутами користувача. Користувач може виконувати перегляд своїх маршрутів та редагувати їх. В випадку якщо у користувача відсутні маршрути йому буде запропоновано створити маршрут.

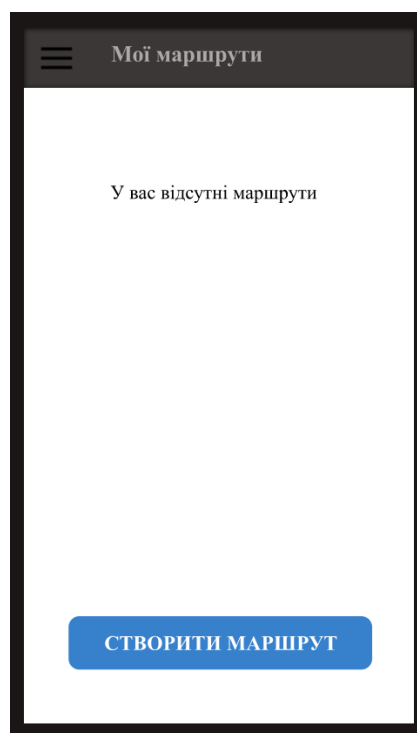


Рис 3.6.4. Інтерфейс перегляді наявних маршрутів користувача в android додатку

ВИСНОВКИ

1. Був розроблений та реалізований алгоритм з побудови додатку з побудови маршрутів пересування залізничним транспортом.
2. В результаті дослідження був проведений аналітичний огляд питання та побудована універсальна модель розробки програмного забезпечення.
3. Було сформовано підхід до отримання інформації для аналізу з використанням технологій web scraping та машинного навчання, реалізація була виконана мовою програмування Python
4. Створено прототип android-додатку, в якому за допомогою уявлення руху потягів між станціями в вигляді графу та використання алгоритму пошуку по ширині реалізовано розробку маршрутів пересування залізничним транспортом. Додаток було реалізовано в IDE Android Studio мовою програмування Java
5. Додаток може використовуватися користувачами для побудови маршруту пересування залізничним транспортом, що може допомагати дібратися з одного місця в інше.

СПИСОК ЛІТЕРАТУРИ

1. Логістика : [навч. посіб. для студ. вищ. навч. закл.] / Бержанір А. Л., Рибчак В. І., Слободяник Н. П. ; Уман. держ. аграр. ун-т. — Умань (Черкас. обл.) ; Уман. вид.-поліграф. п-во, 2009. — 347 с. : іл., табл. ; 21 см. — Присвяч. 165-річчю заснування ун-ту. — Бібліогр.: с. 344—347 (56 назв). — 200 пр. — ISBN 978-966-477-026-9
2. О.М. Сумець, О.Б. Білоцерківський, І.Л. Голофаєва, Логістика: теорія, ситуації, практичні завдання [Текст] : навч. посіб. для студ. вищих навч. закл. / за ред. канд. техн. наук О. М. Сумця. - Х. : Міськдрук, 2010 .
3. Л.М. Устименко, І.Ю. Афанасьєв. Історія туризму. Навчальний посібник. – К.: Альтерпрес, 2005.
4. RAILWAYS IN THE GREEK AND ROMAN WORLD [Електронний ресурс]. Доступно:
<https://web.archive.org/web/20110721083013/http://www.sciencenews.gr/docs/diolkos.pdf>
5. Berger, Michael L. (2001). The automobile in American history and culture: a reference guide. Greenwood Publishing Group. [ISBN 978-0-313-24558-9](#).
6. Історія повітроплавання та авіації: бібліогр. покажч. літ. / укладач М.Ф. Семенова. –2-ге вид., перероб. та доп. – К.: НАУ, 2008. – 153 с.
7. Збірник інтелектуальні транспортні системи (ІТС). [Електронний ресурс]. Доступно: <https://mobilitylviv.com/its-module-sutp-giz-ua/>
8. Інтелектуальні транспортні системи залізничного транспорту (основи інноваційних технологій) [Текст]: посібник / В. В. Скалозуб, В. П. Соловйов, І. В. Жуковицький, К. В. Гончаров. - Д.: Вид-во Дніпропетр. нац. ун-ту ж.-д. трансп. ім. акад. В. Лазаряна, 2013. - 207 с.
9. Моделі розробки додатків. . [Електронний ресурс]. Доступно: https://studopedia.ru/19_19046_modeli-rozrobki-dodatktiv.html
10. Оцінка ризиків проекту. [Електронний ресурс]. Доступно: <http://ukr.vipreshebnik.ru/upr-rizik/1220-otsinka-rizikiv-proektu.html>

11. Китаєв Е.Л, Скорнякова Р.Ю., StructScraper - Інструмент для динамічного включення в контент веб-сторінки семантичних даних зовнішніх веб-ресурсів // Науковий сервіс в мережі Інтернет. - 2019.
12. Boeing, G.; Waddell, P. New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings // Journal of Planning Education and Research. — 2016.
13. Scrapy 2.1 documentation. [Електронний ресурс]. Доступно: <https://docs.scrapy.org/en/latest/>
14. SQLAlchemy Documentation [Електронний ресурс]. Доступно: <https://docs.sqlalchemy.org/en/13/>
15. Machine Learning - Машинне навчання. [Електронний ресурс]. Доступно: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>
16. Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2
17. MySQL Documentation. [Електронний ресурс]. Доступно: <https://dev.mysql.com/doc/>
18. Documentation. [Електронний ресурс]. Доступно: <https://www.sqlite.org/docs.html>
19. Організація баз даних та знань. [Електронний ресурс] Доступно: https://elearning.sumdu.edu.ua/free_content/lectured:89b3d175c06a6b137e410cb14821d0e94549ad5a/latest/44197/index.html
20. Седжвік, Роберт, Уейн, Кевін. Алгоритми на Java, 4-є вид. : Пер. з англ. - М.: ТОВ "І.Д. Вільямс", 2013. с. : іл. - Парал. тит. англ.
21. Android Studio. [Електронний ресурс]. Доступно: <https://developer.android.com/studio/intro>