

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних
технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка інформаційної-аналітичної системи підтримки
міжнародних мандрівок

Виконав студент 2 курсу
групи МІС-20
спеціальності 122 Комп'ютерні науки
Приложенко Вадим Дмитрович

Керівник д.т.н., професор
Казакова Надія Феліксівна

Рецензент Регіональний координатор
Програми EGAP Копиченко І.Ю.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет _____ Комп'ютерних наук, управління та адміністрування
Кафедра _____ Інформаційних технологій
Рівень вищої освіти _____ магістр
Спеціальність _____ 122 Комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____
д.т.н., професор Казакова Н.Ф.
" 28 " жовтня 2021 року

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Приложенко Вадим Дмитрович

(прізвище, ім'я, по батькові)

1. Тема роботи _____ «Розробка інформаційно-аналітичної системи підтримки
міжнародних мандрівок»

керівник роботи _____ Казакова Надія Феліксівна д.т.н професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ОДЕКУ від " 18 " жовтня 2021 року № 216 «С»

2. Строк подання студентом роботи _____ 9 грудня 2021р.

3. Вихідні дані до роботи _____ Аналіз предметної області

Аналіз існуючих аналогів

Підготовка та обробка вибірки даних для застосування

Застосування інструментів розробки мови для розробки програмного
продукту

Тестування програмного продукту, виявлення помилок та їх виправлення

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

Аналіз предметної області

Вибір програмних засобів для розробки інформаційної системи

Проектування застосувань інформаційної системи

Розробка програмних елементів

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

6. Консультанти розділів роботи

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

7. Дата видачі завдання 28 жовтня 2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-бальною шкалою
1	Отримання завдання	28.10		
2	Збір інформації			
3	Аналіз вимог ТЗ, підбір літератури			
4	Підготовка теоретичної частини			
5	Підготовка практичної частини			
6	Підготовка графічного матеріалу			
7	Рубіжна атестація	22-26.11		
8	Оформлення дипломної роботи			
	Подання роботи на кафедру	09.12		
	Перевірка на плагіат	10.12		
	Рецензування	16.12		
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)			

Студент _____
(підпис)

Керівник роботи _____
(підпис)

Приложенко В.Д. _____
(прізвище та ініціали)

Казакова Н.Ф. _____
(прізвище та ініціали)

АНОТАЦІЯ

на магістерську кваліфікаційну роботу
«Розробка інформаційно-аналітичної системи підтримки
міжнародних мандрівок»,
студента Приложєнка Вадима Дмитровича

Актуальність теми магістерської кваліфікаційної роботи обумовлюється значною важливістю міжнародних мандрівок у наш час. Вважаючи пандемічний стан в світі на час виконання роботи, досить зручно мати застосунок який буде повідомляти про стан пандемії в країні, а також значну роль в подорожах має процес оформлення документів.

Мета роботи – аналіз предметної області та програмних засобів, розробка інформаційно-аналітичної системи підтримки міжнародних подорожей.

Об’єкт дослідження – система автоматизації міжнародних мандрівок та системи автоматизації туристичних агенцій.

Методи дослідження – аналітичний метод для визначення основних процесів під час планування подорожей, метод термінологічного аналізу, який дозволить формалізувати область дослідження і забезпечити розкриття сутності явищ, що досліджуються, метод експертних оцінок, що використовуватиметься для оцінювання якості додатків.

Для виконання всіх поставлених задач попередньо було проведено дослідження, завдяки якому були встановлені потреби туристів, а також слабкі місця сучасних систем автоматизації подорожей. Далі вказані можливості, технології, принципи створення комунікаційних систем та обладнання автоматизації.

Магістерська кваліфікаційна робота містить 70 сторінку, 18 рисунків, 4 таблиці та 15 джерел.

КЛЮЧОВІ СЛОВА: АНАЛІЗ, МІЖНАРОДНІ ПОДОРОЖІ, КОРИСТУВАЧ, ТУРИЗМ, ПОВІДОМЛЕННЯ.

SUMMARY

for a master's degree

"Development of an information-analytical system and support international mandates", students of Prylozhenko Vadym

The relevance of the topic of master's qualification work is due to the significant importance of international travel in our time. Considering the pandemic state in the world at the time of work, it is quite convenient to have an application that will report on the state of the pandemic in the country, as well as a significant role in travel has a process of paperwork.

The purpose of the work - analysis of the subject area and software, development of information and analytical system to support international travel.

The object of research is the system of automation of international travel and the system of automation of travel agencies.

Research methods are an analytical method for determining the main processes during travel planning, a method of terminological analysis that will formalize the field of research and ensure the disclosure of the essence of the studied phenomena, the method of expert evaluation to be used to assess the quality of applications.

To fulfill all the tasks, a preliminary study was conducted, which identified the needs of tourists, as well as the weaknesses of modern travel automation systems. The following are the capabilities, technologies, principles of communication systems and automation equipment. The main advantages of this method are the current development of information and communication technologies.

The master's thesis contains 70 pages, 18 figures, 4 tables and 15 sources.

KEY WORDS: ANALYSIS, INTERNATIONAL TRAVEL, USER, TOURISM, MESSAGES.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
2 ВИБІР ТА ОБҐРУНТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	22
2.1 Обґрунтування вибору архітектури.....	22
2.2 Вибір протоколу віддаленого адміністрування	24
2.2 Вибір платформи розробки бек-енда	26
2.3 Вибір мови програмування для розробки фронт-енду.....	29
2.4 Обґрунтування засобу для розробки бекенду	31
2.5 Обґрунтування засобу для розробки фронтенду	35
2.6 Вибір та обґрунтування СУБД	40
2.7 Вибір IDE	44
3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДТРИМКИ МІЖНАРОДНИХ МАНДРІВОК	52
3.1 Проектування діаграми класів в UML	52
3.2 Проектування бази даних та її реалізація.....	54
4 ОПИС ПРОГРАМНИХ ЕЛЕМЕНТІВ	59
4.1 Огляд інтерфейсу з боку гостя	59
4.2 Опис інтерфейсу користувача	65
4.3 Опис інтерфейсу адміністратора	66
ВИСНОВОК	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ПП – програмний продукт

API – Application Programming Interface

SQL – Structured Query Language

IDE – Integrated Development Environment

ORM – Object-relational mapping

HTTP – HyperText Transfer Protocol

DAO – Data Access Objects

URL – Uniform Resource Locator

JWT – JSON Web Token

JSON – JavaScript Object Notation

СУБД – система управління базою даних

UML – Unified Modeling Language

ВСТУП

Як велику кількість інших важливих в нашому житті галузей, інформаційні технології також досить сильно змінили таку важливу для нас галузь як пасажиро-перевезення, а також спростили таку тяжку для багатьох справу як перетин кордонів, так як тепер не потрібно чекати велику кількість часу на підтвердження ваших документів, а після цього ще й стояти в чергах на отримання потрібних вам документів для подорожі. В наш час все це автоматизовано до такого стану, що ви можете здобути всі необхідні документи не виходячи з дому.

Український ринок туристичних послуг, що склався останніми роками, в цілому, за відсутності координуючого початку знаходиться в кризисному стані, і українські об'єкти, не дивлячись на величезний рекреаційний потенціал країни, до цих пір не включені в постійні міжнародні туристичні маршрути. Одна з причин подібного протиріччя полягає у відсутності надійної і достовірної інформації про стан ринку і звичного для західного клієнта сервісу, що базується на всіх можливостях, які забезпечуються сучасними засобами зв'язку. На сьогодні, процес формування і відправки туристичних груп за кордон і, навпаки, без використання інформаційних технологій недостатньо ефективний.

Але нажалі в останні два роки організувати подорожі за кордон стало досить важко навіть з таким рівнем автоматизації, через декілька факторів:

1. Пандемія – ситуація при якій по всьому світу розповсюджується невідоме до цього захворювання. Такі ситуації виникають при розповсюдженні нового типу грипу до якого населення ще не здобуло імунітету, що викликає епідемії у різних країнах з великою кількістю захворювань а також смертей. Під час пандемії більшість країн перекриває свої кордони, а також вводить надзвичайний стан.

2. Загострення відносин між країнами – також, нажаль, в наші хоч і мирні часи бувають конфлікти між державами, через це подорожі у деякі країни повністю або лише для громадян окремих країн неможливі.

Було б досить зручно мати додаток який повідомляє про стан епідемії в країні а також показує останні новини про взаємовідносини між країнами.

Актуальність тематики. Сьогоднішні темпи комп'ютеризації перевищують темпи розвитку всіх інших галузей. Без комп'ютерів і комп'ютерних мереж не обходиться сьогодні жодна середня фірма, не кажучи про великі компанії. Сучасна людина починає взаємодіяти з комп'ютером постійно – на роботі, вдома, в машині і навіть в літаку.

Останніми роками з'явилася значна кількість невеликих організацій, які надають різноманітні посередницькі послуги. Прикладами таких організацій можуть служити агентства по працевлаштуванню, агентства нерухомості, туристичні фірми. Для таких організацій характерна необхідність в централізованому зберіганні інформації, до якої необхідно забезпечити оперативний доступ менеджерам фірм, що працюють у філіалах агентства, розміщених в різних населених пунктах. Часто подібні філіали дуже невеликі (1–2 співробітники). Тому для автоматизації діяльності таких організацій необхідно вибирати достатньо дешеві і стандартні засоби. Доцільним рішенням в подібній ситуації є використання технології World Wide Web і мережі Інтернет.

Для досягнення поставленої мети в роботі треба реалізувати наступні задачі:

- дослідити предметну область;
- проаналізувати основні проблеми предметної області;
- провести аналіз існуючих аналогів додатків для планування подорожей
- провести аналіз існуючих статей про подорожі та її проблеми.

Об'єктом дослідження є індустрія подорожей, її проблеми та напрямки розвитку у майбутньому.

Предмет дослідження. Предметом дослідження є методи та технології розроблення та реалізації додатків для подорожей.

Методи дослідження. Для досягнення поставленої мети та розв'язання визначених завдань будуть використані наступні методи дослідження: аналітичний – для визначення основних процесів під час планування подорожей; метод термінологічного аналізу, який дозволить формалізувати область дослідження і забезпечити розкриття сутності явищ, що досліджуються; метод експертних оцінок, що використовуватиметься для оцінювання якості додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Інформаційні технології допомагають досягати поставлених цілей, автоматизуючи виробничі процеси, забезпечувати виконання стандартів, удосконалювати продукти на основі аналізу попиту споживачів.

Завдяки додаткам в значній мірі спрощена організація мандрівок, закупівля білетів, пошук готелю або квартири, оренда транспорту, отримання інформації про те на скільки безпечно в країні, вартість локальної валюти в країні.

Для подорожей використовують різні типи транспорту і автоматизувати процес оформлення документів, а також спростити вибір шляху є метою розроблюваного ПП. Слід розглянути різні види транспорту для подорожей.

Залізничний транспорт відіграє велику роль в Україні. Він залишається найбільш прибутковим для держави і найпопулярнішим у громадян нашої країни. У порівнянні з іншими видами транспорту ці перевезення порівняно недорогі і менш небезпечні, ніж, скажімо, перевезення повітряним транспортом.

Для подорожей на великі дистанції найзручнішим типом транспорту є літак, хоч багато людей і вважають його небезпечним по статистиці порівнюючи з автомобільними подорожами він безпечніше. Додаток дозволяє побудувати план подорожі літаком зважаючи на пересадки між країнами.

Багато туристичних фірм часто суміщають функції туроператора і туристичного агентства, виступаючи як туроператор (компанія, що повністю відповідає за тур) поодиночі або декільком напрямам і як туристичне агентство по багатьох інших напрямів, для асортименту.

Туристична фірма є багатопрофільним оператором і працює як на внутрішньому так і на зовнішньому ринку, і займається в'їзним і виїзним туризмом. Фірма розробляє і реалізує тури. Управлінням даною системою

займається туроператор, якою виконує основну роботу, і юрист, який займається юридичними питаннями. Коли клієнт приходить в турфірму він повинен оформити замовлення і надати відповідні документи, всі дані мають бути представлені за державними правилами і відповідно до закону України "Про туризм", а також відповідати внутрішнім правилам фірми і її зразкам. Всі надані дані перевіряються, а потім зберігаються в базу фірми за допомогою програмного технічного забезпечення. Якщо клієнт звертатиметься до цієї ж туристичної фірми, йому буде надана знижка і додаткові послуги. Якщо клієнта все влаштовує, то він отримує на руки путівку і оплачує надані послуги, якщо ж ні, то йому потрібно переоформити замовлення.

Ключовим об'єктом предметної області є тур. Для туру атрибутивна назва, дата початку, тривалість, вартість і курорт, який є кінцевим пунктом перебування.

Також, найпопулярнішим видом мандрівок на незначні порівняно з іншими видами транспорту є мандрівки автобусами або власним авто. Для цього типу подорожей ПП надає можливість переглянути через який кордон зручніше покинути країну, а також допомагає оформити документи для перетину кордону, що зберігає велику кількість часу і надає вам більше часу.

Основними етапами при розробці програмного продукту були:

- Огляд платформи на якій розроблявся продукт;
- Огляд інструментів для розробки фронт-енд частини;
- Огляд інструментів для розробки бек-енд частини;
- Вибір СУБД для створення БД продукту;
- Пошук додаткового інструментарію;
- Вибір середовища розробки (IDE) у якому створюється програмний продукт;
- Ознайомлення і вибір версії платформи
- Вибір менеджера пакетів

1.2 Огляд та аналіз аналогів

Зараз в світі існує маса інформаційних джерел, домінуючим засобом зберігання яких є системи управління базами даних. Але відвертість інформації в багатьох базах даних зовсім не означає легкість доступу до даних для непрофесійного користувача, оскільки для цього необхідний не тільки фізичний доступ до відповідної системи управління базами даних, але також і знання про використовувану модель даних, схемі бази даних, уміння користуватися мовою запитів. Тому на сьогодні дана проблема надання зручного доступу до наявних у наявності баз даних залишається дуже актуальною для багатьох організацій, компаній, наукових установ, і рішення її бачиться тільки у світлі застосування Web-технології. World Wide Web дозволяє здійснювати доступ до бази даних, надаючи засоби для розробки простого, зручного інтерфейсу користувача і засобу взаємодії з прикладними програмами.

Як приклади існуючих сьогодні Інтернет-сайтів були розглянуті сайти популярних у всьому світі, зокрема в Україні туристичних фірм.

Сайт «Go Nomad»

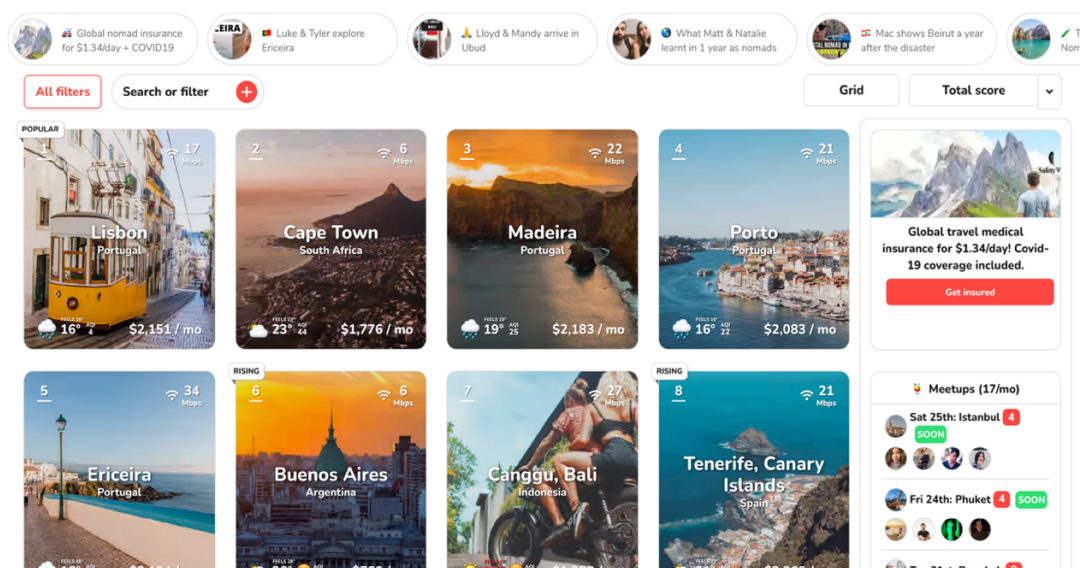


Рисунок – 1.1 сайт «Go Nomad»

Даний веб-додаток є найбільш популярним за багатьма рейтингами, ознайомившись з ним було виявлено що він надає досить зручні можливості. В додатку доступний фільтр пошуку країни по критеріям, також можна створювати набори фільтрів які зберігаються. Розділ з інформацією про країни організований у вигляді карток з інформацією по головним для туризму містам, на картці відображається фото міста, середня вартість проживання, середня швидкість інтернету і погода на сьогоднішній день. При наведенні на картку відображається рейтинг по основним критеріям, таким як:

- Рейтинг вартості проживання
- Рейтинг рівня злочинності
- Рейтинг оснований на відгуках користувачів
- Підсумковий рейтинг

Доступний перегляд останніх світових новин, а також є можливість оформлення страхування.

Зв'язок з оператором виконується через заповнення форми де вказується ваш Email і питання.

Сайт «Tripadvisor»

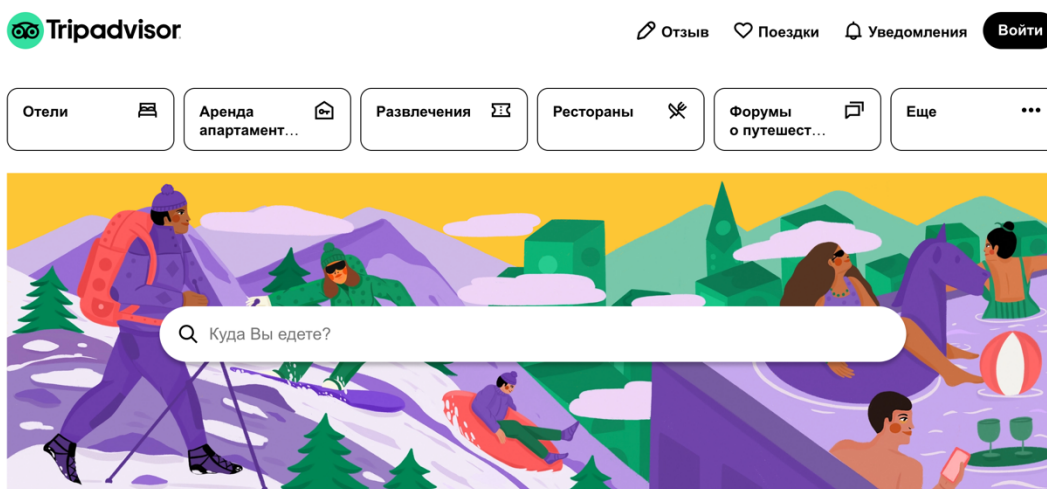


Рисунок – 1.2 сайт «Tripadvisor»

Даний додаток є досить популярним в нашій країні, надає досить зручні для користувача можливості. Увійшовши на головну сторінку ми можемо як авторизуватися, так і переглядати потрібну нам інформацію без авторизації облікового запису. На головній сторінці ми бачимо кнопки з розділами по яким можна отримати інформацію, а також вікно пошуку інформації по країні в якій ми зацікавлені.

Після авторизації ми можемо зв'язатися з оператором, відмічати цікаві для нас подорожі, користуватися форумом і залишати відгуки по поїздкам. Також у користувача є можливість зберегти подорожі. При нажаті на кнопку «Оренда апартаментів» з'являється вікно де користувач може обрати готель в якому він хотів би зупинитися и одразу ж орендувати кімнату.

Сайт «TripIt»

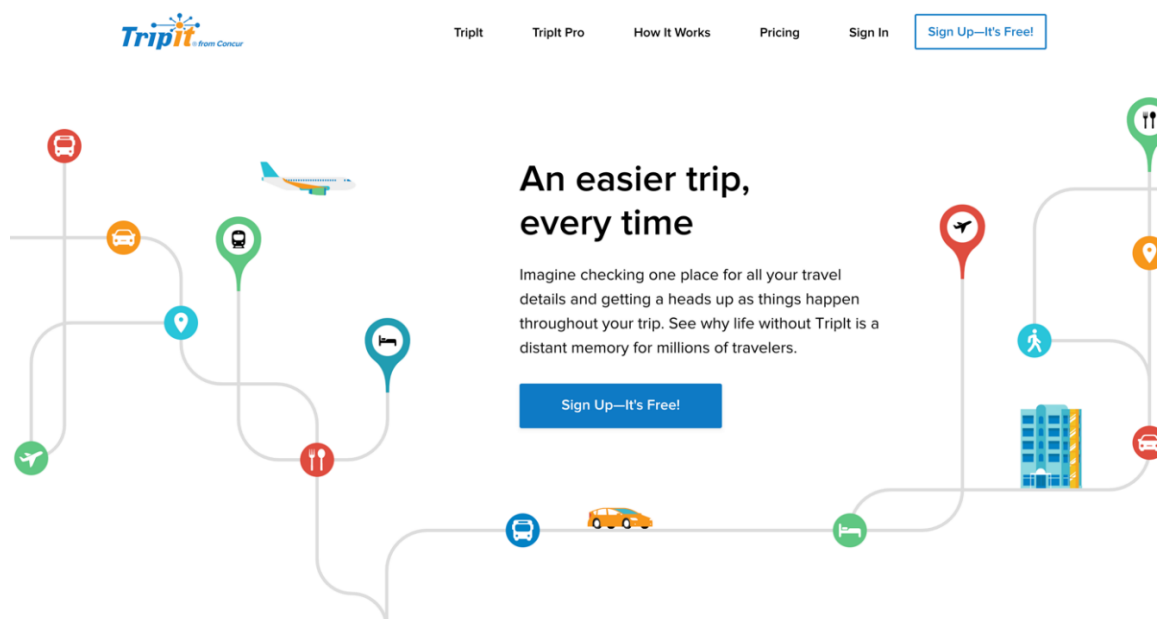


Рисунок 1.3 – сайт «TripIt»

Сайт має досить зручний інтерфейс, але недоліком є те що для роботи потрібна реєстрація. Як можна зауважити на рисунку, при реєстрації можна обрати тип облікового запису «TripIt» та «TripIt Pro».

Увійшовши на сайт ми можемо бачити головну сторінку на якій в верхній частині розташований логотип компанії а також меню. В основній

частині розташована кнопка для авторизації. Як вже було зауважено вище без авторизації на даному веб-сайті ми не маємо доступу до його функціоналу, який досить вражає

Спектр функціоналу даного сайту вражає, при його використанні процес подорожей в значній кількості спрощений. Основними можливостями сайту є:

- Трекінг подорожі – основною особливістю даного сайту є зручний трекер подорожі, в якому ви можете не лише обрати країну в яку хочете поїхати, а повний шлях з усіма проміжними пунктами від виклику таксі до аеропорту або залізничної станції до прибуття до готелю і навіть по прибутті можна запланувати всі подорожі по місту.
- Інформація по країнам в які можна подорожувати в даний час
- Нотифікація користувача при зміненнях у часі посадки на літак або відправки/прибуття потягу
- Можливість замовлення квитків через сайт
- Зберігання мандрівок у відповідні секції: Персональна або Ділова

Також даний веб-додаток має мобільний-додаток завдяки якому можна користуватися трекером навіть у режимі онлайн.

Інтерфейси Інтернет-систем туристичних агенцій багато в чому схожі і є структурою, що складається з трьох основних компонентів. Розглянемо їх докладніше:

Список країн, з якими працює туристична фірма. Як правило, це лінійний або випадний список. Цей компонент інтерфейсу значно спрощує навігацію користувача, який визначився в своєму виборі. Стандартна послідовність дій “Тури → Пошук → Вводимо дані → Одержуємо результат” спрощується до єдиного “Вибираємо результат”. Проте, це не дасть переваг для клієнта, який не визначився в своєму виборі.

Новини туризму і туристичної компанії. Тут представлена інформація, яка може зацікавити потенційного покупця. Крім того, робиться упор на події,

які відбуваються або відбуватимуться в курортах, з якими працює туристичне агентство.

Основна частина сторінки сайту, що служить в різних системах по різному. Найчастіше зустрічається схема, де представлені тури, курорти або країни, вибрані випадковим чином. Так само поширена схема, коли в основній частині розташовані додаткові механізми навігації, наприклад, список турів. Ця схема має істотний недолік, оскільки, фактично, дублює інформацію, розташовану в першому компоненті.

При проектуванні застосувань Інтернет-системи фірми-організатора подорожей необхідно виключити подібні операції, оскільки більшість з наведених не має ніякого відношення до виконуваної користувачем роботи.

Інформаційна система, що розробляється, повинна увібрати в себе найбільш важливі позитивні сторони приведених систем. Головною її відмінністю повинна бути орієнтованість на управління Інтернет-сайтом туристичної фірми. Це дозволить виконати оптимізації, як на клієнтському, так і на серверному боці. Наприклад, замість використання об'єктів представлених у вигляді “списку атрибутів”, можна використовувати об'єктну систему. Саме це, у свою чергу, дозволить розробити зручний призначений для користувача інтерфейс, який не обмежений узагальненою моделлю використання, застосованою в інших інформаційних системах.

Застосування повноцінного сервера застосувань дозволить забезпечити вищий рівень безпеки. Розмежування прав доступу користувачів аж до мінімальної атомарної операції виключить можливість отримання доступу до даних без дозволу на те системного адміністратора.

Крім того, необхідно зробити упор на полегшення роботи з самою програмою. Всі операції в системі повинні бути структуровані і однозначні. Наприклад, якщо потрібно додати нову новину, досить виконати перехід News→Add, ввести дату публікації, заголовок новини, а так само її повний текст, як новина буде доступна на сайті і збережена в базі даних.

Розглянувши приведені вище аналоги можна порівняти їх, а також

розроблюваний додаток у таблиці приведеній нижче.

Назва продукту	Переваги	Недоліки	Мобільний додаток
Сайт «Go Nomad»	Детальна інформація. Не потребує авторизації.	Досить складний інтерфейс. Багато реклами.	Немає додатку.
Сайт «Tripadvisor»	Зручний інтерфейс. Форум для користувачів.	Не надає інформацію про країну.	Немає додатку.
Сайт «TripIt»	Має трекінг подорожі. Має мобільний додаток	Потребує авторизації. Обмежений функціонал без покращеної версії.	Має додаток.

Розглянувши переваги і недоліки аналогів можна зробити висновок, що розробка проекту є актуальною, так як програма має зрозумілий інтерфейс та поєднує в собі основні переваги проаналізованих аналогів.

1.3 Постановка задачі

Останнім часом у громадян України з'явилося більше можливостей подорожувати не лише по Україні, але і за кордоном. Аби підібрати країну відпочинку потрібно виділити час для походу в турагентство, для цього існують сайти турагентств. На сайтах розміщена необхідна інформація для клієнтів. Робота туристичного агентства включає виконання ряду операцій, які вимагають великих витрат часу, якщо виконувати їх уручну. До таких

операцій можна віднести пошук конкретних турів, обчислення вартості турів, аналіз заявок клієнтів. Цю проблему можна вирішити за допомогою розробки програмного комплексу для автоматизації пошуково-інформаційного сервісу турагентства. Цінність створюваної програми в наданні клієнтові можливості через сайт здійснювати пошук туру, а потім відправляти заявку на вподобаний тур. Програмний комплекс забезпечує аналіз даних з виявленням найбільш коштовної для користувача інформації, а також надає користувачеві дані в найбільш інформативному вигляді. Таким чином, дана розробка повинна забезпечити користувачеві скорочення часу на пошук даних про тур і його замовлення. На основі отриманої інформації користувачеві легко прийняти відповідне рішення. Крім того, база даних, створена в рамках інформаційної системи, забезпечує працівникам туристичного агентства аналіз поданих заявок, коментарів людей, що побували в турі.

В роботі потрібно спроектувати і реалізувати застосування Internet-системи підтримки міжнародних мандрівок. Необхідно забезпечити високу продуктивність і відмово стійкість системи, а також легкість адміністрування системи.

Програмний комплекс повинен складатися з серверної частини, модуля управління системою, з яким взаємодіятиме адміністратор і користувачі, а також базою даних, що представляє постійне сховище даних системи.

Так само повинен бути створений Інтернет-сайт, що надає послуги компанії кінцевим користувачам. Сайт повинен дозволяти переглядати інформацію про ті країни, курорти і готелі, з якими працює туристичне агентство, надавати функції пошуку даних, відповідно до побажань клієнтів. Крім того, сайт повинен дозволяти переглядати фотографії країн, курортів і готелів.

Важливою вимогою є забезпечення безпеки інформації. Система не повинна надавати прямий доступ до бази даних ні для однієї з підсистем, за виключення сервера додатків. Сервер застосувань, у свою чергу, не повинен містити інтерфейс доступу до бази даних. Потрібно реалізувати систему

розмежування прав доступу на основі Access Control List.

Нижче наведений список функціональних можливостей системи.

1) Створення, видалення, редагування країн, курортів і готелів. Для кожного об'єкту передбачити можливість задання фотографій. Кількість зображень не обмежена.

2) Створення, видалення, редагування турів.

3) Реєстрація і обробка замовлень клієнтів. Передбачити можливість здійснення замовлення клієнтом через Web-інтерфейс застосування, а так само можливість замовлення туру безпосередньо з робочого місця користувача клієнтського застосування, у разі обслуговування клієнтів в офісі.

4) Реалізація гнучкої системи пошуку країн, курортів, турів і готелів. Пошук повинен бути можливий по всіх атрибутах об'єктів.

5) Забезпечення можливість редагування, створення і видалення застарілих новин. Відразу ж після редагування новини всі зміни повинні бути доступні на Інтернет-сайті.

6) Забезпечення можливості створення, видалення і редагування груп і користувачів.

7) За умовчанням в Інтернет-системі повинні бути присутніми групи Адміністратори, Менеджери і Сайт. Надалі може бути створено необмежену кількість груп.

8) Користувачі, що належать до ролі Адміністратори, повинні мати повний і необмежений доступ до всіх функцій системи. На них покладається обов'язок управління інформаційною безпекою.

9) Користувачі, що належать до ролі Менеджери, повинні мати право на реєстрацію і обробку замовлень і турів, додавання, видалення і редагування новин. Так само можливий перегляд всіх даних, за винятком даних про інших користувачів і груп, до яких вони належать.

10) Користувачі, що будуть належати ролі Сайт, повинні мати право на реєстрацію нових замовлень, перегляду новин, країн, турів і

курортів. У користувача повинні бути права на перегляд фотографій, що асоціюються з яким-небудь об'єктом системи.

У дипломній роботі треба заповнити базу даних, розробити сервер застосувань та сайт.

2 ВИБІР ТА ОБҐРУНТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Обґрунтування вибору архітектури

У даній дипломній роботі реалізована, так звана, "трирівнева архітектура". Це дозволило створити високопродуктивний додаток, який просто підтримувати і масштабувати. Завдяки даній архітектурі можливо використання одного і того ж програмного інтерфейсу як для роботи клієнтського додатку, так і для роботи Web-сайту.

Архітектура клієнтського застосування побудована на основі бібліотеки Spring. Завдяки цьому, стало можливим забезпечити широкі можливості щодо інтеграції додаткових модулів безпосередньо в інтерфейс застосування без модифікації основного коду. Використання даної технології дозволило створити модульну структуру, в якій немає жорстких обмежень на функціональність. Кожен модуль ізольований від системи до певного ступеня і вільний сам вирішувати яку функціональність надавати системі.

Трирівневий додаток складається з наступних компонентів (рис.2.1).

1) Інтерфейсний (звичайно графічний) компонент представляє перший рівень. Це може бути, як віконний додаток, так і консольний клієнт. У багатьох випадках цей рівень може бути представлений додатком з Web-інтерфейсом. Перший рівень не повинен мати прямих зв'язків з базою даних і зберігати стан додатку.

2) На другому рівні розташовується сервер додатків. Це може бути програмна система, написана на будь-якій сучасній мові програмування, але, як правило, найчастіше реалізація створюється на мовах Java C# або C++. На другому рівні зосереджена велика частина прикладної логіки. Поза ним залишаються фрагменти, що експортуються на клієнтські системи, а також, занурені в третій рівень, такі, що зберігаються процедури і тригери.

3) Третій рівень забезпечує зберігання даних. Звичайно це стандартна реляційна або об'єктно-орієнтована СУБД. Якщо третій рівень є базою даних разом з процедурами, що зберігаються, тригерами і схемою, що описує додаток в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що пов'язує клієнтські компоненти з прикладною логікою бази даних.

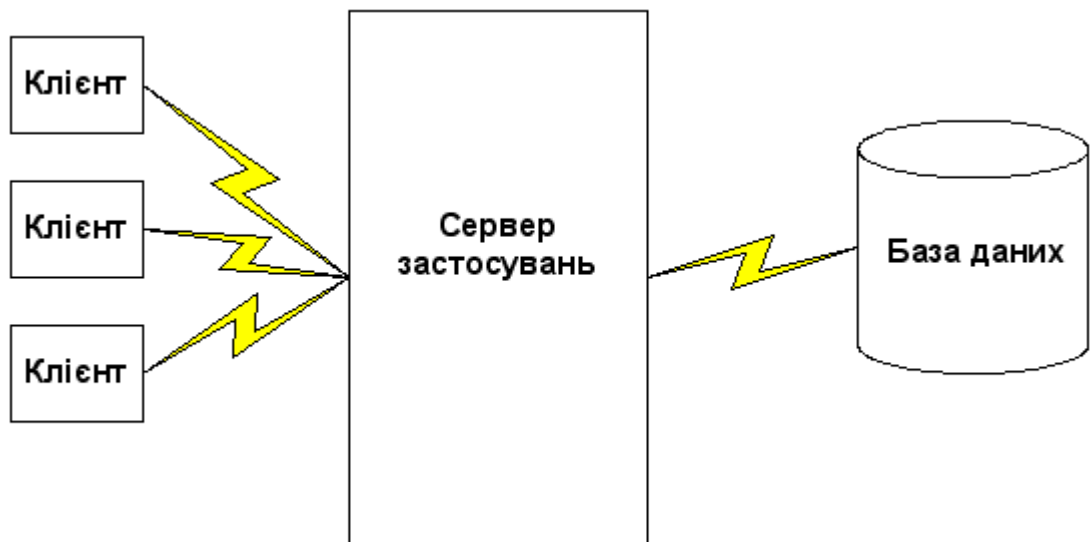


Рисунок 2.1 – Трирівнева архітектура «клієнт-сервер»

Завдання багаторівневої архітектури – ізолювати рівні один від одного. У трирівневій архітектурі відносно легко можна поміняти СУБД (наприклад, з Oracle на MSSQL або, навпаки) – для цього не потрібно переписувати додаток клієнта наново. Також, відносно легко можна створити новий тип клієнта (наприклад, є віконний клієнт з багатим призначенням для користувача інтерфейсом, а ми додаємо Web-клієнта).

Не рекомендується концентрувати виконання всіх логічних компонентів додатків на другому рівні. Під час роботи деякі фрагменти можуть частково виконуватися як на першому, так і на третьому рівні. До фрагментів, які, так чи інакше, виконуватимуться на першому рівні, можна віднести:

- перевірку коректності вхідних даних;
- засоби проведення нескладних операцій з даними, вже присутніми

на клієнтській системі, наприклад, фільтрацію раніше завантаженої інформації.

Основними перевагами трирівневої архітектури в порівнянні з дворівневою є:

- масштабованість;
- конфігуруємість;
- можливість модифікації бізнес-логіки;
- багатократне використання сервісів.

2.2 Вибір протоколу віддаленого адміністрування

Розробці і розгортанню додатків з трирівневою архітектурою передують рішення ряду питань проектування. Проектування трирівневих додатків повинне включати вибір протоколу віддаленого адміністрування та моніторингу. Додати підтримку подібного протоколу після того, як додаток розроблений і/або розгорнений, значно складніше.

Будь-який додаток повинен підтримувати деякий захисний протокол, здатний розмежовувати доступ до сервісів. У трирівневій архітектурі необхідно поклопотатися про безпеку сервісів другого і третього рівнів [4].

Традиційний підхід до захисту дворівневих систем полягає в тому, щоб забезпечити сервер СУБД за допомогою виділення єдиного входу, відомого тільки клієнтському програмному забезпеченню. При запуску клієнтської програми вона використовує цей вхід, підключається до бази даних і видає на екран вікно діалогу, в яке користувач вводить свої вхідне ім'я і пароль. Клієнтська програма перевіряє їх (звичайно вони зберігаються в таблиці бази даних) і лише після цього користувач дістає можливість почати роботу з додатком. Оскільки доступ до бази даних має тільки прикладна програма, написана так, що минути процедуру ідентифікації і аутентифікації користувачів неможливо, яких-небудь додаткових перевірок надалі не вимагається.

У трирівневій архітектурі можна уявити собі аналогічну схему, при якій клієнтське програмне забезпечення встановлює з'єднання з сервером додатків другого рівня, що заздалегідь підключився до бази даних за допомогою процедури входу, відомої тільки цьому серверу. Потім клієнтська програма запрошує у користувача вхідне ім'я і пароль і передає їх серверу другого рівня, що здійснює аутентифікацію за допомогою звернення до бази даних. Тільки після того, як клієнтська програма одержує підтвердження успішної ідентифікації і аутентифікації, вона відкриває користувачу доступ до прикладних компонентів.

На перший погляд, здається, що описана схема надійно захищає від несанкціонованого доступу, проте це не так. Всякий, що знає серверний програмний інтерфейс, може дістати доступ до бази даних через сервер другого рівня, не проходячи процедуру аутентифікації. Це вірно, зокрема, якщо на другому рівні використовуються механізми видаленого виклику методів, оскільки серверний інтерфейс можна з'ясувати, наприклад, засобами рефлексії. Таким чином, хоча база даних і захищена входом, відомим тільки серверу другого рівня, вона насправді виявляється беззахисною перед зловмисником, здатним дістати доступ до цього сервера.

У трирівневій архітектурі необхідно захищати не тільки базу даних, але і сервери другого рівня. Цього можна добитися, якщо надавати доступ до об'єктного посилання на сервер другого рівня тільки авторизованим клієнтам, що успішно пройшли процедуру ідентифікації/аутентифікації.

Web-сервер реалізує схему забезпечення безпеки на основі списків управління доступом, яку можна застосувати і для розмежування доступу до сервера додатків.

Трирівнева архітектура дозволяє централізувати процедуру аутентифікації і авторизації при доступі до другого рівня і використовувати стандартний протокол аутентифікації між першим і другим рівнями.

Користувачам потрібно буде пам'ятати єдину комбінацію "вхідне ім'я/пароль", а накладні витрати на адміністрування вхідних імен на другому

рівні значно знизяться. Користувачі зможуть частіше міняти паролі, спроститься процедура зміни паролів після порушень безпеки. Деактивацію призначених для користувача входів після звільнення співробітників можна буде проводити централізований.

2.2 Вибір платформи розробки бек-енда

Перед початком розробки ПП було розглянуто популярні у наш час платформи розробки: Java, PHP, Python та .NET

Опис Java

На сьогоднішній момент мова Java є однією з найпоширеніших та найпопулярніших мов програмування. Перша версія мови з'явилася ще в 1996 в надрах компанії Sun Microsystems, згодом поглиненої компанією Oracle. Java замислювався як універсальна мова програмування, яку можна застосовувати для різноманітних завдань. І до теперішнього часу мова Java пройшла великий шлях, було видано безліч різних версій. Поточною версією є Java 17, яка вийшла 14 вересня 2021 року. А Java перетворилася з просто універсальної мови на цілу платформу та екосистему, яка поєднує різні технології, що використовуються для цілого ряду завдань: від створення десктопних додатків до написання великих веб-порталів та сервісів. Крім того, мова Java активно застосовується для створення програмного забезпечення для багатьох пристроїв: звичайних ПК, планшетів, смартфонів та мобільних телефонів і навіть побутової техніки. Досить популярність мобільної ОС Android, більшість програм для якої пишуться саме на Java[5].

Для розробки мовою програмування Java використовується спеціальний комплект інструментів, який називається JDK або Java Development Kit. Однак варто відзначити, що існують різні реалізації JDK, хоча всі вони використовують ту саму мову – Java. Дві найбільш популярні продажі – Oracle JDK і OpenJDK[1].

Найбільші відмінності щодо ліцензування. Відповідно до ліцензії Oracle JDK можна використовувати безкоштовно для персональних потреб, а також для розробки, тестування та демонстрації програм. В інших випадках (наприклад, для отримання підтримки) потрібна комерційна ліцензія у вигляді підписки. А OpenJDK повністю безкоштовна.

У плані функціоналу набору можливостей Oracle JDK і OpenJDK практично не повинні відрізнятися. А ось у плані продуктивності зазначається, що Oracle JDK працює дещо швидше, ніж OpenJDK.

Опис PHP

PHP (Hypertext Preprocessor) – це поширена мова програмування загального призначення з відкритим вихідним кодом. PHP спеціально сконструйований для веб-розробок, і його код може впроваджуватися безпосередньо в HTML.

Замість рутинного виведення HTML коду командами мови (як це відбувається, наприклад, Perl або C), скрипт PHP містить HTML з вбудованим кодом (у нашому випадку, це виведення тексту "Привіт, я – скрипт PHP!"). Код PHP відокремлюється спеціальними початковими та кінцевими тегами `<?php` та `?>`, які дозволяють "перемикатися" в "PHP-режим" і виходити з нього.

PHP відрізняється від JavaScript тим, що PHP-скрипти виконуються на сервері та генерують HTML, який надсилається клієнту. Якби у вас на сервері був розміщений скрипт, подібний до вищенаведеного, клієнт отримав би тільки результат його виконання, але не міг би з'ясувати, який саме код його зробив. Ви навіть можете налаштувати свій сервер таким чином, щоб звичайні HTML-файли оброблялися процесором PHP, тому клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта.

PHP дуже простий для освоєння, але водночас здатний задовольнити запити професійних програмістів. Не лякайтеся довгого списку можливостей PHP. Ви можете швидко розпочати і вже протягом кількох годин зможете створювати прості PHP-скрипти.

Хоча PHP, головним чином, призначений для роботи в середовищі веб-серверів, область його застосування не обмежується лише цим. Читайте далі і не пропустіть розділ Можливості PHP або, почніть безпосередньо з Вступного посібника, якщо вас цікавить виключно веб-програмування.

Опис Python

Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів[14].

Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Основними перевагами мови програмування Python є:

- Чистий синтаксис
- Кросс-платформерність
- Стандартний дистрибутив має велику кількість корисних модулів
- Стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написане мовою Python;
- Відкритий код

У Python для розробки бек-енду використовується фреймворк Flask. Фреймворк для створення веб-програм на мові програмування Python, що використовує набір інструментів Werkzeug, а також шаблонизатор Jinja2. Належить до категорії так званих мікрофреймворків — мінімалістичних каркасів веб-додатків, які свідомо надають лише самі базові можливості.

Після проведення аналізу по приведеним вище мовам програмування було прийнято рішення розробляти ПП використовуючи мову програмування Java. Вибір платформи обгрунтовано тим що Java має допоміжні модулі для роботи з СУБД Oracle. Java гірша за .NET тим, що треба качати додаткове ПЗ, прикладом такого ПЗ може бути JVM, але у сучасний час JVM стоїть майже на кожному комп'ютері.

2.3 Вибір мови програмування для розробки фронт-енду

При розробці фронт-енду в наш час головним є HTML (HyperText Markup Language — «мова гіпертекстової розмітки») — базовий будівельний блок Інтернету. Він визначає зміст та структуру веб-контенту. Інші технології, крім HTML, зазвичай використовуються для опису зовнішнього вигляду/уявлення (CSS) або функціональності/поведінки (JavaScript) веб-сторінки.

Під гіпертекстом ("hypertext") розуміються посилання, які з'єднують веб-сторінки один з одним або в межах одного веб-сайту або між веб-сайтами. Посилання є фундаментальним аспектом Інтернету. Завантажуючи контент в Інтернет і пов'язуючи його зі сторінками, створеними іншими людьми, ви стаєте активним учасником Всесвітньої мережі.

Хоч HTML з CSS створюють основу сторінки, те що надає їй сенсу це JavaScript — це мова керування сценаріями перегляду гіпертекстових сторінок Web на стороні клієнта. Якщо бути точнішим, то JavaScript — це не лише мова програмування на стороні клієнта. Liveware, прабатько JavaScript, є засобом підстановок на стороні сервера Netscape. Проте найбільшу популярність JavaScript забезпечило програмування за клієнта.

Основна ідея JavaScript полягає у можливості зміни значень атрибутів HTML-контейнерів та властивостей середовища відображення у процесі перегляду HTML-сторінки користувачем. При цьому перезавантаження сторінки не відбувається.

Назва "JavaScript" є власністю Netscape. Реалізація мови, здійснена розробниками Microsoft офіційно називається Jscript. Версії JScript сумісні (якщо бути точним, то до кінця) з відповідними версіями JavaScript, тобто. JavaScript є підмножиною мови JScript.

Можливості JavaScript залежать від оточення, в якому він працює. Наприклад, Node.JS підтримує функції читання/запису довільних файлів, виконання мережевих запитів тощо.

У браузері для JavaScript доступне все, що пов'язане з маніпулюванням веб-сторінками, взаємодією з користувачем та веб-сервером.

TypeScript — мова програмування, представлена Microsoft у 2012 році і позиціонується як засіб розробки веб-додатків, що розширює можливості JavaScript [14].

TypeScript є сумісним з JavaScript і компілюється в останній. Фактично, після компіляції програму на TypeScript можна виконувати у будь-якому сучасному браузері або використовувати разом із серверною платформою Node.js. Код експериментального компілятора, що транслює TypeScript JavaScript, поширюється під ліцензією Apache. Його технологія ведеться у громадському репозиторії через обслуговування GitHub.

TypeScript відрізняється від JavaScript можливістю явного статичного призначення типів, підтримкою використання повноцінних класів (як у традиційних об'єктно-орієнтованих мовах), а також підтримкою підключення модулів, що покликане підвищити швидкість розробки, полегшити читання, рефакторинг та повторне використання коду, допомогти здійснювати пошук помилок на етапі розробки та компіляції, і, можливо, прискорити виконання програм.

Планується, що з повної зворотної сумісності адаптація існуючих додатків новий мову програмування може відбуватися поетапно, шляхом поступового визначення типів.

Переваги TypeScript були підхоплені творцями ряду поширених і широко використовуваних фреймворків. До слова, одні з найпопулярніших фреймворків для Web – Angular 2+ та Vue3 повністю написані на TypeScript.

Однак, здавалося б, навіщо потрібна ще одна мова програмування для клієнтської сторони в середовищі Web, якщо з усією тією ж роботою чудово справляється і традиційний JavaScript, який використовується практично на кожному сайті, яким володіє безліч розробників і підтримка якого в спільноті програмістів досить висока. Але TypeScript це не просто новий JavaScript.

По-перше, слід зазначити, що TypeScript – це строго типізована і компілювана мова, чим, можливо, буде ближче до програмістів Java, C# та інших типових мов. Хоча на виході компілятор створює той самий JavaScript, який потім виконується браузером. Однак сувора типізація зменшує кількість потенційних помилок, які могли б виникнути під час розробки JavaScript.

По-друге, TypeScript реалізує багато концепцій, які властиві об'єктно-орієнтованим мовам, як, наприклад, успадкування, поліморфізм, інкапсуляція та модифікатори доступу тощо.

По-третє, потенціал TypeScript дозволяє швидше та простіше писати великі складні комплексні програми, відповідно їх легше підтримувати, розвивати, масштабувати та тестувати, ніж на стандартному JavaScript.

2.4 Обґрунтування засобу для розробки бекенду

Spring Framework (або коротко Spring) – універсальний фреймворк із відкритим вихідним кодом для Java-платформи. Також існує форк для платформи .NET Framework, названий Spring.NET[12]

Незважаючи на те, що Spring не забезпечував будь-якої конкретної моделі програмування, він став широко поширеним в Java-спільноті головним чином як альтернатива і заміна моделі Enterprise JavaBeans. Spring надає більшу свободу Java-розробникам у проектуванні; крім того, він надає добре

документовані та легкі у використанні засоби вирішення проблем, що виникають під час створення програм корпоративного масштабу.

Тим часом, особливості ядра Spring застосовні в будь-якому Java-додатку, і існує безліч розширень та удосконалень для побудови веб-програм на Java Enterprise платформі. З цих причин Spring набув великої популярності і визнається розробниками як стратегічно важливий фреймворк.

Використовувати всю функціональність Spring можна на будь-якому J2EE сервері, і більшість – в некерованих середовищах. Spring робить фокус на бізнес об'єктах та об'єктах доступу до даних повторно використання, які не прив'язані до специфічних сервісів J2EE. Такі об'єкти можуть бути багаторазово використані в середовищах J2EE (веб або EJB), автономних додатках, тестових середовищах, і т.п. без будь-яких перешкод.

Головна мета Spring – полегшити використання J2EE та пропагувати хорошу практику програмування. Він реалізує це, забезпечуючи модель програмування на основі POJO, яка застосовна у значній кількості середовищ.

Spring не перевинаходить колесо. Тому в Spring не знайти логуючих пакетів, пулів з'єднань, координатора розподілених транзакцій. Всі ці речі надаються проектами open source (такими як Commons Logging або Commons DBCP) або завдяки серверу застосувань. З тієї самої причини, O/R mapping рівень теж не забезпечується. Для цього хорошим вирішенням буде TopLink, Hibernate та JDO.

Spring не має на меті полегшити використання наявних технологій. Наприклад, хоча координація низькорівневих транзакцій не охоплюється спектром, абстрактний рівень над JTA чи будь-якою іншою транзакційною стратегією забезпечується.

Spring відкрито не змагається з іншими open source до того часу, поки немає можливості запропонувати щось нове. В деяких сферах, таких як IoC контейнер та AOP фреймворк, Spring має пряму конкуренцію, але Spring був піонером в цих сферах.

Spring має переваги від внутрішньої consistency. Всі розробники співають гімн з одного листка, тобто фундаментальні ідеї черпаються з Expert One-on-One J2EE Design and Development. Центральні концепції, як наприклад IoC, використовуються в багатьох місцях.

Spring – мобільна між серверами застосувань. Звичайно, мобільність – це завжди певні труднощі і небезпека, але творці Spring намагаються уникати будь-що платформенно-залежне чи нестандартне з погляду розробника, та надають підтримку на WebLogic, Tomcat, Resin, JBoss, Jetty, Geronimo, WebSphere та інших серверах застосування. Підхід з POJO надає можливість скористатись перевагами рис, специфічних для певного середовища, не жертвуючи при цьому мобільністю.

Spring Boot – це корисний проект, метою якого є спрощення створення програм на основі Spring. Він дозволяє найпростішим способом створити web-додаток, вимагаючи від розробників мінімум зусиль з його налаштування та написання коду

Spring Boot додаток створюється за допомогою Spring Initializer, який дозволяє вибрати компоненти, що використовуються в програмі, а потім зібрати їх проект.

Класи Spring Boot програми були розміщені за наступними пакетами:

- 1) config – для класів конфігурацій;
- 2) controllers – для класів-контролерів, що обробляють запити;
- 3) exceptions – для винятків користувача;
- 4) jwt – для класів, пов'язаних з автентифікацією;
- 5) mappers – для класів, які служать для перетворення об'єктів;
- 6) models – для jpa сутностей та репозиторіїв;
- 7) services – для сервісів.

JWT (JSON Web Token) – JSON об'єкт, що визначається стандартом RFC 7519. JWT служить для безпечної передачі між двома учасниками.

JWT складається з 3 частин – заголовок (header), корисного навантаження (payload) та підписи (signature). Тобто JWT – це рядок у форматі `headerbase64-string.payload-base64-string.signature-base64-string`.

Header є також JSON об'єкт. Він складається із двох полів – `typ`, що описує тип токена, та `alg`, що визначає алгоритм шифрування.

Таким чином, заголовок описує як обчислювати JWT підпис.

Корисне навантаження – це також JSON об'єкт, який стандартно містить три поля: `sub` – предмет, `iss` – додаток, що відправляє токен, `exp` – Час життя токена.

Ключові об'єкти контексту Spring Security:

1. `SecurityContextHolder` забезпечує доступ до `SecurityContext`.
2. `SecurityContext` містить `Authentication` та інформацію про запит у разі потреби.
3. `Authentication` представляє користувача (`Principal`), який включає у собі `GrantedAuthority`.
4. `GrantedAuthority` відображає дозволи для всієї програми, надані користувачеві (`Principal`).
5. `UserDetails` містить необхідну інформацію для створення об'єкта `Authentication` з DAO або іншого джерела даних безпеки.
6. `UserDetailsService` допомагає створювати `UserDetails` з імені користувача на основі рядка.
7. `SecurityContext` встановлюється викликом `SecurityContextHolder.getContext().setAuthentication(...)`
8. `AuthenticationEntryPoint` обробляє виняток `AuthenticationException`.

Коли http запит надходить на сервер, запит проходить через ланцюжок фільтрів.

Так як мовою програмування була обрана Java, на основі аналізу фреймворку Spring було вирішено що він надає достатній функціонал для розробки бекенду.

2.5 Обґрунтування засобу для розробки фронтенду

Для розробки фронт-енду були розглянуті такі програмні засоби як бібліотека React і фреймворк Angular. На основі опису кожного з застосунків для розробки користувацького інтерфейсу було обрано бібліотеку React.

Огляд бібліотеки React.js

React – JavaScript-бібліотека для роботи з користувацьким інтерфейсом, яка була створена розробниками компанії Facebook.

За допомогою React розробники створюють веб-додатки, які змінюють відображення без перезавантаження сторінки. Завдяки цьому програми швидко реагують на дії користувача, наприклад, заповнення форм, застосування фільтрів, додавання товарів у кошик і так далі, такий підхід називається SPA (Single Page Application).

React застосовують для відображення компонентів інтерфейсу користувача. Також бібліотека може повністю управляти фронтендом. У цьому випадку React використовують бібліотеки для керування станом і роутингу, наприклад, Redux і React Router.

React базується на компонентах, це ще одна ключова особливість бібліотеки. Кожен компонент повертає частину інтерфейсу користувача зі своїм станом. Поєднуючи компоненти, програміст створює завершений інтерфейс веб-програми.

Важливою особливістю React є використання JSX. Це розширення синтаксису JavaScript, яке зручно використовувати для опису інтерфейсу. JSX схожий на HTML, проте це все-таки JavaScript.

JSX дозволяє писати JavaScript код за допомогою готових компонентів, які практично повністю повторюють HTML. Це спрощує розробку.

Вся структура веб-сторінки може бути представлена за допомогою DOM (Document Object Model) – організація елементів html, якими ми можемо маніпулювати, змінювати, видаляти чи додавати нові. Для взаємодії з DOM використовується мова JavaScript. Однак, коли ми намагаємося маніпулювати

html-елементами за допомогою JavaScript, ми можемо зіткнутися зі зниженням продуктивності, особливо при зміні великої кількості елементів. А операції над елементами можуть зайняти деякий час, що неминуче позначиться на досвіді користувача. Однак якби ми працювали з коду js з об'єктами JavaScript, то операції робилися б швидше [15].

Для вирішення проблеми продуктивності якраз і виникла концепція віртуального DOM.

Віртуальний DOM є легковажною копією звичайного DOM. І відмінністю React є те, що дана бібліотека працює саме з віртуальним DOM, а не звичайним.

Якщо програмі потрібно дізнатися інформацію про стан елементів, відбувається звернення до віртуального DOM.

Якщо потрібно змінити елементи веб-сторінки, то зміни спочатку вносяться до віртуального DOM. Потім новий стан віртуального DOM порівнюється із поточним станом. І якщо ці стани різняться, то React знаходить мінімальну кількість маніпуляцій, які потрібні до оновлення реального DOM до нового стану і виробляє їх.

У результаті така схема взаємодії з елементами веб-сторінки працює набагато швидше і ефективніше, ніж якщо ми працювали з JavaScript з DOM безпосередньо.

У React для роботи з станами використовуються Хуки – це функції JavaScript, які накладають два додаткові правила:

- Хуки слід викликати лише верхньому рівні. Не викликайте хуки всередині циклів, умов або вкладених функцій.
- Хуки слід викликати лише з функціональних компонентів React. Не викликайте хуки із звичайних JavaScript-функцій. Є тільки один виняток, звідки можна викликати хуки – це ваші хуки. Ми розповімо про них далі.

Ось декілька основних хуків:

- useState

- UseEffect
- UseMemo
- UseContext

Розглянувши бібліотеку React можна зробити висновки що з ним досить легко працювати, він має досить зручний движок який оновлює сторінку під час змінення коду, що досить спрощує процес розробки, також Virtual DOM забезпечує велику продуктивність і мале навантаження на браузер і мережу.

Огляд фреймворку Angular

Angular представляє фреймворк від компанії Google для створення клієнтських програм. Насамперед він націлений розробку SPA-решень (Single Page Application), тобто односторінкових додатків. У цьому плані Angular є спадкоємцем іншого фреймворку AngularJS. У той же час Angular – це не нова версія AngularJS, а принципово новий фреймворк.

Angular надає таку функціональність як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, шаблони, маршрутизація і так далі.

Однією з ключових особливостей Angular є те, що він використовує мову програмування TypeScript.

Але ми не обмежені мовою TypeScript. За бажанням можемо писати програми на Angular за допомогою таких мов як Dart або JavaScript. Однак TypeScript є основною мовою для Angular.

AngularJS – структурований фреймворк для динамічних веб-додатків. Дозволяє використовувати HTML як мову шаблонів, а також розширювати HTML-синтаксис, щоб код вашої програми виглядав коротко і лаконічно. Використовуючи Data-binding і Dependency injection (що йдуть із коробки), ви також зменшуєте кількість свого коду. І все це на браузерному JavaScript та працює з будь-якою серверною технологією[13].

AngularJS — це набір інструментів для створення фреймворка, який найбільше підходить для розробки ваших додатків. Він повністю розширюється і добре працює з іншими бібліотеками. Кожну функцію можна

змінити або замінити відповідно до вашого унікального робочого процесу розробки та потреб функцій.

Функції AngularJs

Пайпи (pipes)

1. fuel-ui – OrderBy і Range, портовані з AngularJS 1.x в Angular
2. ngx-filter-pipe Пайп (pipe) для фільтрації масивів
3. ngx-pipes набір корисних пайпів для Angular
4. ngx-order-pipe OrderBy – сортування колекцій
5. angular2-camelcase Пайп для перетворення рядків у camelCase
6. angular-pipes — Використовуємо круті пайпи
7. ngx-pipes — Пайпи без жодної залежності
8. ng-pipes — Набір корисних пайпів
9. angular-linky — Linky пайп

Структури даних

1. angular-localstorage — Декоратор для автоматичного збереження та відновлення полів класів з LocalStorage
2. ng-webstorage – LocalStorage та SessionStorage менеджер
3. ng-storage localStorage та sessionStorage обгортки
4. angular-safeguard — Обертання над cookies/sessionStorage/localStorage
5. @ngx-cache/core — Розумне кешування в Angular
6. angular-cookie Бібліотека імплементуюча з AngularJS 1.x \$cookies-сервіс в Angular
7. ng-http-cache – Кешування http-запитів

Роутінг

1. ng-breadcrumb – генератор ієрархії маршрутизації на основі вкладеного роутингу
2. ng-page-transition — Простий компонент з анімованими переходами під час маршрутизації
3. @ngx-118n-router/core — Інструмент інтернаціоналізації роутингу

Валідація

1. `ng-validators` — Простий валідатор реактивних форм на основі `validator.js`
2. `ng-validation` -Проста валідація на Angular
3. `ngx-dynamic-form-builder` — Крутий валідатор реактивних форм на основі `class-validator` та `class-transformer`

Логування

1. `angular-logger` — Реалізація `log4j` для `angular`
2. `@nsalaun/ng-logger` — Простий сервіс логування

i18n

1. `@ngx-translate/core` — Зручна бібліотека для роботи з файлами локалізацій (i18n)
2. `@angular-ua/ngx-i18n-combine` — Об'єднання файлів i18n з компонентів та загальних файлів для ваших локалізацій (json, ts, js, jsx, tsx)
3. `angular-i10n` — Бібліотека для перекладу повідомлень, дат та цифр
4. `@ngx-universal/translate-loader` — Лоадер, який забезпечує переклади на стороні браузер або сервера

Недоліки:

1. Вище поріг входження через `Observable (RxJS)` та `Dependency Injection`;
2. Щоб все працювало добре і швидко потрібно витратити час на додаткові оптимізації (він не супер швидкий, за замовчуванням, але швидше за `AngularJS` у багато разів і з кожною новою версією стає все швидше);
3. `Angular-Universal` має багато підводних каменів;
4. Динамічне створення компонентів виявляється нетривіальним завданням.

Form Builder – для розробки справді складних форм, вам слід знати реактивні форми, точніше важливо забути про декларативні форми. Ось один із добрих прикладів.

На основі опису фреймворку Angular можна зробити висновок що це є закритим фреймворком який веде вас по всьому процесу створення користувацького інтерфейсу, тож вплинути на процес обробки коду досить важко.

2.6 Вибір та обґрунтування СУБД

Огляд СУБД PostgreSQL

PostgreSQL – це популярна вільна об'єктно-реляційна система управління базами даних. PostgreSQL базується на мові SQL і підтримує численні можливості.

Переваги PostgreSQL:

- підтримка бази даних необмеженого розміру;
- потужні і надійні механізми транзакцій і реплікації;
- розширювана система вбудованих мов програмування і підтримка завантаження C-сумісних модулів;
- спадкування;
- легка розширюваність.

Поточні обмеження PostgreSQL:

- Немає обмежень на максимальний розмір бази даних
- Немає обмежень на кількість записів у таблиці
- Немає обмежень на кількість індексів у таблиці
- Максимальний розмір таблиці – 32 Тбайт
- Максимальний розмір запису – 1,6 Тбайт
- Максимальний розмір поля – 1 Гбайт
- Максимум полів у записі 250-1600 (залежно від типів полів)

Функції в PostgreSQL є блоками коду, що виконуються на сервері, а чи не на клієнта БД. Хоча вони можуть писатися на чистому SQL, реалізація

додаткової логіки, наприклад умовних переходів і циклів, виходить за рамки власне SQL і вимагає використання деяких мовних розширень. Функції можуть писатися за допомогою різних мов програмування. PostgreSQL допускає використання функцій, що повертають набір записів, який можна використовувати так само, як і результат виконання звичайного запиту. Функції можуть виконуватися як із правами їхнього творця, так і з правами поточного користувача. Іноді функції ототожнюються з процедурами, що зберігаються, проте між цими поняттями є відмінність.

PostgreSQL може бути розширений користувачем для потреб практично в будь-якому аспекті. Є можливість додавати власні:

- Перетворення типів
- Типи даних
- Домени (користувацькі типи з спочатку накладеними обмеженнями)
- Функції (включаючи агрегатні)
- Індeksi
- Оператори (включаючи перевизначення вже існуючих)
- Процедурні мови

Огляд СУБД Oracle

СУБД Oracle має унікальні якості переносимості, а також надає відкриту платформу для розробки додатків, що переносяться клієнт/сервер і Internet/Intranet-додатків.

Серед основних властивостей СУБД Oracle слід зазначити такі, як:

1. Висока надійність;
2. Можливість розбиття великих баз даних на розділи (large-database partition), що дозволяє ефективно управляти гігантськими гігабайтними базами;
3. Наявність універсальних засобів захисту;
4. Ефективні методи максимального підвищення швидкості обробки запитів;

5. Вільні таблиці;
6. Розпаралелювання операцій у запиті;
7. Наявність широкого спектру засобів розробки, моніторингу та адміністрування;
8. Орієнтація на інтернет технології та інші.

На основі СУБД Oracle Database було побудовано Oracle Apex (Oracle Application Express) вільне середовище швидкої розробки прикладного програмного забезпечення, повністю реалізоване як веб-додаток[3].

Oracle Application Express – це вільне середовище розробки програмного забезпечення на основі СУБД Oracle Database, реалізована і представлена як веб-додаток. Елементи кожної її розробки зберігаються у самій Oracle Database, цим отримуючи спільну роботу розробників і контроль версій без використання файлів і додаткових систем управління версіями.

Апекс сертифікований і може бути встановлений без додаткових витрат. Використовуючи лише веб-браузер, ми отримуємо гнучку систему, за допомогою якої можна розробляти та розгортати професійні програми. Швидко, ефективно та безпечно.

Огляд СУБД MySQL

MySQL – це реляційна система управління базами даних із відкритим вихідним кодом. В даний час ця СУБД одна з найбільш популярних у веб-додатках – переважна більшість CMS використовує саме MySQL (часто тільки її, без альтернатив), а майже всі веб-фреймворки підтримують MySQL вже на рівні базової конфігурації (без додаткових модулів)[2].

З переваг СУБД MySQL варто відзначити простоту використання, гнучкість, низьку вартість володіння (щодо платних СУБД), а також масштабованість та продуктивність.

MySQL дозволяє зберігати цілочислові значення зі знаком та беззнакові, довжиною в 1, 2, 3, 4 та 8 байтів, працює з рядковими та текстовими даними фіксованої та змінної довжини, дозволяє здійснювати SQL-команди SELECT,

DELETE, INSERT, REPLACE та UPDATE, забезпечує повну підтримку операторів та функцій у SELECT- та WHERE- частинах запитів, працює з GROUP BY та ORDER BY, підтримує групові функції COUNT(), AVG(), STD(), SUM(), MAX() та MIN(), дозволяє використовувати JOIN у запитах, у т.ч. LEFT OUTER JOIN та RIGHT OUTER JOIN, підтримує реплікацію, транзакції, роботу із зовнішніми ключами та каскадні зміни на їх основі, а також забезпечує багато інших функціональних можливостей.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, і таблиці InnoDB, підтримують транзакції лише на рівні окремих записів. Існують інші типи таблиць, розроблені спільнотою.

СУБД MySQL з'явилася в 1995 році. Написана на C і C++, протестована на безлічі різних компіляторів і працює на різних платформах. З 2010 року розробку та підтримку MySQL здійснює корпорація Oracle. Продукт поширюється як під GNU GPL, і під власною комерційною ліцензією. Однак за умовами GPL, якщо будь-яка програма включає вихідні коди MySQL, то ця програма теж повинна поширюватися за ліцензією GPL. Для небажаючих відкривати вихідні тексти своїх програм передбачена комерційна ліцензія, яка, на додаток до можливості розробки під «закритою» ліцензією, забезпечує якісну сервісну підтримку. Спільнотою розробників MySQL створено різні відгалуження – Drizzle, OurDelta, Percona Server і MariaDB, всі ці відгалуження вже існували на момент отримання прав на MySQL корпорацією Oracle.

Зараз MySQL разом із форком MariaDB займають почесне перше місце, а слідом за ними йде PostgreSQL. Інші СУБД у веб-проектах використовуються значно рідше.

Після аналізу я обрав СУБД Oracle через ряд переваг: висока надійність та безпека, можливість роботи на платформі будь-якої операційної системи. Також для розгоротання БД мені було надано особистий сервер.

2.7 Вибір IDE

У своєму проекті я обираю між двома IDE – Eclipse, IntelliJ Idea, NetBeans.

Огляд Eclipse

Eclipse – вільне модульне інтегроване середовище розробки програмного забезпечення. Розробляється і підтримується Eclipse Foundation і включає проекти, такі як платформа Eclipse, набір інструментів для програмістів на мові Java, системи контролю версій, конструктори GUI тощо. Написаний в основному на Java, може бути використаний для розробки застосунків на Java і, за допомогою різних плагінів, на інших мовах програмування, включаючи Ada, C, C++, COBOL, Fortran, Perl, PHP, Python, R, Ruby, Scala, Clojure та Scheme. Середовища розробки зокрема включають Eclipse ADT (Ada Development Toolkit) для Ada, Eclipse CDT для C/C++, Eclipse JDT для Java, Eclipse PDT для PHP. Eclipse насамперед повноцінна Java IDE, націлена на групову розробку, має засоби роботи з системами контролю версій (підтримка CVS входить у поставку Eclipse, активно розвиваються кілька варіантів SVN модулів, існує підтримка VSS та інших). З огляду на безкоштовність, у багатьох організаціях Eclipse – корпоративний стандарт для розробки ПЗ на Java[9].

Друге призначення Eclipse – служити платформою для нових розширень. Такими стали C/C++ Development Tools (CDT), розроблювані інженерами QNX разом із IBM, засоби для підтримки інших мов різних розробників. Безліч розширень доповнює Eclipse менеджерами для роботи з базами даних, серверами застосувань та інших.

З версії 3.0 Eclipse став не монолітною IDE, яка підтримує розширення, а набором розширень. У основі лежать фреймворки OSGi, і SWT/JFace, на основі яких розроблений наступний шар – платформа і засоби розробки повноцінних клієнтських застосунків RCP (Rich Client Platform). Платформа

RCP є базою для розробки різних RCP програм як торент-клієнт Azareus чи File Arranger.

Eclipse написана на Java, тому є платформонезалежним продуктом, крім бібліотеки графічного інтерфейсу SWT, яка розробляється окремо для більшості поширених платформ.

Структура і склад eclipse

Eclipse складається з проектів, кожен з яких займається Комітетом управління проектом (Project Management Committee, PMC) відповідно до Статуту проекту. Проекти поділяються на підпроекти (subprojects). Для оформлення нового проекту (підпроекту) ініціативною групою вноситься пропозиція на проект (project proposal), основним документом якого є декларація проекту (project declaration) та проект отримує статус запропонованого (proposed project). Для запропонованих проектів проводиться обговорення їх цілей, тематичного охоплення, організаційної структури проекту, результатів проекту та формування проектної команди.

Докладніше організаційна структура, проектні ролі та життєвий цикл проектів у рамках Eclipse описані у документі The Eclipse Development Process.

На жовтень 2021 року склад Eclipse за проектами, підпроектами та запропонованими проектами відображено на наступній схемі:

Архітектура платформи eclipse

Графічно архітектура платформи Eclipse може бути виражена схемою:

Кожен інструментальний засіб розробки оформляється у вигляді плагіна (plug-in), що реалізує певний набір функцій, що приєднується до платформи Eclipse за допомогою API і написаного на Java. Як правило, плагін являє собою Java-код, оформлений у вигляді архіву JAR, кілька файлів для читання та набір інших ресурсів, необхідних для роботи плагіна, наприклад бібліотеки, графічні зображення, шаблони і т.д.

Open-source технології, що уваги в eclipse

IDE Eclipse є результатом колективної праці як компаній-вендорів, так і некомерційних об'єднань компаній та приватних осіб. Будучи універсальним

середовищем розробки додатків, Eclipse використовує і включає велику кількість open-source технологій та програмних продуктів, які добре зарекомендували себе на ринку у відповідних областях. Нижче ми наводимо перелік цих розробок:

- Ant – Apache Ant Apache Ant; <http://ant.apache.org/index.html>
- Blowfish Encryption Algorithm – плагін, що реалізує 64-бітний алгоритм шифрування;
- GTK+ – багатоплатформний інструментарій створення GUI, частина проекту GNU Project; <http://www.gnu.org>
- JSch – Java-реалізація SSH2; <http://www.jcraft.com/jsch>
- JUnit – інструментарій для тестування та контролю роботи додатків; <http://www.junit.org/index.htm>
- Lucene – пошуковий "движок", частина проекту Apache Jakarta; <http://jakarta.apache.org>
- Open Motif for Linux – частина проекту opengroup.org;
- The Java Ssh Applet – плагін, доступний за адресою: <http://www.cl.cam.ac.uk/~fapp2/software/java-ssh>;
- Tomcat – веб-контейнер, частина проекту Apache Jakarta;
- XML Parser for Java – XML-парсер для Java (<http://www.alphaworks.ibm.com/tech/xml4j>) заснований на Apache Xerces.

В цілому, платформа Eclipse надає базис, що складається із загальних будівельних блоків та API для робочих областей та робочого середовища, а також різні точки розширення, через які може інтегруватися нова функціональність. Через ці точки розширення утиліти, реалізовані як окремих плагінів, можуть розширювати платформу Eclipse. Користувач бачить інтегроване середовище розробки (IDE), спеціалізоване набором доступних плагінів. Проте замість того, щоб бути закінченням історії, це насправді лише її початок. Утиліти можуть також задавати нові точки розширення та API, служачи цим будівельними блоками і точками інтеграції з іншими утилітами.

Огляд IntelliJ IDEA

IntelliJ IDEA – комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну ліцензію. Сирцеві тексти Community-версії поширюються в рамках ліцензії Apache 2.0. Бінарні збірки підготовлені для Linux, Mac OS X і Windows[11].

Community версія середовища IntelliJ IDEA підтримує інструменти (у вигляді плагінів) для проведення тестування TestNG і JUnit, системи контролю версій CVS, Subversion, Mercurial і Git, засоби складання Maven, Ant, Gradle, мови програмування Java, Scala, Clojure, Groovy і Dart. Підтримується розробка застосунків для мобільної платформи Android. До складу входить модуль візуального проектування GUI-інтерфейсу Swing UI Designer, XML-редактор, редактор регулярних виразів, система перевірки коректності коду, система контролю за виконанням завдань і доповнення для імпорту та експорту проєктів з Eclipse. Доступні засоби інтеграції з системами відстеження помилок JIRA, Trac, Redmine, Pivotal Tracker, GitHub, YouTrack, Lighthouse.

Комерційна версія «Ultimate Edition» відрізняється наявністю підтримки додаткових мов програмування (наприклад, PHP, Ruby, Python, JavaScript, CoffeeScript, HTML, CSS, SQL), підтримкою технологій Java EE, UML-діаграм, підрахунок покриття коду, можливістю роботи з фреймворками (Rails, Grails, Google Web Toolkit, Spring, Play Framework і Hibernate), засобами інтеграції з Perforce, Microsoft Team Foundation Server і Rational ClearCase.

IntelliJ IDEA – це насамперед середовище розробки для Java. З цією мовою вона дружить найбільше, добре його розуміє і допомагає в написанні розробнику. Але це не означає, що все закінчується на Джаві та власній мові Kotlin. Не менш важлива підтримка їхньої коробки таких передових

технологій, як Groovy, Scala та інших. Вони поєднують у собі можливості динозаврів, як Java разом із функціоналом Ruby, Smalltalk та інших. Не забувайте в компанії та засобах веб-пошуку для окремих середовищ, заснованих на IntelliJ IDEA.

Автодоповнення – не винятковий функціонал. Такі самі можливості ми можемо спостерігати практично в кожній безкоштовній програмі. Причому реалізовані вони на досить високому рівні. Але інтелект, як завжди, відрізняє IntelliJ IDEA. Там, де звичайна програма запропонує кілька, IntelliJ IDEA видає лише один, але вірний. Це заощаджує лише секунду, зате в сукупності робить розробку набагато більш оптимізованою.

Однією із сильних сторін у JetBrains вважають підтримку широкого кола технологій. Наприклад, у всьому світі технології Flash вважаються застарілими. Але незважаючи на це, в деяких продуктах Flash реалізується досить широко і скидати його з рахунків рано. Так вирішили і розробники IntelliJ IDEA. Вони не виключають технологію, поки всі довкола не перестануть нею користуватися. У результаті більшість Flash-розробників мігрує на IntelliJ IDEA. Компанії не такі важливі світові тенденції, у той час як лояльність користувачів зростає.

Не менш важливим інструментом є графічний редактор для створення інтерфейсу. Це значно зручніше, ніж писати шаблонний код. Розробник може лише перетягувати мишкою елементи на робоче поле, тоді як код генерується самостійно. Серед досвідчених програмістів такий підхід можна вважати дилетантським. Але правда в тому, що таким чином економиться багато часу, а якість коду не стає гіршою. Через це візуальні редактори використовують не лише новачки, а й ті, хто просто хоче оптимізувати свою роботу.

Можливості IntelliJ IDEA:

- 1) розумне автодоповнення, інструменти для аналізу якості коду, зручна навігація, розширені рефакторинги та форматування для Java, Groovy, Scala, Clojure та Erlang;

- 2) професійний набір інструментів для розробки Android-додатків;
- 3) підтримка JavaFX 2.0 та інтеграція з SceneBuilder;
- 4) дизайнер інтерфейсу для Swing;
- 5) інтеграція з автоматизованими інструментами збирання та управління проектом, включаючи Maven, Gradle, Ant та інші.
- 6) інструменти для тестування програм з підтримкою JUnit, TestNG, Spock, ScalaTest та spec2;
- 7) інтеграція із системами управління версіями, включаючи Git, Subversion, Mercurial та CSV.

Огляд NetBeans IDE

NetBeans IDE — вільне інтегроване середовище розробки додатків (IDE) мовами програмування Java, JavaFX, Python, PHP, JavaScript, C++, Ада та інших.

Для розробки програм у середовищі NetBeans та для успішної інсталяції та роботи самого середовища NetBeans має бути попередньо встановлений Sun JDK або J2EE SDK відповідної версії. Середовище розробки NetBeans за умовчанням підтримувало розробку для платформ J2SE та J2EE. Починаючи з версії 6.0, Netbeans підтримує розробку для мобільних платформ J2ME, C++ (тільки g++) і PHP без встановлення додаткових компонентів[10].

Проект NetBeans IDE підтримується та спонсорується компанією Oracle, проте розробка NetBeans ведеться незалежно спільнотою розробників-ентузіастів (NetBeans Community) та компанією NetBeans Org.

За якістю та можливостями останні версії NetBeans IDE не поступаються кращим комерційним (платним) інтегрованим середовищам розробки для мови Java, таким, як IntelliJ IDEA, підтримуючи рефакторинг, профільування, виділення синтаксичних конструкцій кольором, автодоповнення набираються конструкцій на льоту, безліч визначених шаблонів .

У версії NetBeans IDE 6.1 декларується підтримка UML, SOA, мови програмування Ruby (включно з підтримкою Ruby on Rails), а також засоби для створення програм на J2ME для мобільних телефонів. У версії 6.5 додано підтримку мови PHP. Також для тестування викладено модуль підтримки Python.

NetBeans IDE підтримує плагіни, дозволяючи розробникам розширювати можливості середовища. Одним із найпопулярніших плагінів є потужний дизайнер звітів iReport (заснований на бібліотеці JasperReports).

На ідеях, технологіях та в значній частині на вихідному коді NetBeans IDE базуються пропоновані фірмою Sun комерційні інтегровані середовища розробки для Java – Sun Java Studio Creator, Sun Java Studio Enterprise та Sun Studio (для розробки на C, C++ або Фортран). Порівняно недавно Sun стала пропонувати ці середовища розробки безкоштовно для розробників, що зареєструвалися в Sun Developer Network (SDN), сама ж реєстрація на сайті безкоштовна і не вимагає жодних попередніх умов, крім згоди з ліцензією CDDL.

NetBeans IDE доступна у вигляді готових дистрибутивів (прекомпільованих бінарних файлів) для платформ Microsoft Windows, Linux, FreeBSD, Mac OS X, OpenSolaris та Solaris (як для SPARC, так і для x86 – Intel та AMD). Для решти платформ доступна можливість скомпілювати NetBeans самостійно з вихідних текстів.

У релізі NetBeans IDE 6.7 була додана інтеграція з Project Kenai, підтримка мови Groovy та веб-фреймворку Grails. У версії 6.8 – підтримка PHP-фреймворку Symfony, а в 6.9 – Zend Framework.

Поточні версії:

NetBeans IDE 6.0, створена на основі попередньої версії 5.5.1, надає гнучку підтримку створення модулів для IDE та інтернет-додатків, заснованих на платформі NetBeans, новий дизайнер інтерфейсів користувача (відомий під назвою «Проект Matisse»), нову та перероблену підтримку системи управління версіями CVS, підтримку Weblogic 9 та JBoss 4, та безліч покращень у

редакторі. NetBeans 6.0 постачається у складі дистрибутивів Ubuntu 8.04 та Debian.

NetBeans IDE 6.5, випущена у листопаді 2008 року, розширює можливості Java EE (включаючи підтримку Java Persistence, EJB 3 та JAX-WS). Додатково NetBeans Enterprise Pack підтримує розробку програм Java EE 5 Enterprise, включаючи візуальні засоби SOA, засоби для роботи з XML schema, роботу з веб-сервісами (для BPEL), та моделювання мовою UML. Складання NetBeans IDE Bundle for C/C++ підтримує проекти мовами C/C++.

Поточна версія середовища – NetBeans IDE 7.0

Після аналізу для себе я обрав IntelliJ Idea, яку компанія JetBrains надає безкоштовно для студентів на рік, та в Ultimate Edition є спеціальний додаток для UML – діаграм, також ця IDE являє собою дуже зручну систему підказок.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДТРИМКИ МІЖНАРОДНИХ МАНДРІВОК

3.1 Проектування діаграми класів в UML

На даній UML (англ. Unified Modeling Language – уніфікована мова моделювання) діаграмі вказано взаємодію класів відповідаючих за звернення до бази даних та API, з класами відповідаючими за обробку даних у системі. Взагалі DAO (Data Access Objects) – це абстрактний інтерфейс до якогось типу бази даних чи механізму зберігання. Певні можливості надаються незалежно від того, який механізм зберігання і без необхідності спеціальним чином відповідати цьому механізму зберігання. Цей шаблон проектування застосовується до безлічі мов програмування, більшої частини програмного забезпечення, що потребує зберігання інформації, і до більшості баз даних. Але зазвичай цей шаблон пов'язують із додатками на платформі Java Enterprise Edition, взаємодіючи з реляційними базами даних через інтерфейс JDBC, оскільки він з'явився у рекомендаціях від фірми Sun Microsystems[7].

У цій моделі головним виступає інтерфейс DAO, він є посередником для взаємодії класів що відповідають за оброблення запитів з фронт-енду і методами що виконуються необхідні операції з БД. Основними проміжними діями є:

- авторизація
- реєстрація
- редагування інформації
- отримання даних про країни
- створення/видалення/редагування маршрутів

На рис. 3.1 зображено UML діаграму DAO проекту, на якій можна побачити усі зв'язки між класами

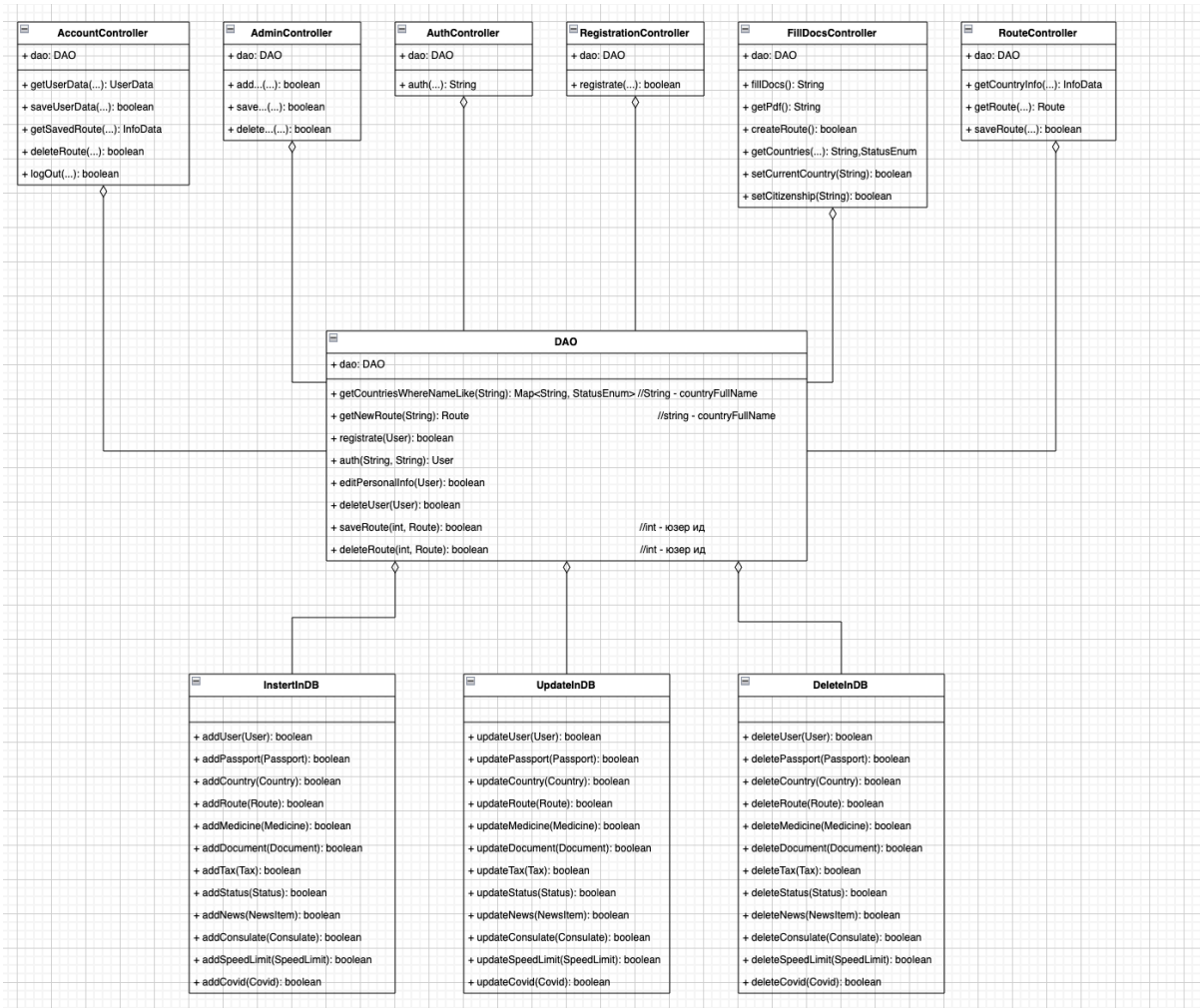


Рисунок 3.1 – діаграма DAO

На даному рисунку зображено UML діаграму взаємодії класів. Як ми можемо бачити головним класом який збирає в собі всі оброблені в системі дані в одну цілу сутність InfoData для збереження даних в БД, а також для обробки і видачі цих даних на фронт-енд. На рис. 3.2 зображено UML діаграму класів проекту, на якій можна побачити усі зв'язки між класами. Можна відзначити основні класи. Клас User відповідає за обробку інформації користувача. Клас Covid відповідає за роботу з сторонніми API для отримання інформації про стан пандемії в світі.

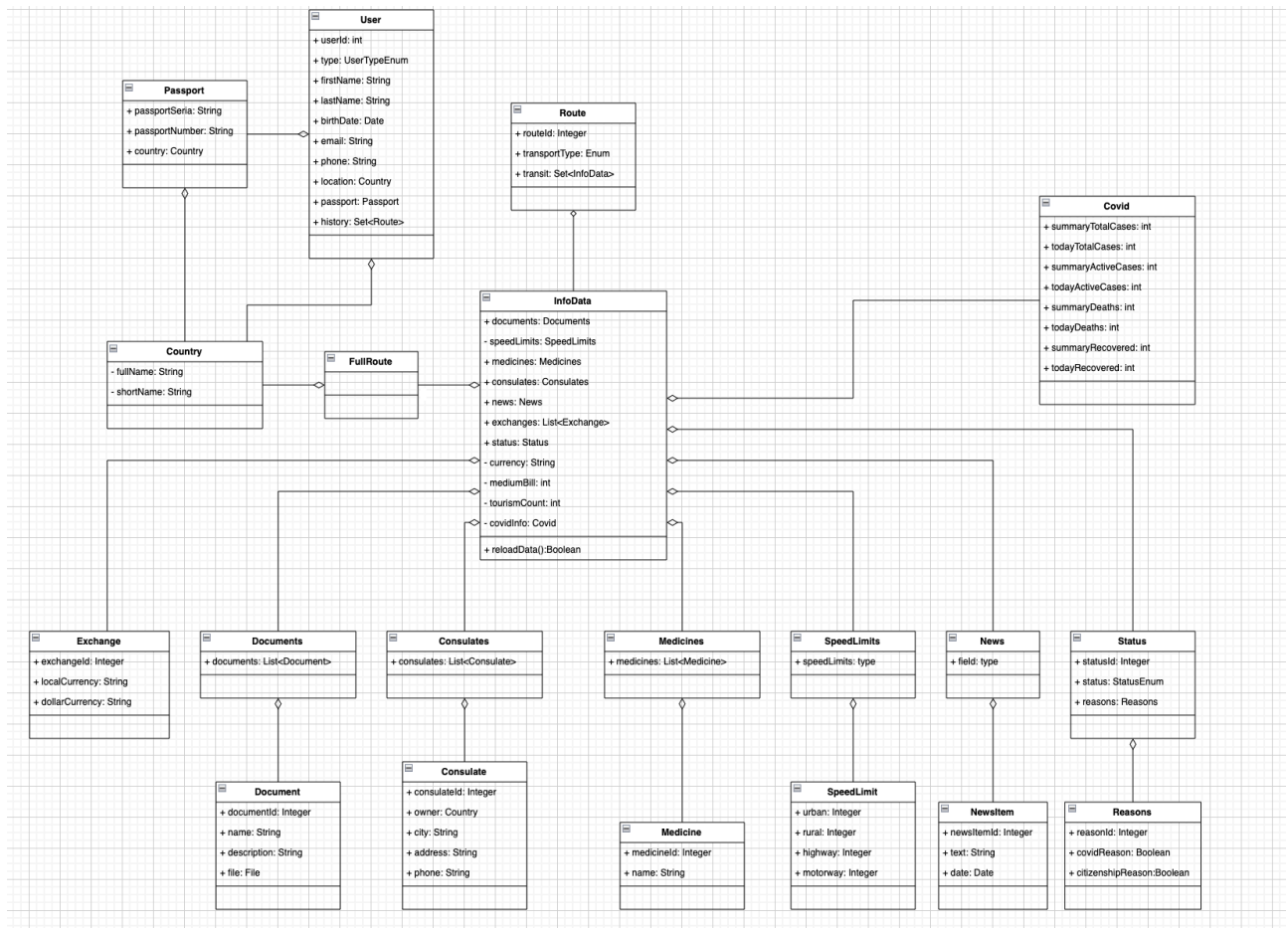


Рисунок 3.2 – Діаграма класів

3.2 Проектування бази даних та її реалізація

Таблиці створеної бази даних нормалізовані, приведені до третьої нормальної форми. У базі даних логічно зв'язані дані зберігаються в декількох різних таблицях. З'єднання зв'язаної інформації здійснюється за допомогою установки між таблицями зв'язків.

Зв'язок – це асоціація, що встановлюється між двома сутностями. Ця асоціація завжди є бінарною і може існувати між двома різними сутностями або між сутністю і її же самій (рекурсивний зв'язок). Як правило, зв'язки виражаються дієсловами або дієслівними поєднаннями. Будь-яка сутність може мати довільну кількість зв'язків з іншою сутністю. При встановленні зв'язку між сутностями одна з них буде головною, а друга – підпорядкованою. У головній сутності завжди доступні записи, що все містяться в ній. У підлеглий же сутності доступні тільки ті записи, у яких значення атрибутів

зовнішнього ключа співпадає із значенням відповідних атрибутів поточного запису головної сутності. Причому зміна поточного запису головної сутності приведе до зміни безлічі доступних записів підлеглої сутності, а зміна поточного запису в підлеглій сутності не викличе ніяких змін ні в одній з таблиць.

Розрізняють чотири типу кардинальності зв'язків між сутностями.

- 1) Один до одного (1:1) – кожному запису однієї сутності відповідає тільки один запис іншої сутності.
- 2) Один до багатьох (1:Б) – одного запису головної сутності можуть відповідати декілька записів підлеглої сутності.
- 3) Багато до одного (Б:1) – декільком записам головної сутності може відповідати один і той же запис підлеглої сутності.
- 4) Багато до багатьох (Б:Б) – один запис головної сутності пов'язаний з декількома записами підпорядкованої, а один запис підлеглої сутності пов'язаний з декількома записами головної.

Якщо між таблицями був встановлений зв'язок, то додатково для неї можна визначити вимоги підтримки цілісності даних. Забезпечення посилальної цілісності є найважливішою вимогою до будь-якого зв'язку, оскільки в цьому випадку гарантується узгодженість даних між двома зв'язаними таблицями. Посиладельна цілісність підтримується за рахунок встановлення між батьківською і дочірньою таблицями зв'язку, здійснюваного через поле, присутнє в обох таблицях.

Між таблицями спроектованої бази даних встановлені зв'язки із забезпеченням цілісності даних; тип зв'язку між таблицями «один до багатьох».

У діаграмах сутність представляється у вигляді прямокутника, що містить ім'я сутності. При цьому ім'я сутності – це ім'я типа, а не деякого конкретного екземпляра цього типа. Для більшої виразності і кращого розуміння ім'я сутності може супроводитися прикладами конкретних об'єктів цього типа.

Атрибутом сутності є будь-яка деталь, яка служить для уточнення, ідентифікації, класифікації, числової характеристики або вираження стану сутності. Атрибути сутності представляються іменниками або відповідними поєднаннями.

Зв'язок представляється у вигляді лінії, що зв'язує дві сутності або ведучі від сутності до неї ж самої. При цьому в місці "стиківки" зв'язку з сутністю використовуються триточковий вхід в прямокутник сутності, якщо для цієї сутності в зв'язку можуть використовуватися багато (many) екземплярів сутності, і одноточечний вхід, якщо в зв'язку може брати участь лише один екземпляр сутності.

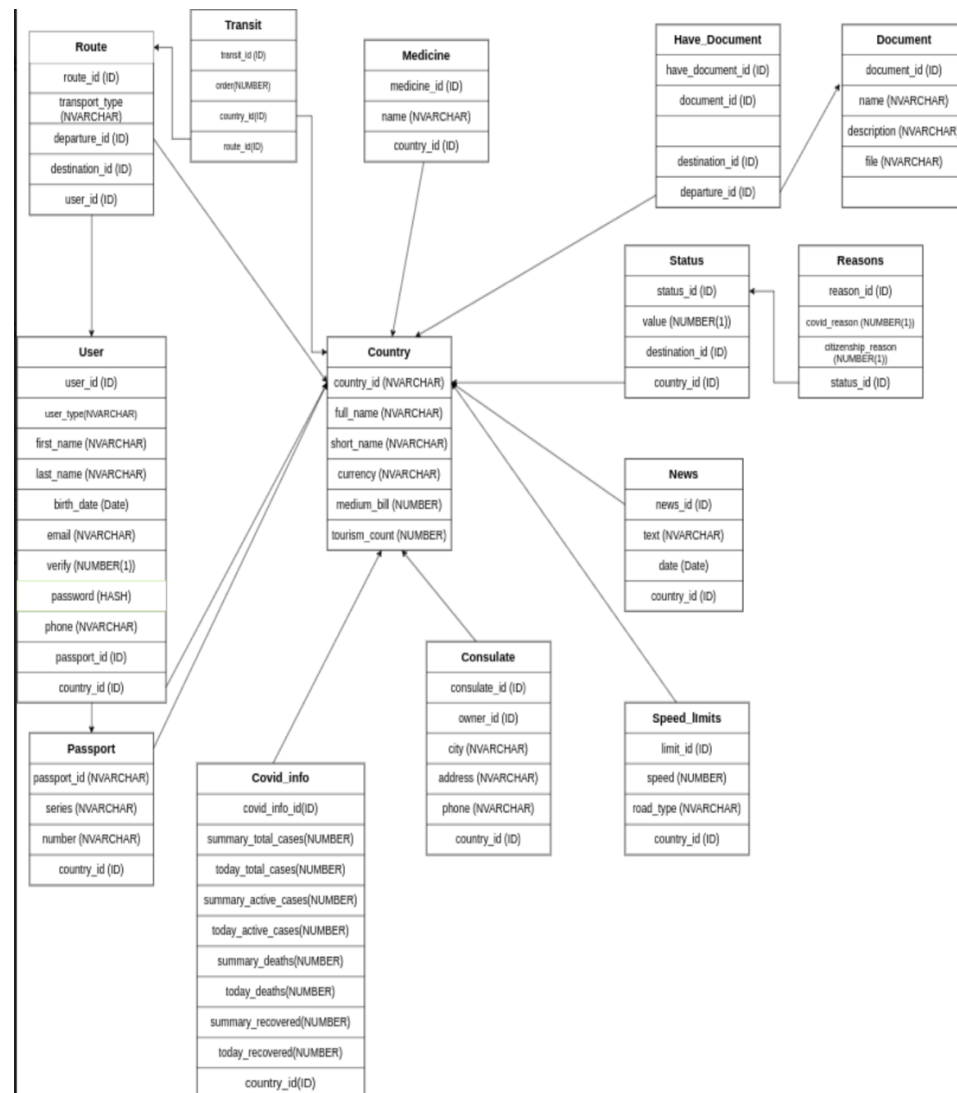


Рисунок 3.3 – схема бази даних

База даних складається з 14 таблиць. Опис основних таблиць наведено нижче.

Таблиця User відповідає за зберігання інформації користувача, вона має зв'язок з таблицями Country та Passport в яких зберігається інформація по країнам і по паспортам. Опис полів приведено у таблиці 3.1.

Таблиця 3.1 – Опис таблиці User (Користувачі)

Ім'я поля	Тип даних	Опис
User_id	ID	Ідентифікатор, первинний ключ
User_type	Varchar	Роль користувача
First_name	Varchar	Ім'я користувача
Last_name	Varchar	Прізвище користувача
Birth_date	Date	Дата народження користувача
Email	varchar	Email та логін користувача
Verify	Number	Статус облікового запису
Password	hash	Пароль від облікового запису
Phone	varchar	Номер телефону користувача
PassporId	ID	Зовнішній ключ
CountryId	ID	Зовнішній ключ

Таблиця Country є однією з основних таблиць, з нею пов'язані майже всі таблиці. Таблиця створена для зручного зберігання всіх країн, а також основної інформації про країну. В таблиці є поля Medium_bill (середня вартість) та Tourism_count (Лічильник туристів), вони оновлюються раз у тиждень. Поля таблиці представлені у таблиці 3.2.

Таблиця 3.2 – Опис таблиці Country (Країни)

Ім'я поля	Тип даних	Опис
Country_id	ID	Ідентифікатор, первинний ключ

Full_name	Varchar	Повна назва країни
Short_name	Varchar	Скорочена назва
Currency	Varchar	Валюта в країні
Medium_bill	Money	Середня вартість
Tourism_count	Number	Лічильник туристів

Таблиця Document(Документи) містить в собі інформацію про документи потрібні для заповнення, поле file містить в собі посилання на потрібний файл. Нижче приведено опис в таблиці 3.3.

Таблиця 3.3 – Опис таблиці Document(Документи)

Ім'я поля	Тип даних	Опис
Document_Id	ID	Ідентифікатор, первинний ключ
Name	Varchar	Ім'я файлу
Description	Varchar	Опис файлу
File	Varchar	Посилання на файл

4 ОПИС ПРОГРАМНИХ ЕЛЕМЕНТІВ

4.1 Огляд інтерфейсу з боку гостя

Розглянемо інтерфейс з боку гостя, починаючи з входу на сторінку.

4.1.1 Головна сторінка веб-сайту.

На рис 4.1 зображено головну сторінку веб-додатку, на якому зображено головну сторінку веб-додатку. На ній ми бачимо інтерактивну карту, вікно пошуку і іконку користувача.



Рисунок 4.1 – Головна сторінка додатку

При натисканні на одну з країн, або введення назви у вікно пошуку можна отримати інформацію про країну, але для отримання інформації потрібно спершу авторизуватися в іншому випадку буде повідомлення про те що ви не авторизований. Як зображено на рисунку 4.2.

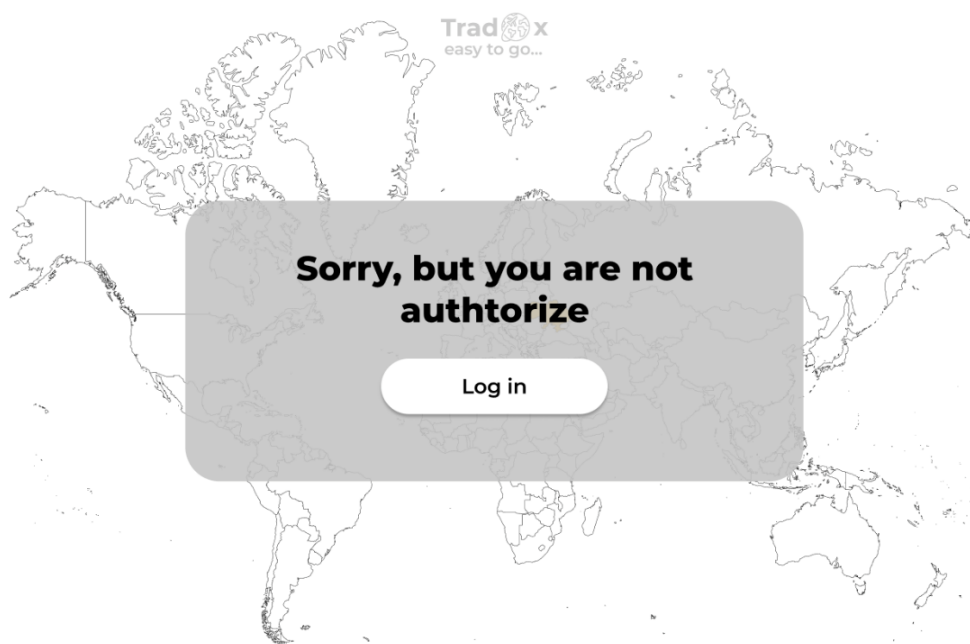


Рисунок 4.2 – Повідомлення що потрібно авторизуватися

Як бачимо, при спробі отримати інформацію про країну без авторизації з'являється вікно з повідомленням що ви не авторизовані і кнопкою авторизації, також зникає вікно пошуку і мапа перестає реагувати на курсор користувача. При натисканні кнопки “Log In” ви переходите на нову сторінку з авторизацією як зображено на рисунку 4.3.

4.1.2 Сторінка авторизації.

На сторінці авторизації ви можете авторизуватися під існуючим обліковим записом або натиснути кнопку «Register» і вас переправить на сторінку реєстрації. Також якщо ви забули свій пароль, у вас є можливість його відновити. При натисканні на кнопку «forgot password» вам на електрону пошту прийде повідомлення з посиланням на відновлення паролю, який згенерується автоматично, а потім ви зможете змінити його у особистому кабінеті користувача.



Auththorization

E-mail

Password

[Forgot password?](#)
[Register](#)

[Log in](#)

Рисунок 4.3 – Сторінка авторизації

На сторінці відновлення паролю потрібно ввести свою електронну адресу і натиснути кнопку «Send Link». Приклад зображено на рис 4.4.

Password Reset

We will send you reset link

E-mail

[Send link](#)

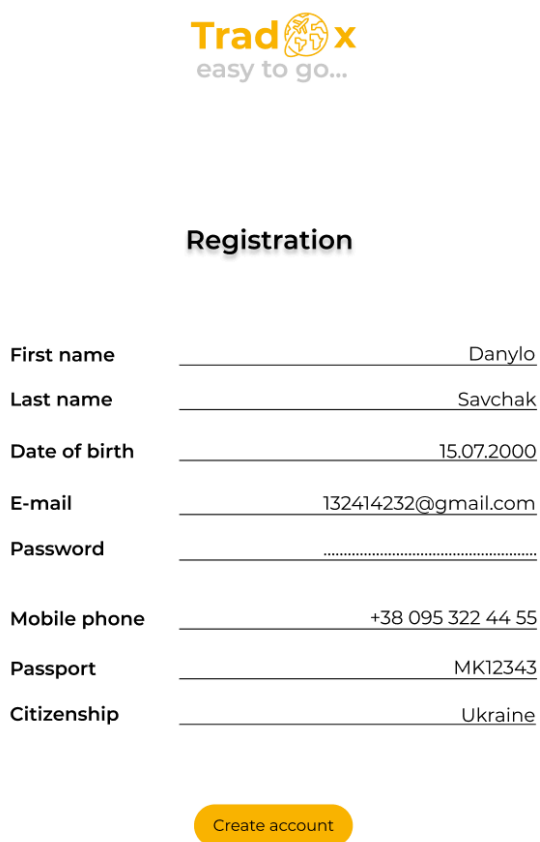
Рисунок 4.4 – відновлення паролю

При натисканні на кнопку «Register» нас переправляє на сторінку реєстрації де нам необхідно заповнити обов'язкові поля для реєстрації:

- Імя

- Прізвище
- Дату народження
- Електронну адресу
- Пароль
- Мобільний телефон
- Код паспорту
- Країну проживання

Приклад продемонстровано на рисунку 4.5.



Tradex
easy to go...

Registration

First name	Danylo
Last name	Savchak
Date of birth	15.07.2000
E-mail	132414232@gmail.com
Password	
Mobile phone	+38 095 322 44 55
Passport	MK12343
Citizenship	Ukraine

Create account

Рисунок 4.5 – форма реєстрації

4.1.3 Інформація про країну.

Після авторизації ми повертаємося на головну сторінку де після натискання на одну з країн або при пошуку отримуємо інформацію про країну:

- Інформація по ковіду

- Необхідні для в'їзду документи
- Посольства вашої держави.
- Курс валюти
- Останні новини в країні

Також можна обрати тип поїздки, на вибір є 3 типи засобів пересування для яких подалі можна буде отримати необхідні документи:

- Літак
- Поїзд
- Авто

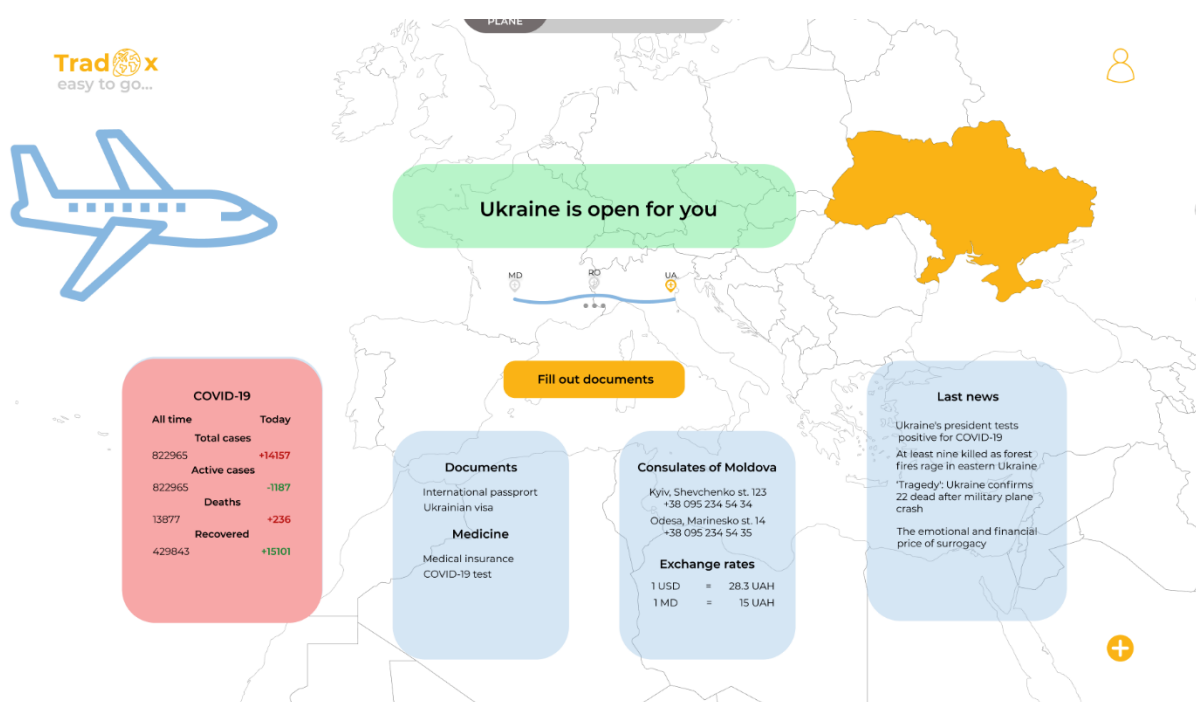


Рисунок 4.6 – Інформація про країни

Також після натискання на країну може виявитися що вона по різноманітним причинам закрыта для користувача. Для перевірки відкритості країни з вашого облікового запису береться місце вашого проживання та ваш паспортний код, який перевіряється через спеціальне API. На даний момент підтримуються 2 типи причин закритої країни:

- Пандемічний стан

- Заборона на в'їзд по законодавчим причинам

Приклад повідомлення закритої країни зображено на рисунку 4.7.



Рисунок 4.7 – інформація про закриту країну

4.1.4 Друк необхідних документів

При натисканні на кнопку “Fill out documents” можна отримати основні документи для подорожі у вибрану країну. Документ формується на основі ваших паспортних даних, які відправляються в спеціальний сервіс і на основі них отримуються необхідні документи які в подальшому заповнюються за допомогою програмного коду на серверній частині ПП. Після чого ви можете отримати документ в PDF форматі. Приклад сформованого документу зображено на рисунку 4.8.

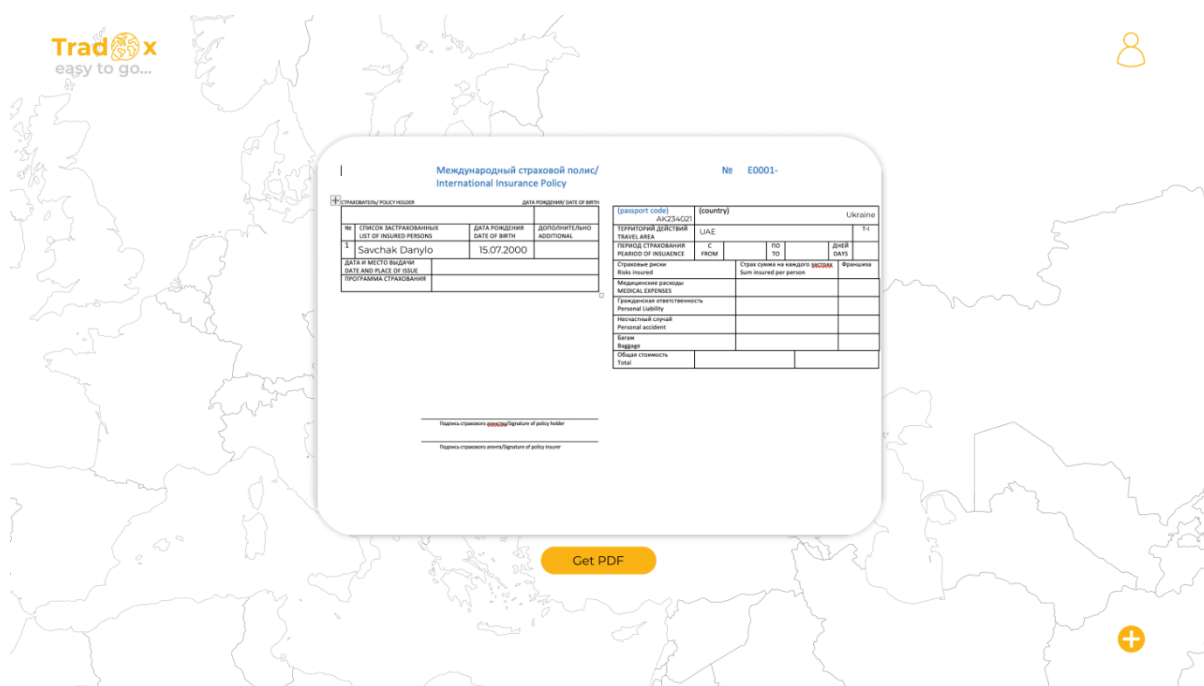


Рисунок 4.8 – Роздруківка документів для подорожі

4.2 Опис інтерфейсу користувача

4.2.1 Сторінка користувача

З головної сторінки через меню можна перейти на сторінку користувача. На якій ми можемо бачити персональну інформацію користувача. Цю інформацію можна змінити, також є змога змінити пароль користувача. Нижче, під інформацією користувача відображено останні подорожі користувача з якими доступні наступну дії:

- Видалення
- Перегляд останнього маршруту

Також при натисканні на іконку над написом «Personal info» з'являється вікно де ви можете вийти з облікового запису або видалити обліковий запис. Видалення облікового запису виконується через підтвердження по електронній пошті.

Сторінку користувача продемонстровано на рисунку 4.9



Personal info

First name	<u>Danylo</u>	Mobile phone	<u>+38 095 322 44 55</u>
Last name	<u>Savchak</u>	Passport	<u>MK12343</u>
Date of birth	<u>15.07.2000</u>	Citizenship	<u>Ukraine</u>
E-mail	<u>132414232@gmail.com</u>	Current country	<u>Moldova</u>

Save

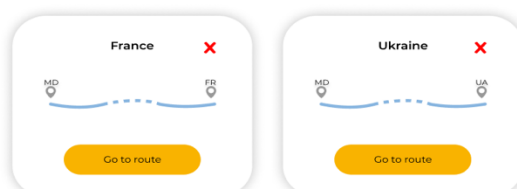


Рисунок 4.9 – Сторінка користувача

4.3 Опис інтерфейсу адміністратора

Також із головної сторінки можна перейти на сторінку адміністратора якщо авторизуватися під спеціальним аккаунтом. На сторінці є панель з вибором даних які треба переглянути, а також таблиця з даними які є у базі даних. На сторінці адміністратор може редагувати або додавати:

- Країни
- Документи
- Статуси
- Консульства

Також адміністратор може видавати права користувачам, а також обмежувати доступ деяким користувачам. Хоча у адміністратора теж є обмеження у внесенні даних до системи. Тому що значна частина даних береться з зовнішніх сервісів, тому їх редагування не є доцільним. А деякі данні є статичними і тому не можуть бути редаговані. Також на сторінці є кнопки для додавання зберігання та видалення даних, а також вікно пошуку даних у таблиці. В таблиці передбачена здатність вибору декількох записів одночасно для їх видалення або редагування. Кожне

поле можна відсортувати по алфавітному порядку. На наступному рисунку зображено панель адміністратора.

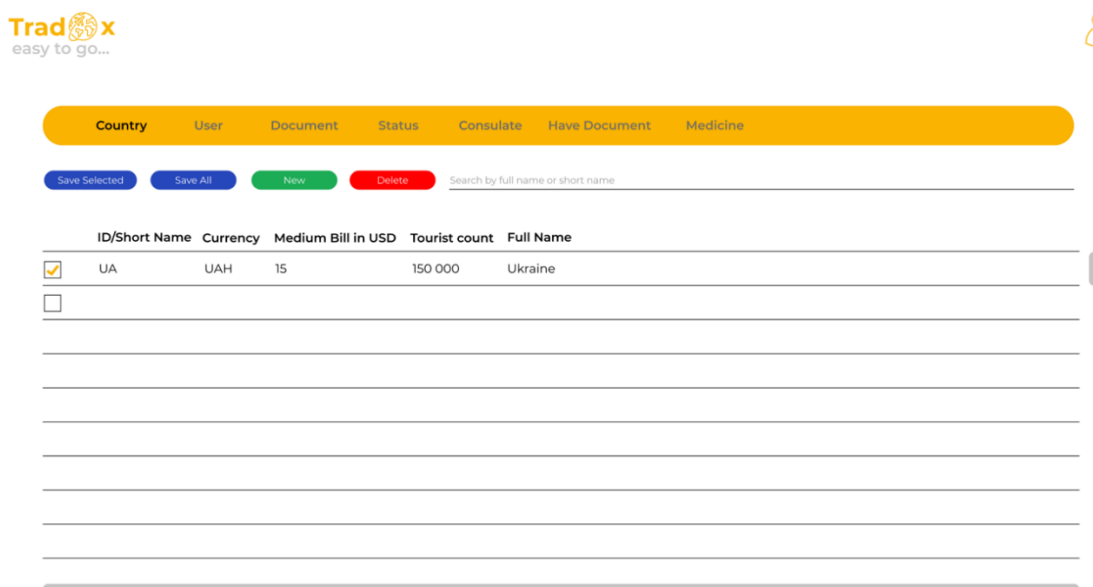


Рисунок 4.10 – панель адміністратора

Адміністратор не може додавати нову країну, або видаляти чи редагувати інформацію користувачів, деяка інформація додається з API. На наступному рисунку зображено редагування інформації про країну.

The screenshot shows a modal form titled 'New Country'. The form has the following fields and values: 'Full name' with the value 'Ukraine', 'Short name' with the value 'UA', 'Currency' with the value 'UAH', 'Medium bill in USD' with the value '145', and 'Tourist count' with the value '150000'. At the bottom of the form are two buttons: 'Create' (yellow) and 'Cancel' (black).

Рисунок 4.11 – Редагування інформації про країну

ВИСНОВОК

У кваліфікаційній роботі було розглянуто питання, щодо розробки застосувань інформаційно-аналітичної системи підтримки міжнародних мандрівок. Система, представлена в дипломній роботі, є ключовою ланкою в автоматизації роботи туристичної агенції. Рішення є комплексним, дякуючи чому, значно знижуються витрати на інтеграцію і навчання користувачів системи, в порівнянні з розрізненим середовищем. За допомогою подібної системи, користувачі можуть витратити значно менше часу на роботу з даними в інформаційній системі, будь-то заповнення нових позицій або обробка замовлень клієнтів.

Також у результаті написання дипломного проекту було розроблено програмний продукт, який призначений для планування подорожей.

Розробка програмного продукту перш за все вимагала розглядання існуючих аналогів на ринку для аналізу поточного стану проблем предметної області. Інтерфейс програми оптимізований для роботи з великими потоками інформації. Після побудування порівняльної таблиці аналогів були визначені функції, які необхідно було реалізувати у програмі.

В процесі дослідження були вивчені етапи та проблеми створення додатків та проблеми індустрії подорожей.

Для досягнення поставленої мети в роботі було реалізовані наступні задачі:

- досліджено предметну область;
- проаналізовано основні проблеми предметної області;
- проведено аналіз існуючих аналогів;
- проведено аналіз існуючих статей про подорожі та її проблеми.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Установка JDK URL: <https://khasang.io/courses/java0/lectures/1214779>
(дата звертання 17.01.2021)
2. MySql – система управління бази даних – « Веб Креатор». URL:
<https://web-creator.ru/articles/mysql> (дата звернення 25.02.2021).
3. СУБД Oracle. URL (дата звернення 25.02.2021).:
<https://ru.wikipedia.org/wiki/Oracle>
4. Дэвид А. Марка, Клемент Л. МакГоуэн Методология структурного анализа и проектирования SADT. – М.: 1993. – 242 с.
5. Java програмна платформа – Вікіпедія.
URL: [https://ru.wikipedia.org/wiki/ Java_\(программная_платформа\)](https://ru.wikipedia.org/wiki/Java_(программная_платформа)) (дата звернення 2.02.2021).
6. Java Runtime Environment (сокр. JRE; укр. середовище виконання для Java) – Вікіпедія. URL:
https://ru.wikipedia.org/wiki/Java_Runtime_Environment (дата звернення 2.02.2021).
7. Data access objects (сокр. DAO). URL :
ru.wikipedia.org/wiki/Data_Access_Object (дата звернення 15.03.2021)
8. Eclipse – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Eclipse> (дата звернення 10.03.2021).
9. NetBeans – Википедия URL:
<https://ru.wikipedia.org/wiki/NetBeans> (дата звернення 25.02.2021).
10. IntelliJ Idea – Вікіпедія. URL:
https://uk.wikipedia.org/wiki/IntelliJ_IDEA (дата звернення 10.03.2021)
11. Spring Framework – Вікіпедія. URL:
https://ru.wikipedia.org/wiki/Spring_Framework (дата звернення 25.03.2021).
12. Angular Framework – Вікіпедія. URL:

https://ru.wikipedia.org/wiki/Angular_Framework (дата звернення 25.02.2021).

13. TypeScript – Вікіпедія. URL: <https://ru.wikipedia.org/wiki/TypeScript> (дата звернення 26.03.2021).

14. Python огляд мови програмування. URL: <https://pythonworld.ru>

15. React.js – офіційний сайт. URL: <https://reactjs.org>