

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних систем

«До захисту в ЕК»

Директор інституту(декан
факультету)

_____ Форсюк А.В.
(підпис) (прізвище та ініціали)

«___» _____ 20__р.

«До захисту допущено»

Завідувач кафедри

_____ Чумаченко С.М.
(підпис) (прізвище та ініціали)

«___» _____ 20__р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми «Комп'ютерні науки»

на тему: «Розроблення IOS додатку путівника по Вінниці»

Виконав: здобувач 4 курсу, групи 5

_____ Мельниченко Роман Васильович
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник _____ Мазуренко Ольга Олександрівна
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти _____ Мазуренко О.О.
(прізвище та ініціали) (підпис)

_____ (прізвище та ініціали) (підпис)

_____ (прізвище та ініціали) (підпис)

Рецензент _____
(прізвище та ініціали) (підпис)

Засвідчую, що в цій
кваліфікаційній роботі немає
запозичень із праць інших авторів
без відповідних посилань.

Здобувач _____
(підпис)

Київ - 20__р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизація і комп'ютерних систем

Кафедра Інформаційних систем

Освітній ступінь Бакалавр

Спеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітньо-професійна програма «Комп'ютерні науки»

(назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інформаційних систем

С.М.Чумаченко

“___” _____ 20__ року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Мельниченко Роману Васильовичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення IOS додатку путівника по Вінниці»

керівник роботи Мазуренко Ольга Олександрівна, к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “29” квітня 2021 року №248-кв

2. Строк подання здобувачем роботи 03.06.2021 року

3. Вихідні дані до роботи Туристичні місця, маршрути

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) системний аналіз предметної області, функціональні моделі «Процесу знаходження інформації про туристичну пам'ятку», постановка задачі, логічна та фізична модель бази даних, база даних, опис процесу розроблення, опис реалізованих функцій, інструкція користувача, висновки

5. Перелік графічного матеріалу

функціональні моделі «AS-IS» та «TO-BE», логічної та фізичної моделі бази даних, згенерована модель бази даних, знімки екрану мобільного додатку, фрагменти

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	К.Т.Н. Мазуренко О.О.		
2	К.Т.Н. Мазуренко О.О.		
3	К.Т.Н. Мазуренко О.О.		

7. Дата видачі завдання 29 квітня 2021 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Перший етап – системний аналіз процесу знаходження інформації про пам'ятки Вінниці	05.05-15.05	Виконано
2	Другий етап – реалізація задач автоматизації знаходження інформації про пам'ятки Вінниці	16.05-20.05	Виконано
3	Третій етап – опис охорони праці	20.05-23.05	Виконано
4	Четвертий етап – оформлення пояснювальної записки та створення презентації	24.05-28.05	Виконано

Здобувач _____
(підпис)

Мельниченко Р.В.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Мазуренко О.О.
(прізвище та ініціали)

АНОТАЦІЯ

Головною метою даної бакалаврської роботи є розробка IOS додатку путівника по місту Вінниця, для спрощення туристам дізнаватись про історію міста, навігацію до пам'яток. Мобільний додаток допоможе вибрати та прокласти маршрут до потрібного місця.

Для реалізації мобільного додатку розроблено схеми декомпозиції процесу знаходження інформації пам'яток, структуру бази даних. Також спроектовано інтуїтивно-зрозумілий інтерфейс.

Об'єктом дослідження є процес знаходження інформації пам'яток Вінниці.

Предметом дослідження є автоматизація процесу знаходження інформації пам'яток та навігації до них через реалізацію електронного путівника.

Бакалаврська робота містить 79 сторінок, 10 таблиць, 18 рисунків, 6 додатків і 12 літературних джерел.

КЛЮЧОВІ СЛОВА: ЕЛЕКТРОННИЙ ПУТІВНИК, ПАМ'ЯТКИ МІСТА. НАВІГАЦІЯ, БАЗА ДАНИХ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ANNOTATION

The main purpose of this bachelor's thesis is to develop an IOS application for a guide to the city of Vinnytsia, to make it easier for tourists to learn about the history of the city, navigation to attractions. The mobile application will help you choose and route to the right place.

For the implementation of the mobile application, decomposition schemes of the process of finding information of monuments, the structure of the database have been developed. An intuitive interface is also designed.

The object of research is the process of finding information about the monuments of Vinnytsia.

The subject of the research is the automation of the process of finding information of monuments and navigation to them through the implementation of an electronic guide.

The bachelor's thesis contains 79 pages, 10 tables, 30 figures, 7 appendices and 17 references.

KEY WORDS: ELECTRONIC GUIDE, CITY SIGHTS. NAVIGATION, DATABASE, SOFTWARE.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРОЦЕСУ ПОШУКУ ІНФОРМАЦІЇ ПРО ПАМ'ЯТКИ ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ.....	9
1.1 Опис та особливості процесу функціонування туризму.....	9
1.2 Туризм у Вінниці.....	9
1.3 Аналіз інформаційних можливостей пошуку інформації по Вінниці.....	10
1.4 Розроблення функціональної моделі додатку путівника.....	11
1.4.1 Опис методології IDEF0	11
1.4.2 Опис функціональних схем (AS-IS)	11
1.5 Виявлені проблеми при знаходженні інформації про пам'ятки.....	14
1.5.1 Виявлені недоліки	14
1.5.2 Задачі автоматизації	14
1.6. Огляд існуючих систем-аналогів для пошуку інформації про пам'ятки	15
1.6.1 Паперовий путівник «Місто».....	15
1.6.2 Сайт «МоеMisto.ua»	15
1.6.3 Порівняння систем-аналогів.....	16
1.7 Обґрунтування доцільності проектування й розроблення системи для функціонування туризму у місті Вінниця	17
1.8 Концептуальна модель системи	17
1.9 Постановка задачі на проектування.....	19
1.9.1 Призначення та цілі створення системи.....	19
1.9.2 Вимоги до створюваної системи.....	19
1.9.3 Функції, які має виконувати система	19
1.9.4 Вхідні та вихідні дані системи	20
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ.....	21
2.1 Загальні положення	21

2.2. Призначення і цілі створення системи	21
2.3. Характеристика об'єкта автоматизації	21
2.4. Вимоги до системи	22
2.5. Склад і зміст робіт по створенню системи.....	30
2.6. Порядок контролю і приймання додатку	31
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію	31
2.8. Вимоги до документації.....	31
2.9. Джерела розробки.....	31
РОЗДІЛ 3. РОЗРОБЛЕННЯ КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ	33
3.1 Обґрунтування вибору засобів розроблення системи.....	33
3.2 Проектування БД.....	37
3.3 Створення інтерфейсу користувача	38
3.4 Інструкція користувача	51
3.5 Техніко-економічне обґрунтування	58
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	65
4.1 Безпека праці.....	65
4.2 Умови праці.....	67
4.3 Ергономіка робочого місця.....	67
4.4 Пожежна безпека	79
ВИСНОВКИ	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	82
ДОДАТКИ	83
ДОДАТОК А «ФУНКЦІОНАЛЬНА МОДЕЛЬ (AS-IS)»	83
ДОДАТОК Б «ФУНКЦІОНАЛЬНА МОДЕЛЬ (TO-BE)».....	85
ДОДАТОК В «ЛОГІЧНА СХЕМА БАЗИ ДАНИХ»	87
ДОДАТОК Г «ФІЗИЧНА СХЕМА БАЗИ ДАНИХ».....	88
ДОДАТОК Д «ЗГЕНЕРОВАНА БАЗА ДАНИХ У COREDATA».....	89
ДОДАТОК Є «ФРАГМЕНТИ КОДУ ПРОГРАМИ».....	92

ВСТУП

Індустрія сучасного світового туризму – це одна з найбільш прогресивних галузей у світі. Розглядається вона як окрема частина, так і як комплекс галузей.

На сьогодні галузь туризму увійшла у життя кожної людини на планеті. Навіть на початку XXI ст. туристична сфера увійшла до найбільш прибуткових галузей світу.

З отриманням Україною незалежності туристичний ринок нашої держави тільки починав розвиватися. Сьогодні – це одна з найперспективніших сфер економічного розвитку країни.

Сучасна людина робить все для того щоб досягти максимального комфорту. Одним з бажань більшості людей є доступ до інформації у своєму телефоні. Саме тому величезною актуальністю користується така послуга, як розробка мобільних додатків під ОС iOS. Все це стало актуальним разом з появою мобільного Інтернету, бажанням бути проінформованим, завжди мати доступ до інформації, яка може стати у пригоді. Під час поїздок завжди є можливість відкрити у мобільному телефоні інформацію (туристичні місця, розклад громадського транспорту і тд.), яка може бути корисна користувачу. Але відразу ж варто відзначити, що без спеціальних додатків навряд чи б була досягнута необхідна можливість. Без них не обійтись і при вирішенні таких завдань, як подорож у незнайоме місто.

Процес туризму зараз проходить також здебільшого з використанням мобільних пристроїв, зручними є путівники по містам, що допомагають швидко знайти потрібне місце і більше дізнатись про нього.

На даний момент сфера туризму є дуже актуальною, навіть попри сучасні реалії з COVID-19. І тому виникає потреба у спрощенні процесу подорожі, особливо в межах своєї країни.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРОЦЕСУ ПОШУКУ ІНФОРМАЦІЇ ПРО ПАМ'ЯТКИ ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ.

1.1 Опис та особливості процесу функціонування туризму

У процесі подорожі все частіше виникає потреба у різних допоміжних сервісах, наприклад для навігації по не знайомому місті, перегляду інформації про певну пам'ятку, її фото. До недавнього часу ці проблеми туризму було важко вирішити. Сучасний же ринок програмного забезпечення пропонує для автоматизації ринку туризму різні програми навігатори, путівники, інформаційні джерела.

Електронні туристичні путівники – програмні засоби туристичного призначення, у яких відображена певна інформація про певне місто або міста, у тій або іншій мірі реалізована технологія навігації, забезпечені умови для реалізації різних видів туристичної діяльності.

На сьогодні подорожі вже не є привілеєм людей, які мають багато коштів, щоб дозволити собі відпочинок закордоном. Інтернет-технології допомагають у різних процесах подорожі та допомагають застосування її у практиці. Вони надають такі можливості:

- Отримувати інформацію про певну пам'ятку;
- Знаходити певні пам'ятки, через навігацію;
- Отримувати графік роботи музею, вартість входу.

Комп'ютеризація туризму та розвиток мережі Інтернет спонукають турагенства до розроблення електронних засобів туризму. Зокрема, для місцевого туризму, навігації по місту та перегляд визначних місць, адже потреба в таких додатках збільшується з кожним роком.

1.2 Туризм у Вінниці

На даний момент у місті Вінниця сфера туризму поділяється на такі категорії (Рис. 1). У місті є багато цікавих місць, які можна відвідати, є і такі

відомі місця, які входять в ТОП-10 у світі. Є і унікальні пам'ятки, аналогів яким немає у всьому світі. Тому перспективи розвитку цієї сфери дуже великі.

Туризм у місті знаходиться в процесі розвитку, тому на даний момент кількість відвідувачів міста не є високою, але спостерігається певний ріст бажаючих ближче познайомитись з містом та його історією. Сучасний турист завжди продумує свій план подорожі, тож люди які навідуються у місто шукаю різні засоби, які допоможуть в цьому процесі.

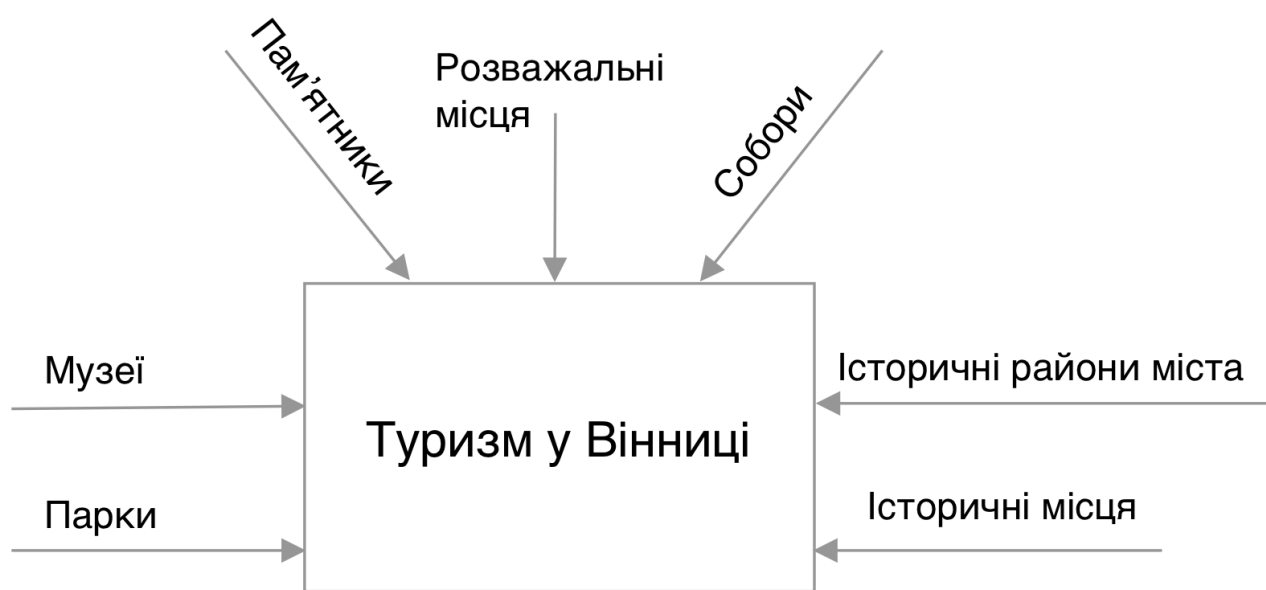


Рис. 1 Туристична схема Вінниці

1.3 Аналіз інформаційних можливостей пошуку інформації по Вінниці

У даний час процес подорожі по місту займає тривалий час, оскільки доступних засобів дуже мало. Зазвичай використовуються дані з різних джерел для пошуку потрібних місць та інформації про них. Об'єм матеріалів може бути великим, що також значиться на складності самого процесу .

Для навігації по місту до визначних місць використовують звичайний додаток google maps.

Весь процес проходить занадто складно, так як в додатку google можна провести тільки маршрут до місця яке ви повинні знати, або знати його адресу, також не відображається інформація про визначне місце.

Туристи використовують різні способи щоб зібрати всю інформацію в одне ціле, наприклад інформацію про туристичні місця вони беруть з таких джерел як google пошук, паперові буклети. Підготовка до подорожі займає багато часу, особливо якщо турист збирається відвідувати багато місць. Є також готові екскурсійні програми, які спрощують підготовку, всі вони в паперовому вигляді, вони будуть коштувати приблизно від 150грн до 400грн, але щоб дістатись до певного місця, все одно потрібно скористуватись додатковими засобами навігації.

1.4 Розроблення функціональної моделі додатку путівника

Для виявлення наявних недоліків у процесі функціонування туризму по місту Вінниця потрібно провести аналіз процесу у такому вигляді, як він зараз існує. Для цієї задачі створюється модель бізнес процесів, моделі AS-IS за допомогою методу IDEF0.

1.4.1 Опис методології IDEF0

Функціональна модель IDEF0 являє собою набір блоків, кожен з яких представляє собою «чорний ящик» з входами і виходами, управлінням та механізмами, які деталізуються (декомпонуються) до необхідного рівня. Найбільш важлива функція розташована у верхньому лівому кутку. А з'єднуються функції між собою за допомогою стрілок і описів функціональних блоків. При цьому кожен вид стрілки або активності має власне значення. Дана модель дозволяє описати всі основні види процесів, як адміністративні, так і організаційні. Стрілки можуть бути:

- Вхідні - вступні, які ставлять певне завдання. Вихідні - виводять результат діяльності;
- Керуючі (зверху вниз) - механізми управління (положення, інструкції тощо).
- Механізми (від низу до верху) - що використовується для того, щоб зробити необхідну роботу.

Вхідні та вихідні стрілки точніше було б називати такими, що вводять і виводять, так як по-англійськи вони називаються Input і Output відповідно. Але особливості перекладу і звичні назви виглядають вже так, як склалося. І все ж для правильного розуміння термінів важливо пам'ятати їх значення в даному випадку. Це підтверджується ще і тим, що дана нотація створена насамперед для розробки ПО, і терміни переводити правильніше в цій точки зору. Стрілки підписуються за допомогою іменників (досвід, план, правила), а блоки - за допомогою дієслів, тобто в них описуються дії, які виробляються (створити товар, укласти договір, провести відвантаження). IDEF0 - це дуже простий і одночасно наочний мова опису бізнес-процесів. За допомогою цього стандарту можлива передача інформації між розробниками, консультантами та користувачами. Стандарт дуже ретельно розроблявся, він зручний для проектування, універсальний. Для роботи з ним існує безліч інструментів, наприклад, VISIO, BPWIN, ERWIN, Bussines studio і т.д. Крім того, використання для створення бізнес-моделей IDEF0 - це не тільки зручно, це ще й правильно. Цей інструмент був розроблений для бізнес-аналітики, він пройшов тривалу і ретельну налагодження і шліфування. За допомогою IDEF0 створити функціональну модель без помилок набагато простіше, ніж без застосування цього стандарту. Як відомо, забивати цвяхи найкраще молотком. Звичайно, ви можете для цього застосовувати і інші інструменти, але молоток - найбільш функціональний і з його допомогою найпростіше забити цвях акуратно і точно. Так і з застосуванням IDEF0 - цей інструмент був створений для функціонального моделювання, і з його допомогою ви набагато швидше і точніше зможете отримати потрібний результат.

1.4.2 Опис функціональних схем (AS-IS)

Розроблена функціональна модель описує процеси функціонування туризму у місті Вінниця в тому вигляді, як вони здійснюються зараз (AS-IS). Контекстна діаграма наведена на рис.1 у додатку А.

Входи:

- Буклети – матеріали, з яких береться інформація про певні пам’ятки.
- Сайти путівники – сайти, через які можливо переглядати інформацію про пам’ятки і історичні факти про них.
- Туристичні агентства – агентства, у яких можливо дізнатися про наявні у них екскурсійні програми та при бажанні придбати.
- Екскурсійні агентства – агентства, у яких можливо придбати екскурсію містом або буклет з пам’ятками.
- Пошукові Web-системи – пошук інформації про наявні у певний період часу розважальні заходи, більш детальну інформацію про пам’ятки.

Виходи:

- Інформація про пам’ятку.
- Маршрут.
- Посилання на сайт – сайт, на якому знаходиться потрібна інформація.

Механізми:

- Користувач.
- Персональний комп’ютер.
- Канцелярські товари.
- Транспорт.
- Мобільний пристрій.

Декомпована модель наведена на рис.2 у додатку А.

Дана модель показує трирівневу ієрархію взаємозв’язаних і упорядкованих діаграм, відображає існуючий порядок бізнес-процесів та має статус «AS-IS»– «як є».

На першому рівні декомпозиції діаграма складається з трьох блоків:

- Пошук інформації
- Пошук маршруту

- Упорядкування даних

Як можна побачити з наведених вище діаграм і їх описів, користувач (механізм) працює з багатьма матеріалами (вхід) для пошуку потрібної інформації.

Огляд існуючих процесів за допомогою моделі AS-IS здійснюється для того, щоб наочно переглянути всі недоліки та зробити висновки виходячи з цього.

1.5 Виявлені проблеми при знаходженні інформації про пам'ятки

Через велику кількість джерел з яких потрібно взяти інформацію, процес пошуку займає велику кількість часу. Крім того деякі джерела у паперовому вигляді, що не зовсім зручно при обробці цієї інформації.

1.5.1 Виявлені недоліки

Отже, за наявного стану процесу пошуку виявлено наступні проблеми:

- Оскільки процес знаходження інформації проходить з використанням паперових матеріалів, уповільнюється пошук потрібних даних;
- Немає можливості швидко упорядкувати дані;
- Є велика кількість джерел, з яких береться потрібна інформація, через що збільшується час на пошук і на об'єднання всіх джерел.

Тому нинішній стан процесу пошуку інформації про пам'ятки по місту Вінниця є незадовільним, оскільки важлива частина процесу є занадто складною, а саме пошук місць та знаходження інформації про них; займає багато часу; та вся наявна інформація знаходиться в різних джерелах і не завжди у повному обсязі, що ускладнює її перегляд та знаходження.

1.5.2 Задачі автоматизації

Було виділено такі задачі автоматизації:

- Пошук інформації про пам'ятки та інформації про неї;
- Пошук можливостей отримати маршрут до пам'яток.

1.6.Огляд існуючих систем-аналогів для пошуку інформації про пам'ятки

Для туристичних цілей у місті Вінниця, є певна кількість сайтів та інших онлайн-ресурсів. Саме путівники реалізовані переважно у паперовому вигляді, електронними є переважно додатки для навігації та сайти на яких можна знайти інформацію про певне історичне місце.

1.6.1 Паперовий путівник «Місто»

Даний засіб доступний у паперовому вигляді, у ньому представлені пам'ятки міста та їх місцезнаходження.

Основні функції путівника:

- Є туристичні пам'ятки міста;
- Є інформація про пам'ятки;

Недоліки путівника:

- Застарілі фото;
- Паперовий вигляд;
- Відображена мала кількість туристичних місць;
- Немає можливості навігації;

1.6.2 Сайт «MoeMisto.ua»

MoeMisto.ua - сайт, багатоцільовий сайт, у якому викладено достатня кількість інформації про місто.

Основні функції сайту:

- Інформаційний портал міста;
- Можливість перегляду афіш подій;
- блог з достатньою кількістю пам'яток міста;

Недоліки засобу:

- Немає пошуку по пам'яткам;
 - Немає навігації до обраних пам'яток;
 - Багато інформації яка не пов'язана з туризмом;
 - Сайт більшою частиною призначений не для туристів для мешканців цього міста.
- Присутня велика кількість реклами;

1.6.3 Порівняння систем-аналогів

Таблиця 1.1 Порівняння систем-аналогів

Системи Функції	Паперовий путівник «Місто»	Сайт «МоеMisto.ua»
Пошук визначних місць	-	-
Навігація до пам'яток	-	-
Інформація про пам'ятки	+	+
Інтерактивна карта з пам'ятками	-	-
Актуальність інформації	+	+
Можливість відкрити у браузері телефона	-	+
Зручний та простий інтерфейс	-	+
Показ реклами	-	+
Витрати	150-350грн	Безкоштовно

Висновок: після дослідження двох путівників: паперовий путівник «Місто» та сайт «МоеMisto.ua», бачимо, що є вибір ресурсів для пошуку туристичних місць, з достатнім об'ємом функцій. Але всі ці електронні засоби розроблені як сайти, або ж представлені у паперовому вигляді.

1.7 Обґрунтування доцільності проектування й розроблення системи для функціонування туризму у місті Вінниця

На даний момент не існує аналогів які б працювали на мобільних телефонах в тому числі і на пристроях під операційної системою iOS. Є можливість користуватися загальними путівниками в електронному вигляді, але на таких сайтах не повністю реалізована весь функціонал путівника у тому числі немає навігації до пам'яток, для того щоб прокласти маршрут до певної точки потрібно витратити багато часу, який можна використати для других цілей.

Наявні програмні продукти потребують багато часу або в них відсутні важливі функції, і, в підсумку, системи які були наведені у п.1.6, не підходять для розв'язання задач виявлених під час функціонального моделювання.

Із урахуванням цього вирішено створити додаток, який буде повністю задовольняти і виконувати поставлені задачі для функціонування туризму у місті Вінниця. Даний засіб має стати підручним у подорожах для гостей міста та для людей які бажають ознайомитись пам'ятниками, оскільки туризм в Україні розвивається, а з подібним програмним продуктом це буде швидше.

1.8 Концептуальна модель системи

Метою створюваного електронного путівника є надання швидкого доступу до інформації про пам'ятки, можливості їх знайти за допомогою функції навігації. Використання туристами такого путівника дозволить пришвидшити і удосконалити процес пошуку інформації про пам'ятки, та навігацію до них.

Знайдені в моделі AS-IS недоліки були виправлені та існуючі процеси удосконалені при створенні нової моделі організації бізнес-процесів «як має бути» – TO-BE (див. рис.Б.1 у додатку Б).

На цій схемі показано покращені і спрощені процеси пошуку інформації про пам'ятки.

При розробці даної концептуальної моделі були створені нові входи, виходи та механізми.

Входи:

- Запит на пошук пам'яток – запит, який викликає функцію для пошуку пам'яток по назві;
- Запит на прокладання маршруту - запит, який викликає функцію яка визначає місцеположення користувача;
- Пам'ятки

Виходи:

- Готовий маршрут – прокладається маршрут, у якому показується приблизний час прибуття та відстань;
- Впорядковані дані – дані про пам'ятки представлені у зручному вигляді;
- Результат пошуку – результат виконання функції пошуку;
- Електронний засіб – мобільний додаток.

Побудована модель «як має бути» (TO-BE) процесу створення електронного путівника, складається з таких процесів:

- Запуск додатку;
- пошук пам'яток;
- прокладання маршруту;
- результат.

які показано на першому рівні декомпозиції (див. рис.Б.2 у додатку Б).

Засіб заповнюється пам'ятками, і має функції пошуку, для яких входами є пам'ятки і запити на пошук та прокладання маршруту. Після виконання цих функцій, виводиться результат на екран.

Також в майбутньому для розвитку засобу, додати функцію бронювання житла, це ще більше автоматизувало б процес. Для зручного використання додатку можна додати англійську мову, що в свою чергу розширило б кількість користувачів.

1.9 Постановка задачі на проектування

1.9.1 Призначення та цілі створення системи

Метою даного проекту є спрощення та пришвидшення навігації до пам'яток міста Вінниця та пошуку інформації про них.

Головним завданням є зручне для користувача відображення матеріалу з використанням технології пошуку, і можливістю навігації до певного місця.

Користувачем даного електронного засобу є туристи.

Створювана система дозволить туристам пришвидшити та спростити процес ознайомлення з містом. Турист буде мати зручний мобільний матеріал при подорожі та можливість швидко знайти потрібний історичний об'єкт або переглянути інформацію про нього.

1.9.2 Вимоги до створюваної системи

Додаток, який створюється повинен мати зручний вигляд для користувача, для не складного використання додатку. Інформація має бути показана у формі простій і зрозумілій для користувача. Мають бути виконані методи, які будуть надавати доступ до матеріалу та навігацію.

- Вимоги щодо ОС — додаток має працювати на смартфонах під ОС IOS із стандартним набором компонентів;

- Вимоги до вигляду користувача — інтерфейс, який буде у зручному та мінімалістичному вигляді; зручний пошук потрібних системних кнопок або функцій; вікно для пошуку даних, додаток повинен мати просту взаємодію з кінцевим користувачем.

- Вимоги до мережевого під'єднання – додаток повинен мати під'єднання до мережі для виявлення GPS координат.

1.9.3 Функції, які має виконувати система

Одні з основних функцій, які має виконувати додаток:

- Виконання функції пошуку у головному меню та у вікні навігації;

- Зручний вигляд навігаційного меню із різними варіантами транспорту;
- Прокладання маршруту;
- Зручний вивід інформації.

1.9.4 Вхідні та вихідні дані системи

Вхідні дані розроблюваної системи:

- Інформація про туристичні місця;
- Запити на пошук;

Вихідні дані розроблюваної системи:

- Результати пошуку;
- Результат прокладання маршруту.
- Впорядковані дані;

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ

2.1 Загальні положення

2.1.1. Найменування системи: «Розроблення IOS додатку путівника по Вінниці»

2.1.2. Результати робіт зі створення системи оформлюються згідно з вимогами ДСТУ на відповідні етапи розробки. Порядок оформлення і передачі результатів у даному випадку визначається змістом і календарним планом виконання розробки.

2.1.3. У випадку необхідності на наступних стадіях робіт по створенню системи окремі положення можуть уточнюватися і розвиватися.

2.2. Призначення і цілі створення системи

2.2.1 Призначення системи.

Система призначена для спрощення пошуку інформації про пам'ятки міста Вінниця. Система служить як IOS-додаток, для туристичних цілей. В додатку присутня інформація про пам'ятки міста, вартість відвідування, години роботи, а також функція за допомогою якої можна здійснювати навігацію до цих пам'яток та їх пошук.

2.2.2. Цілі створення системи.

Основною метою створення системи є забезпечення інформації про туристичні місця міста в одному ресурсі. Це дасть змогу спростити даний процес для туристів та має збільшити потік людей, які бажають ознайомитись з історією Вінниці.

Процес пошуку інформації про пам'ятки складається з використанням великої маси різних джерел і в тому числі паперових, що ускладнює пошук. Тому, якщо автоматизувати цей процес, можливо скоротити час для знаходження інформації і перевести його на більш сучасний рівень, що в свою чергу забезпечить зручність, точність та оперативність знаходження потрібних даних.

2.3. Характеристика об'єкта автоматизації

2.3.1. Короткі відомості про об'єкт автоматизації.

Об'єктом автоматизації є процес пошуку інформації про пам'ятки. Базовий об'єкт впровадження – Туристична сфера міста Вінниці.

2.4. Вимоги до системи

2.4.1. Вимоги до системи в цілому.

2.4.1.1. Вимоги до структури і функціонування системи.

2.4.1.1.1. Система повинна бути написана на основі об'єктно-орієнтованого програмування, також для збереження даних, має бути база-даних.

2.4.1.1.2. Діагностування додатку в процесі використання має показати відхилення або ж помилки виконання певної функції та попередити користувача про відповідне порушення.

2.4.1.1.3. Покращення додатку та його розвиток має проводитись в процесі розвитку відповідних функцій, які можуть покращити роботу функцій, які присутні в додатку.

Програмно-технічні засоби функціонування мають мати функції пошуку, збереження даних, відображення та впорядкування інформації у зручному для користувача вигляді. Додаток повинен мати систему навігації до певного туристичного місця. Додаток повинен мати сумісність з загальносистемним протоколом обміну.

2.4.1.1.4. Функціонування системи має забезпечувати розподілену обробку даних.

2.4.1.2. Вимоги до чисельності і кваліфікації персоналу.

2.4.1.2.1. Персонал, який використовує додаток, повинен дотримуватися наступних вимог:

- отримати навички роботи в операційній системі IOS;
- дотримуватись технічних інструкцій при роботі додатку;
- дотримуватись умов експлуатації мобільного пристрою під операційною системою IOS у відповідності з інструкціями по експлуатації;

- дотримуватись визначених правил збереження даних;
- дотримуватись правил з техніки безпеки при використанні цифрових пристроїв.

2.4.1.2.2. Користувачем системи може виступати турист. Вхід до додатку відбувається без паролей та без реєстрації, що означає що доступ не обмежений.

2.4.1.3. Показники призначення.

2.4.1.3.1. Відповідно до п. 2.1, показники повинні характеризувати ступінь автоматизації та якість додатку, для правильного функціонування. Допустимі значення та перелік показників повинні бути визначені при проектуванні техноробочого процесу.

2.4.1.3.2. Додаток також повинен мати можливість налаштування під користувача та модернізацію його.

2.4.1.4. Вимоги до надійності.

2.4.1.4.1. Додаток є функціональний, його можна використовувати не обмежено. Функції додатку, які присутні виконуються при підключенні до мережі з мобільного пристрою. У відповідності з ДСТУ 2226-93 оцінка надійності додатку проводиться в кожному методі окремо. Враховуючи всі функціональні особливості додатку, показники надійності додатку є показниками надійності обраної бази даних, та засобів технічного реалізування проекту, на яких вона працює. Показниками основних характеристик надійності є:

L_i — ймовірність того що певний запит буде виконано;

K_r — коефіцієнт готовності;

T_v — час відновлення(середній);

T_e — час між першою і другою відмовою протягом одного місяця.

2.4.1.4.2. Комплекс засобів додатку повинен передбачати:

- можливість запуску розв'язання різних функціональних завдань;

Для того щоб було досягнуто надійного та безпечного виконання функцій, потрібно дотримуватись наступних правил:

- структурного і модульного програмування за принципом об'єктно-орієнтованого програмування;
- функцій контролю правильного та безпечного показу інформації;
- засобів для виявлення помилок і коригування в базі даних;
- засобів захисту інформації від різних збоїв або помилок в сховищі даних.

2.4.1.5. Вимоги до безпеки.

Для забезпечення безпеки потрібно виконувати такі вимоги ДСТУ: ДСТУ 2293-99, ДСТУ ISO 6309:2007, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011; для доступних рівнів освітлення, вібраційних, шумових норм, потрібно виконувати наступні вимоги: ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94.

2.4.1.6. Вимоги з ергономіки та технічної естетики.

Ергономічні функції повинні відповідати всім стандартам, а саме ДСТУ 8604:2015, ДСТУ 7298:2013. Робоче місце має дотримуватись таких правил освітлення ДСТУ EN 12464-1:2016, ДБН В.2.5-28-2006.

Прилади, які розміщуються для показу зображення мають стояти щоб кут екрану спостереження не перевищував норму більше, ніж 45 градусів, а відстань до екрану мінімум — 0,3 м, рекомендується — 0,5 м.

При розробці додатку потрібно зробити зручні умови для створення, щоб передбачити стомлення користувача.

2.4.1.7. Норми по використанні, обслуговування техніки, зберігання і ремонту деталей системи.

2.4.1.7.1. Обслуговування додатку контролюється стандартом ДСТУ EN 13306:2019. Загальноприйняті норми по користуванню, технічному ремонті і підтриманні мають бути потримані по стандарту ДСТУ 3576-97.

2.4.1.7.2. Щоб розмістити всі компоненти системи потрібно використовувати площі, які зазначені в ДБН В.2.2-9-2009 і також потрібно дотримуватись правил, які вказані в правилах використання додатку. Напруга технічних засобів живлення системи 220/380 В змінного струму, частотою (50 ± 1) Гц. Відхилення напруги, яке допускається знаходиться в значеннях від +10 до -15%, час періодів відключення живлення не має бути більший за значення 0,001 с.

2.4.1.7.3. Режим роботи, кваліфікація персоналу та кількість, повинна регулюватись відповідно до рекомендацій, які описані в інструкціях технічних приладів.

2.4.1.7.4. Розміщення, умови зберігання приладів на складах повинна регулюватись відповідно до рекомендацій, які описані в інструкціях цих приладів.

2.4.1.7.5. Порядок обслуговування має дотримуватись їх умовам роботи і рівню, щоб у випадку помилки додатку підтримувати його .

2.4.1.8. Вимоги, які захищають додаток від зловмисного доступу.

Для доступу до потрібної інформації і надійності збереження потрібно виконувати такі засоби захисту:

- 1) засобів захисту на стороні клієнта
- 2) методів захищення даних при обміні даними клієнт сервер (токен, рефреш токен, розділення прав доступу)

2.4.1.9. Вимоги, які потрібні для збереження інформаційних даних.

2.4.1.9.1. Потрібно передбачити резервне копіювання бази даних у випадку, коли трапляється помилкове руйнування даних, також додати можливість аварійного копіювання БД.

2.4.1.9.2. Архів БД, який використовується як резервний, повинен зберігатися на різних пристроях або носіях, не на тому ж пристрої що і основна БД.

2.4.1.10. Вимоги для захисту від впливу зовнішніх факторів.

2.4.1.10.1. Складова електромагнітного поля завад не повинна перевищувати в приміщеннях 0,3 В/м² в частотному діапазоні від 0,15 до 300 МГц. Для захисту від випромінювання потрібно використовувати різноманітні екранні засоби захисту.

2.4.1.10.2. Прилади, які не впливають на використання ряду технічних засобів, мають бути виконані відповідно з ДБН В.2.2-9-2009. Засоби для обчислювання по стійкості до зовнішніх проявів повинні дотримуватись ДСТУ 2506-94.

2.4.1.11. Вимоги до патентної чистоти.

При створенні цього додатку таких досліджень не проводилось.

2.4.1.12. Вимоги по стандартизації і уніфікації.

У структурі інформаційного кодування потрібно проводити за світовим класифікатором і стандартом.

2.4.2. Вимоги до функцій.

2.4.2.1. Всі функції із показанням інформації яка входить та виходить показано в таблиці 2.1.

Функції додатку мають виконуватись належним чином а також виконувати задачі, які прописані в цьому документі

Таблиця 2.1 – Перелік функцій, вхідної та вихідної інформації

№ п/п	Назва функції	Інформація, яка входить	Інформація, яка виходить
1	Формування даних про туристичне місце, та виведення інформації	Таблиця БД «Місця»	Вікно, у якому вказано всі туристичні місця у зручному вигляді.
2	Пошук туристичного місця	Запит користувача	Вікно зі списком

№ п/п	Назва функції	Інформація, яка входить	Інформація, яка виходить
		на пошук по назві туристичної пам'ятки	інформації, яка потрібна
3	Прокладання маршруту	Запит користувача на прокладання маршруту до пам'ятки	Вікно мапи у якому відображається шлях
4	Процес перегляду інформації про режим роботи	Таблиця БД «Інформація про пам'ятку»	Вікно відображення інформації у зручному вигляді

2.4.3. Вимоги, які відносять до видів забезпечення.

2.4.3.1. Стосовно вимог мат. забезпечення (МЗ) додатку не потрібно математичне забезпечення для використання функцій, які будуть доступні. Достатньо всіх програмних засобів і влаштованого фреймворку CoreData.

2.4.3.2. Вимоги, які потрібні для інформаційного забезпечення (ІЗ).

2.4.3.2.1. Для інформаційного забезпечення додатку потрібно містити інформацію, щоб можливо було виконати поставлені задачі додатку. ІЗ має гарантувати розподілену оптимальну організацію збереження даних та доступу до них.

Внесення даних в БД виконує додаток та розробник.

Структура, склад і спосіб управління даними відображаються у логічній моделі БД і можуть коригуватись на етапі проектування додатку.

2.4.3.2.2. При порушеннях у енергоживленні додатку потрібно передбачити збереження інформації у БД.

2.4.3.3. Вимоги до лінгвістичного забезпечення (ЛЗ).

2.4.3.3.1. Для створення програмних засобів додатку, які виконують функції і забезпечують обслуговування користувачів мають бути використанні мови високого рівня, через які можливо створити структурний додаток.

2.4.3.3.2. Структура спілкування користувача з додатком повинна відбуватись з застосуванням простого меню, орієнтованого для того щоб користувач між виконати функції. Запити в додатку повинні подаватись у вигляді зрозумілому для користувача.

2.4.3.4. Вимоги до програмного забезпечення (ПЗ).

2.4.3.4.1. ПЗ має виконувати всі поставленні функції і задачі. До потрібного ПЗ належить:

- операційна система (далі ОС) — MacOS;
- фреймворк управління БД (далі СУБД) — CoreData.

2.4.3.4.2. Вимоги, які прийнято вважати загальними:

- вимоги, мінімальні, для виконання поставлених задач;
- швидкодія, яка максимально може бути досягнута;
- повне виконання потреб, які будуть поставленні.

2.4.3.4.3. Вимоги до ОС:

- використання потреб у мобільного пристрою як найменше ресурсів, особливо процесора, оперативної пам'яті;
- максимальна швидкодія при використанні пристрою;
- ОС клієнта — IOS.

2.4.3.4.4. Вимоги до фреймворку CoreData:

- задоволення потреб у використанні функцій;
- якісний контроль потрібної структури і обсягу;
- швидкість при виконанні задач користувача;

2.4.3.4.5. Засоби за допомогою яких буде відбуватись процес взаємодії виведення інформації і введення, а також діалог з користувачем:

- процес виведення інформації має бути представлений у зручному графічному вигляді;

- супровід користувача під час введення даних і інформування його про наявні помилки;

2.4.3.4.6. При розробленні додатку слід дотримуватись відповідних вимог:

- додатки мають бути сумісними із загальними і між собою;

- додаток розроблюється згідно із вимог об'єктно-орієнтованої мови програмування ;

- дотриматись відповідності вікон користувача стандартам IOS;

- необхідна модульна структура програм;

- має бути можливість розширювати функції, які виконує додаток, відповідно до нових потреб користувача;

- додаток не повинен залежати від пристроїв, які підключені у процесі використання;

- спілкування додатка і користувача повинно проводитись із використанням жестів по пристрою, та трактуванням дій користувача і можливих підказок .

2.4.3.5. Вимоги до технічного забезпечення.

2.4.3.5.1. Засоби системи (табл. 2.2) мають виконувати поставлені задачі, які наведені у таблиці 2.1.

2.4.3.5.2. Функції техніки мають виконувати обмін даних в розмірах, приведених в п. 2.3.2.

Таблиця 2.2 – Вимоги до технічного забезпечення системи

№ п/п	Основні характеристики телефону
1	iPhone 6s і новіше
2	ОС IOS 13.0 і новіше

4.3.6. Вимоги до метрологічного забезпечення.

Додаток не має метрологічних функцій вимірювання.

2.4.3.7. Вимоги до забезпечення організації.

2.4.3.7.1. Забезпечення організації додатку виконується у відповідності з нормами держ. стандарту по АСУП.

2.4.3.7.2. При організації автоматизації не потрібно додаткового обслуговуючого персоналу.

2.4.3.7.3. До функціонування системи висуваються наступні вимоги:

- відповідальний за систему розробник додатку;

2.5. Склад і зміст робіт по створенню системи

2.5.1. Стадії створення системи і терміни виконання робіт наведені в таблиці

2.3.

Таблиця 2.3 – Найменування робіт при створенні системи

№ п/п	Найменування робіт	Терміни виконання робіт
1	Дослідження автоматизації об'єкта (передпроектне)	5 днів
2	Технічне завдання	5 днів
3	Технічний проект	25 днів
4	Оформлення документації	15 днів

2.6. Порядок контролю і приймання додатку

2.6.1. Додаток викладається на платформі «AppleStore». При викладенні на ресурси, додаток має відповідати всім вимогам компанії Apple та ресурсах де буде розміщено додаток.

2.6.2. Випробування додатка, при вирішенні його працездатності, виконують розробник, а також замовник. Структуру завдань при випробуванні повинен скласти виконавець і замовник.

2.6.4. Введення в роботу додатку підготовлюється актом здачі-приймання.

2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію

Для підготовки до випуску додатка замовник виконує перелік робіт:

- проводить опис всіх функцій;
- виконує навчальний процес для користувачів та пред'являє інструкцію;
- проводить тестову експлуатацію та виводить додаток в публічний доступ.

2.8. Вимоги до документації

2.8.1. Для додатку виконується ряд технічних документів: технічне завдання та технічний проект.

2.8.2. Документи на додаток виготовляються відповідно з вимогами державної документації 19 «Система єдиної програмної документації» та серії 24 «Єдина Система стандартів автоматизованих систем управління».

2.9. Джерела розробки

2.9.1. При виготовленні технічного завдання було використано такі інформаційні джерела:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;

- ДСТУ 3973–2000 Система розроблення та поставлення продукції на виробництво;
- ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом.

РОЗДІЛ 3. РОЗРОБЛЕННЯ КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1 Обґрунтування вибору засобів розроблення системи

CA ERwin Process Modeler - це потужний засіб моделювання, що допомагає в процесі аналізу, що надає підтримку при створенні документації та дозволяє підвищувати ефективність складних бізнес-процесів. Моделі процесів дозволяють формувати детальну документацію, в яку увійдуть дані про необхідні операції, а також інформація про методи управління зазначеними операціями і необхідних ресурсах. Таким чином, ви отримаєте вичерпне уявлення про методики виконання різних операцій, від організації технологічних процесів в невеликих відділах до комплексної діяльності в масштабах всього підприємства. Налаштовується на основі великоформатних таблиць Словниковий інтерфейс дозволяє швидко і без зайвих зусиль вводити інформацію про моделі і керувати нею. Простий у використанні інтерфейс надає чудові можливості заповнення моделей.

Рішення CA ERwin Process Modeler автоматизує більшість операцій в процесі створення моделей, надаючи семантичну строгість, необхідну для досягнення бажаного високого результату. «Підсвічування» об'єктів дозволяє розробникам уникнути багатьох помилок, що виникають при моделюванні. Розробники зможуть налаштовувати параметри CA ERwin Process Modeler для того, щоб забезпечити збір важливої для бізнесу інформації. Зазначена інформація миттєво стає доступною в рамках редактора звітів CA ERwin Process Modeler і може бути експортована в інші додатки (наприклад, в Microsoft Word або Excel). Завдяки підтримці діаграм Swim Lane, користувачі отримують у своє розпорядження ефективний механізм візуалізації і оптимізації складних процесів. Зазначені діаграми допомагають організовувати процеси і дозволяють переглядати інформацію про процеси, ролях і повноваженнях. Структурні організаційні діаграми відображають способи визначення і реалізації процесів електронної комерції.

Продукт CA ERwin Process Modeler передбачає можливість явного визначення ролей, які відповідають за опис і класифікацію задач в рамках окремого процесу.

Swift - мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective C і бути стійкішою до помилкового коду. Swift була представлена на конференції розробників WWDC 2014. Мова побудована з LLVM компілятором, включеного у Xcode 10. Безкоштовний посібник мови програмування Swift доступний для завантаження у магазині iBooks.

Компілятор Swift побудований з використанням технологій вільного проекту LLVM. Swift успадковує найкращі елементи мов C і Objective-C, тому синтаксис звичний для знайомих з ними розробників, але водночас відрізняється використанням засобів автоматичного розподілу пам'яті і контролю переповнення змінних і масивів, що значно збільшує надійність і безпеку коду.

Одним із важливих нововведень Swift, в порівнянні з Objective-C, є можливість писати код і переглядати результати в реальному часі. Ми можемо внести деякі зміни в програмі і відразу побачити результат, так що нам не потрібно перекомпілювати і перезапускати програму. Це має назву Interactive Playgrounds. Також тут можна використовувати Quick Look, який відображає графіку або список результатів, Timeline Assistant, який допомагає експериментувати з кодом UI (інтерфейсом користувача) або продивлятися повний цикл створення анімацій. Даний код можна перенести в основний проект (рис. 2.4).

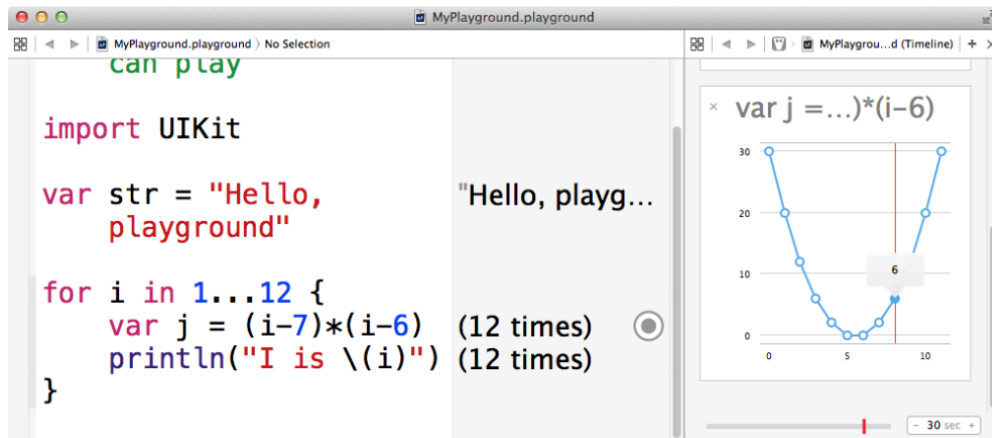


Рисунок 2.1 - Interactive Playgrounds

Порівнюючи Swift і Objective-C можна з впевненістю сказати, що Swift - це сучасні норми синтаксису, ефективне управління пам'яттю, висока швидкість роботи і інтерактивність. У Objective-C цього немає, але зате є надійність, база документації, прикладів, шаблонів і багато досвідчених програмістів. Swift - це майбутнє, яке вже не за горами. Все пізнається в порівнянні і лише відпрацювавши певну кількість проектів, можна стверджувати, чи зручна мова, і чи можна на ній ефективно працювати.

Проаналізувавши особливості кожного з середовищ розробки, Xcode перемагає усіх вище представлених учасників, адже працювати в ньому - одне задоволення. Воно гнучке, швидке, потужне і здатне завжди прийти на допомогу. Xcode стає все краще, незважаючи на складні заходи, які вживаються Apple з метою утримання повного контролю над iOS додатками і пристроями. Робочий простір в Xcode тримає розробника зосередженим. Під час написання коду, в реальному часі можна побачити помилки компілятора або проблеми, а також повідомлення з докладним описом корисної інформації. Відладчик працює плавно, а симулятор - швидкий і орієнтований на користувача. Отже, Xcode перевершує всі інші IDE.

Перед кожним розробником завжди постає питання: які саме інструменти вибрати для роботи. Головними критеріями стають простота у використанні та швидкість. Тому в дипломній роботі будуть використовуватися найновіші продукти від компанії Apple - середовище розробки Xcode і мова

програмування Swift. Адже вони увібрали найкращі особливості універсальних інструментів для розробки мобільного програмного забезпечення. [11].

Core Data - фреймворк від компанії Apple, вбудований в операційну систему iOS, MacOS, який дозволяє розробнику взаємодіяти з базою даних. Був представлений компанією Apple с анонсом Mac OS X 10.4 Tiger і iOS з iPhone SDK 3.0. Дозволяє даними бути організованими в Сутність - Атрибут - Значення (EAV). Управління даними може бути здійснено за допомогою маніпуляцій сутностей і їх взаємозв'язків.

Структура:

- Managed Object Context - компонент з яким йде взаємодія, кожен раз, коли йде збереження або перезапис;
- Persistent Store Coordinator - виконує мета зберігання даних;
- Managed Object Model - модель БД;
- Persistent Store - репозиторій, де зберігаються дані;

Core Data описує дані, які зберігаються в iOS додатку, код може маніпулювати для збереження і записи даних в додатку. Core Data організований в величезні класи.

Суть роботи фреймворка, як внутрішньої БД проста: створення моделі, і при додаванні нових елементів збереження за допомогою методу `saveContext ()`; в модель за допомогою нижче наведених методів в таблиці.

Додаток не може працювати без БД, так як при виході з нього дані будуть втрачені. Це і є головною метою даного фреймворка - зберігання даних, є так само і схожі: Realm, SQLite.

Core Data може конвертувати дані в XML, бінарний код, SQLite для зберігання. Core Data схеми стандартизовані. Якщо на комп'ютері не встановлено Xcode, то можливість прочитати модель даних є.

3.2 Проектування БД

ERWin надає засоби, що допомагають розробнику створити наглядну логічну (див. рис. 1 у додатку В) та фізичну (див. рис. 1 у додатку Г) схему (модель) бази даних, провести її дослідження та оптимізацію.

Таблиці бази даних:

Таблиця «Місце» це головна таблиця, яка поєднує в собі дані з таблиці «Розташування» та «Інформація».

Структуру даної таблиці описано в табл. 2.1.

Таблиця 2.1. Структура таблиці «Місце»

№	Назва поля	Тип даних
1	Код_місця	INTEGER
2	Назва	CHAR(16)
3	Код_Інформації	INTEGER
4	Фото	CHAR(16)
5	Код_розташування	INTEGER

Таблиця «Розташування» містить у собі дані розташування, це координати широти та довжини місця.

Структуру даної таблиці описано в табл. 2.2.

Таблиця 2.2. Структура таблиці «Розташування»

№	Назва поля	Тип даних
1	Код_розташування	INTEGER
2	Широта	CHAR(16)
3	Довжина	CHAR(16)

Дані про туристичні точки містяться у таблиці «Інформація».

Структуру даної таблиці описано в табл. 2.3.

Таблиця 2.3. Структура таблиці «Інформація»

№	Назва поля	Тип даних
1	Код_Інформації	INTEGER
2	Опис	TEXT(255)
3	Графік_роботи	CHAR(16)
4	Вартість	CHAR(16)

Створена база даних призначена для зберігання інформації про туристичні місця та всю потрібну інформацію про них. Вона містить у собі туристичні місця, підключення до неї надасть змогу їх обробляти.

3.3 Створення інтерфейсу користувача

У класі головного вікна PointVC реалізовано 8 методів, також в цьому класі реалізується запис в базу даних.

Основні функції, що реалізують методи вирішення задачі:

- `override func viewDidLoad()` - стандартний метод в якому реалізуються функції, які спрацюють при відкритті головного вікна:

```
override func viewDidLoad() {
    super.viewDidLoad()
    tableView.rowHeight = UITableView.automaticDimension
    tableView.estimatedRowHeight = 600
    //CoreData
    container = appDelegate.persistentContainer
    context = container.viewContext
    //Star program
    Start()
    //Setup the search controller
    searchController.searchResultsUpdater = self
    searchController.observesBackgroundDuringPresentation =
false
```

Вікно конструювання заданого класу в середовищі Xcode 10.0 подано на рисунку 2.2

```

9 import UIKit
10 import CoreData
11 // public variable for More Point
12 var imageM = UIImage()
13 var nameM = String()
14 var infoM = String()
15 //points for map
16 var points2 = [Point]()
17 //
18 class PointTVC: UITableViewController {
19
20     var start = false
21     let appDelegate = UIApplication.shared.delegate as! AppDelegate
22     var container: NSPersistentContainer!
23     var context: NSManagedObjectContext!
24
25     var points = [Point]() {
26         didSet {
27             tableView.reloadData()
28         }
29     }
30     // var search
31     private let searchController = UISearchController(searchResultsController:
32     var filteredPoint = [Point]()
33     private var searchBarIsEmpty: Bool {
34         guard let text = searchController.searchBar.text else { return false }
35         return text.isEmpty
36     }
37     private var isFiltering: Bool {
38         return searchController.isActive && !searchBarIsEmpty
39     }
40     //
41     override func viewDidLoad() {
42         super.viewDidLoad()
43         tableView.rowHeight = UITableView .automaticDimension
44         tableView.estimatedRowHeight = 600
45         //CoreData
46         container = appDelegate.persistentContainer
47         context = container.viewContext
48         //Star program
49         Start()
50         //Setup the search controller
51         searchController.searchResultsUpdater = self
52         searchController.observesBackgroundDuringPresentation = false
53         searchController.searchBar.placeholder = "Пошук"
54         definesPresentationContext = true
55         navigationItem.searchController = searchController

```

Рисунок 2.2 - реалізація класу головного вікна

- override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) - стандартний метод для відображення змісту комірок:

```

override func tableView(_ tableView: UITableView, cellForRowAt
indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier:
"Cell", for: indexPath) as! PointTVCCell

```

```

var point: Point
if isFiltering {
    point = filteredPoint[indexPath.row]
} else {
    point = points[indexPath.row]
}

```

```

container = appDelegate.persistentContainer

```

```

        context = container.viewContext
        //Star program
        cell.infoLabel.text = point.info
        cell.infoLabel.text = point.a
        cell.nameLabel.text = point.name
        cell.infoLabel.text = point.info
        cell.imagePointCell.image
UIImage(named:point.image!)
        return cell
    }

```

Даний клас є одним із головних класів, так як він відповідає за головне вікно програми. Головне вікно (Рис 2.3) програми містить кнопку «Про додаток», кнопку пошуку, показ підвікон з туристичними точками та перехід до навігації по картам.



Рисунок 2.3 - Головне вікно програми

При розроблені вікна MoreInfoVC реалізовано 3 методи.

Основні функції, що реалізують методи вирішення задачі:

- `override func viewDidLoad()` - стандартний метод в якому реалізуються методи, які спрацюють при відкритті вікна:

```
override func viewDidLoad() {
    super.viewDidLoad()
    infoProgram.text = infoString
}
```

- `public func UIColorFromRGB(rgbValue: UInt) -> UIColor` - метод в якому реалізується конвертування кольора з RGB в HEX:

```
public func UIColorFromRGB(rgbValue: UInt) -> UIColor {
    return UIColor(
        red: CGFloat((rgbValue & 0xFF0000) >> 16) / 255.0,
        green: 255.0,
        blue: CGFloat((rgbValue & 0x00FF00) >> 8) /
        255.0,
        alpha: CGFloat(1.0)
    )
}
```

- `public func imageEllipsAndColor(Image: UIImageView)` - метод в якому реалізується кругла рамка для зображення і колір цієї рамки:

```
public func imageEllipsAndColor(Image: UIImageView) {
    Image.layer.cornerRadius = Image.frame.size.width / 2
    Image.clipsToBounds = true
    Image.layer.borderWidth = 3
    Image.layer.borderColor = (UIColorFromRGB(rgbValue:
0x9F3AF8)).CGColor
}
```

У цьому класі описано функції які допомагають реалізувати певні можливості додатка, та декілька змінних для забезпечення відображення інформації у даному вікні.

Вікно програми «Про додаток» містить інформацію про додаток і про розробника, зв'язок з ним. Вікно «Про додаток» зображено на рисунку 2.4.

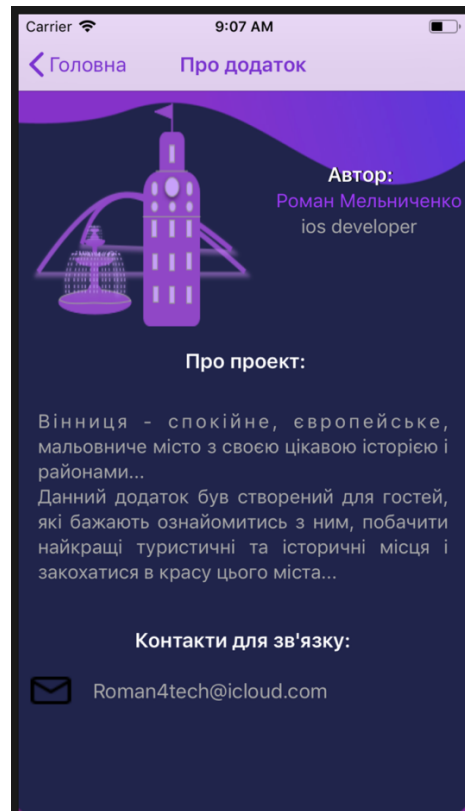


Рисунок 2.4 - Вікно «Про додаток»

У класі головного вікна MorePoint реалізовано 3 методи.

Основні функції, що реалізують методи вирішення задачі:

- `override func viewDidLoad()` - стандартний метод в якому реалізуються методи, які спрацюють при відкритті вікна:

```
override func viewDidLoad() {  
    tableView.rowHeight = UITableView .automaticDimension  
    tableView.estimatedRowHeight = 100  
}
```

- override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) - стандартний метод для відображення змісту комірок:

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "CellMore", for: indexPath) as! MorePointTableViewCell
    cell.imageMore.image = imageM
    cell.nameMore.text = nameM
    cell.infoMore.text = infoM
    return cell
}
```

У цьому класі представленні методи реалізують показ детальної інформації про місце.

Вікно програми «Про туристичну точку» містить інформацію про туристичну точку та фото. Вікно зображено на рисунку 2.5.



Рисунок 2.5 - вікно «Про туристичну точку»

У класі головного вікна MapVC реалізовано 24 методи.

Основні функції, що реалізують методи вирішення задачі:

- `override func viewDidLoad()` - стандартний метод в якому реалізуються методи, які спрацюють при відкритті вікна:

```
override func viewDidLoad() {
    super.viewDidLoad()
    searchBar.delegate = self
    mapView.delegate = self
    tableView.delegate = self
    tableView.dataSource = self
    setHiddenElement(bool: true)
    buttonAutomobileOutlet.isHidden = true
    walkingImage.isHidden = true
    configureLocationServices()
    AnnotationPoint()
    setupViews()
}
```

`func mapView` - метод для налаштування вигляду анотацій, Код можна переглянути у додатку Є.1

- `func mapView(_ mapView: MKMapView, annotationView view: MKAnnotationView, calloutAccessoryControlTapped control: UIControl)` - метод, який був створений для керування аксесуарами анотації, щоб створити такі кнопки як інформація про туристичне місце та навігація до нього, при нажаті на це місце на мапі. Код цього методу:

```
func mapView(_ mapView: MKMapView, annotationView view:
MKAnnotationView, calloutAccessoryControlTapped control:
UIControl) {
    switch control {
        case let left where left ==
view.leftCalloutAccessoryView:
            pointAnnotation = (view.annotation as! Artwork)
            mapView.removeOverlays(mapView.overlays)
            tabButtonRout = false
    }
    if let locMy = locationMy {
        rout(firstCoordinate: locMy, secondCoordinate:
```

```

pointAnnotation!.coordinate, transType: transport )
    }
    showRoute()
    break
//right accessory
case let right where right == view.rightCalloutAccessoryView:
    let pointAnnotation = view.annotation as! Artwork
    for point in points2{
        if point.name == pointAnnotation.title {
            let vc = storyboard?.instantiateViewController(withIdentifier:
"MoreVC")
self.navigationController?.pushViewController(vc!, animated: true)
                imageM = UIImage(named: point.image!)!
                nameM = point.name!
                infoM = point.info!
            }
        }
    }
}

```

- `func rout(firstCoordinate: CLLocationCoordinate2D, secondCoordinate: CLLocationCoordinate2D, transType: Transp)` - метод, який допоможе прокласти маршрут між користувачем і туристичною точкою, в цьому методі ми вказуємо, яким транспортом буде рухатись користувач, скільки часу потрібно до прибуття і яка відстань до цього місця, також реалізовані візуальні ефекти, в тому числі і анімація. Код цієї функції можна переглянути у додатку Є.2

- `private func AnnotationPoint()` - метод, який добавляє анотації на мапу:

```

private func AnnotationPoint(){
    for point in points2{
        let artwork = Artwork(title: point.name!, coordinate:
CLLocationCoordinate2D(latitude: point.latitude, longitude:
point.longitude), image: point.image!)
        mapView.addAnnotation(artwork)
        artworks.append(artwork)
    }
}

```

- `private func configureLocationServices()` - метод, який запитує дозвіл в користувача на доступ до GPS, запускає відстежування і показ GPS:

```

private func configureLocationServices() {
if CLLocationManager.authorizationStatus() == .notDetermined {
locationManager.requestWhenInUseAuthorization()
mapView.showsUserLocation = true
locationManager.startUpdatingLocation()
} else {
mapView.showsUserLocation = true
locationManager.startUpdatingLocation()
}
}
}

```

Вікно програми «Мапа» містить інформацію про місцезнаходження користувача та туристичних точок, навігація до них (рис. 2.6.).

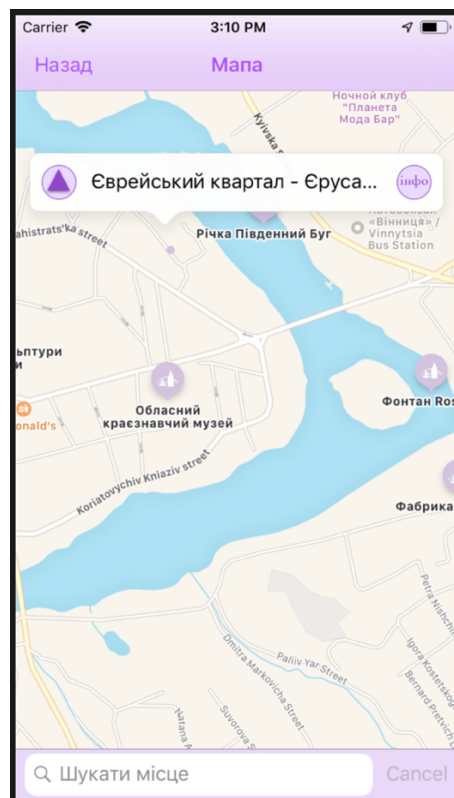


Рисунок 2.6 - вікно «Мапа»

Для реалізації вікна «Мапа» було створено метод func locationManager, який під час оновлення геопозиції оновлює шлях. При натисканні кнопки навігації визначається поточна геопозиція та напрямлення користувача. Потім

створили функцію Rout, яка бере геопозицію користувача і кінцевої точки, визначає оптимальний маршрут і малює його на мапі червоною лінією. Було додано також функцію оновлення прокладеного шляху, за допомогою функції DispatchQueue, при виконанні цієї функції відбувається прокладання нового шляху, а потім видалення старого шляху, все це оновлюється кожену секунду, таким чином користувач буде бачити плавне відображення маршруту. Ось код, за допомогою якого було це все реалізовано:

```
func locationManager(_ manager: CLLocationManager,
didUpdateLocations locations: [CLLocation]) {
    //update rout
    if tabButtonRout {
        if let locMy = locations.last?.coordinate {
            mapView.setUserTrackingMode(MKUserTrackingMode.followWithHeading,
            animated: true)
            rout(firstCoordinate: locMy, secondCoordinate:
            pointAnnotation!.coordinate, transType: transport )
            DispatchQueue.main.asyncAfter(deadline: .now() + .seconds(1),
            execute: {
                if let remove = self.mapView.overlays.last{
                    self.mapView.removeOverlay(remove)
                }
            })
            let region = MKCoordinateRegion(center: locMy,
            latitudinalMeters: 300, longitudinalMeters: 300)
            mapView.setRegion(region, animated: true)
        }
        // Increases map
        if one == false {
            if let myLocation = locations.last?.coordinate{
                locationMy = myLocation
                let region = MKCoordinateRegion(center: myLocation,
                latitudinalMeters: 3000, longitudinalMeters: 3000)
                mapView.setRegion(region, animated: true)
            }
        }
    }
}
```

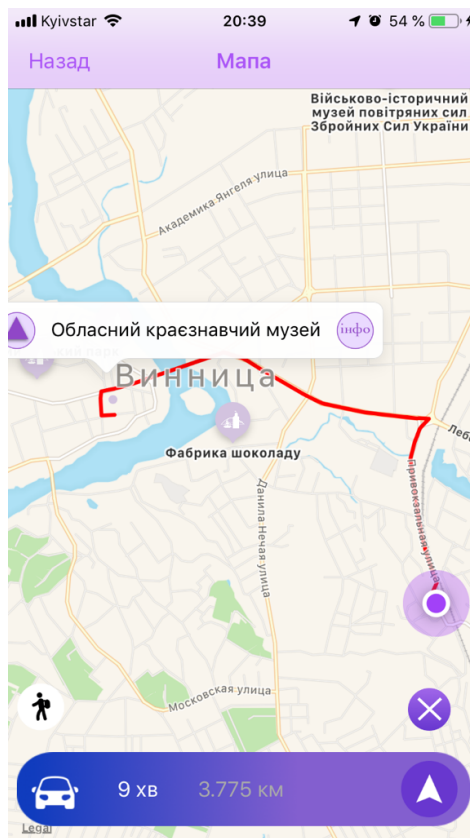


Рисунок 2.7 - Вікно «Мапа», прокладання шляху

- `func animateView(to state: State, duration: TimeInterval)` - метод, який керує анімацією:

```
func animateView(to state: State, duration: TimeInterval){
    guard runningAnimators.isEmpty else { return }
    let basicAnimator = UIViewPropertyAnimator(duration:
duration, curve: .easeIn, animations: nil)
    basicAnimator.addAnimations {
        switch state {
        case .open:
            self.pointButtonConstraint.constant
            =
self.viewOffset
            self.mainViewSearch.layer.cornerRadius = 15
        case .closed:
            self.pointButtonConstraint.constant = 0
            self.mainViewSearch.layer.cornerRadius = 0
        }
        self.view.layoutIfNeeded()
    }
    basicAnimator.addCompletion { (animator) in
```

```

        self.runningAnimators.removeAll()
        self.state = self.state.opposite
        if self.pointButtonConstraint.constant == 0 {
            self.searchBar.endEditing(true)
            self.searchBar.text = ""
            self.searching = false
            self.tableView.reloadData()
        }
    }
    runningAnimators.append(basicAnimator)
}

```

Також у вікні «Мапа» потрібно було реалізувати пошук по місцях, для реалізації даної функції було створено спеціальне розширення для пошуку extension MapVC: UISearchBarDelegate:

```

func searchBar(_ searchBar: UISearchBar, textDidChange
searchText: String) {
    filteredPoint = points2.filter({ (point: Point) -> Bool in
return
(point.name?.lowercased().contains(searchText.lowercased()))!
    })
    searching = true
    tableView.reloadData()
}
func searchBarTextDidBeginEditing(_ searchBar:
UISearchBar) {
    self.pointButtonConstraint.constant = 420
}
func searchBarCancelButtonClicked(_ searchBar:
UISearchBar) {
    searching = false
    searchBar.text = ""
    searchBar.endEditing(true)
    self.pointButtonConstraint.constant = 0
    tableView.reloadData()
}
}
}

```

Вікно програми «Мапа» містить інформацію про місцезнаходження користувача та туристичних точок, навігація до них. Вікно «Мапа» зображено на рисунку 2.8.

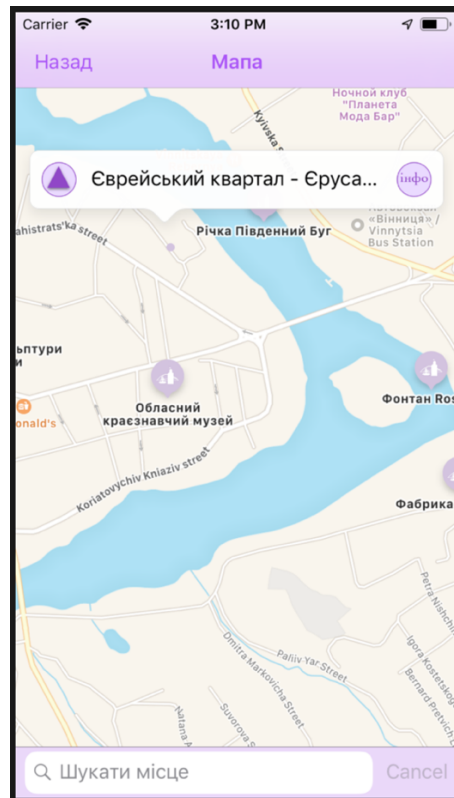


Рисунок 2.8 - вікно «Мапа»

Вікно конструювання заданого класу в середовищі Xcode 10.0 подано на рисунку 2.9, на ньому можна переглянути змінні, які були використанні у розробці цього вікна, а також `enum`-и.

```
6 //
7
8 import UIKit
9 import MapKit
10
11 class MapVC: UIViewController, MKMapViewDelegate, CLLocationManagerDelegate, UITableViewDelegate, UITableViewDataSource {
12     var locationMy: CLLocationCoordinate2D?
13     var artworks: [Artwork] = []
14     var pointAnnotation: Artwork?
15     let objclMP = MoreInfoVC()
16     // user location var
17     var one = false
18     var tabButtonRout = false
19     lazy var locationManager: CLLocationManager = {
20         let manager = CLLocationManager()
21         manager.delegate = self
22         return manager
23     }()
24     //animated main map
25     enum Transp {
26         case avto
27         case walking
28     }
29     enum State {
30         case closed
31         case open
32         var opposite: State {
33             return self == .open ? .closed : .open
34         }
35     }
36     var state: State = .closed
37     var runningAnimators: [UIViewPropertyAnimator] = []
38     let viewOffset: CGFloat = 420
39     //var search
40     var filteredPoint = [Point]()
41     var searching = false
42     @IBAction func buttonInfo(_ sender: UIButton) {
43         var point: Point
44         if searching {
45             point = filteredPoint[(sender as AnyObject).tag]
46         } else {
```

Рисунок 2.9 - реалізація вікна «MapVC»

Це найоб'ємніший клас проекту, в якому всі методи реалізують функції вікна «MapVC».

3.4 Інструкція користувача

Робота з програмою розпочинається з запуску програми «TravelVN» з AppleStore. Після завантаження програми з'являється заставка додатка(рис.2.4.1), потім відкривається головне вікно програми з меню, що зображене на рисунку 2.4.2.

Головне меню програми містить наступні пункти і підпункти меню, що подаються в ієрархії:

- Туристичні точки;
- Відкрити вікно з детальною інформацією;
- Пошук по туристичним місцям;
- Мапа;

- Пошук туристичних місць на карті;
- Навігація до потрібного місця;
- Інформація про місце;
- Відстань та приблизний час в дорозі;
- Про програму;
- Вихід.



Рисунок 2.4.1- Копія вікна програми запуску додатка



Рисунок 2.4.2- Копія головного вікна програми з відкритим пунктом меню «Туристичні точки»

Призначення пунктів та підпунктів меню відповідає їх назві, тому не подається детальне обґрунтування роботи головного меню.

Робота з додатком починається з головного вікна, де можна знайти всі туристичні місця Вінниці та детально з ними ознайомитись.

Після натискання на відповідному пункті програма переходить до другого вікна з детальною інформацією про місце, як показано на рисунку 2.4.3.



Рисунок 2.4.3- Вікно «Про туристичну точку» при виборі «Фонтан Roshen»

Якщо на головному вікні натиснути на кнопку «Мапа» запускається вікно навігації (див. рисунок 2.4.4). При натисканні на вибране місце відкривається вікно в якому написана назва місця та присутні дві кнопки, кнопка «Прокласти шлях» та кнопка «Детальна інформація» (див. рисунок 2.4.5). Якщо натиснути на кнопку навігації, відкриється маршрут від місцезнаходження до вибраної точки користувачем (див. рисунок 2.4.6).

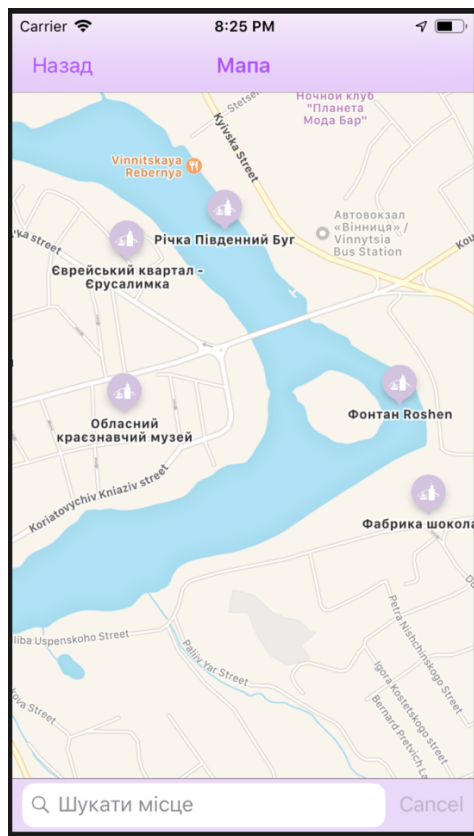


Рисунок 2.4.4- Вікно «Мапа»

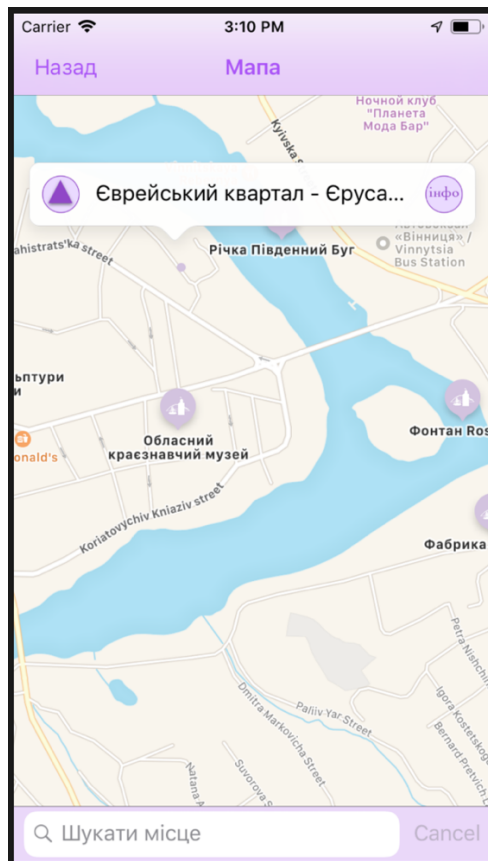


Рисунок 2.4.5 - Вікно «Мапа» з натиском на місце

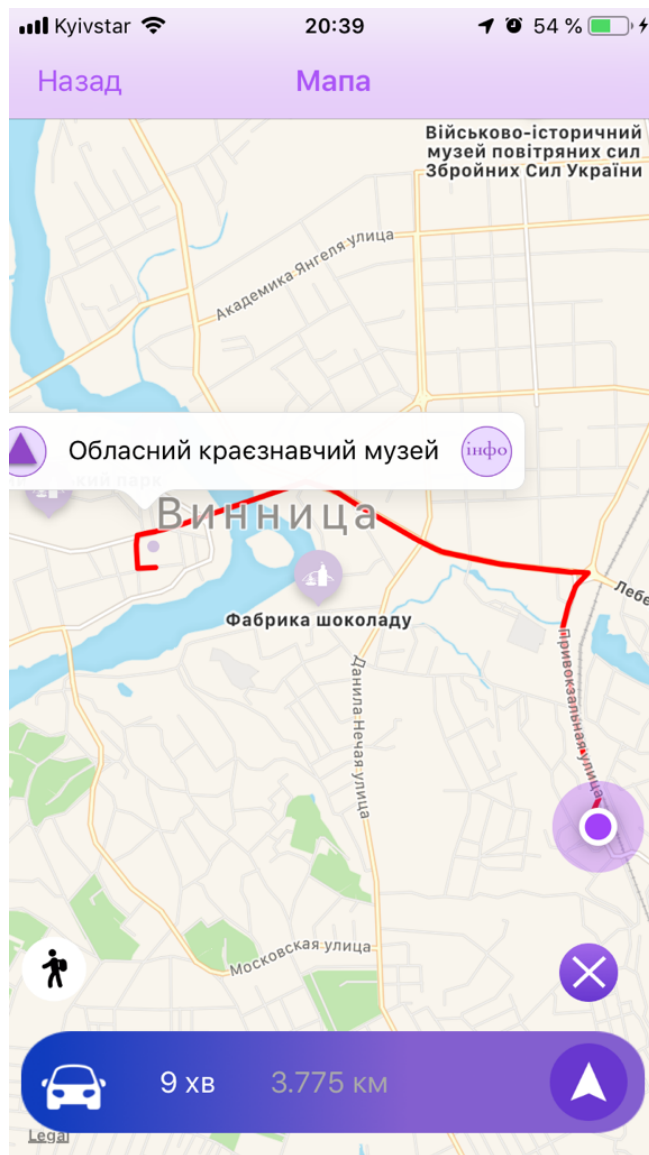


Рисунок 2.4.6 - Вікно «Мапа», прокладання шляху

Після користувач може вибрати тип маршруту (машиною, пішки) та натиснути на кнопку «Вирушити» (див. рисунок 2.4.7).

Також у вікні мапа є додаткове вікно з усіма місцями та пошуку за назвою (див. рисунок 2.4.8), тут теж можна прокласти шлях і подивитися детальну інформацію.

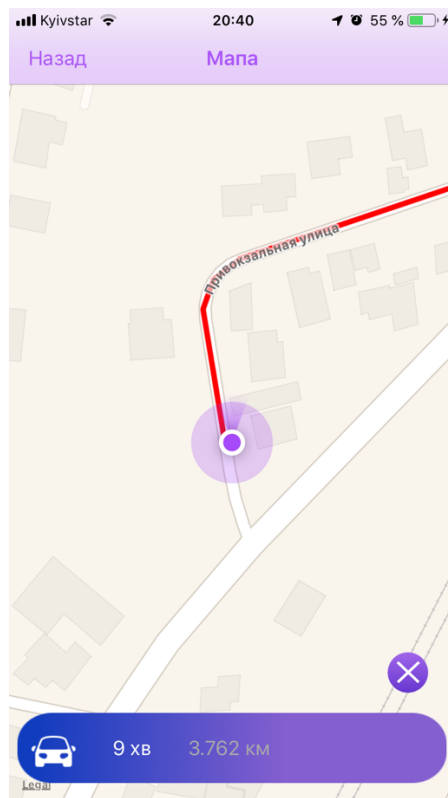


Рисунок 2.4.7 - Вікно «Мапа», навігація до місця

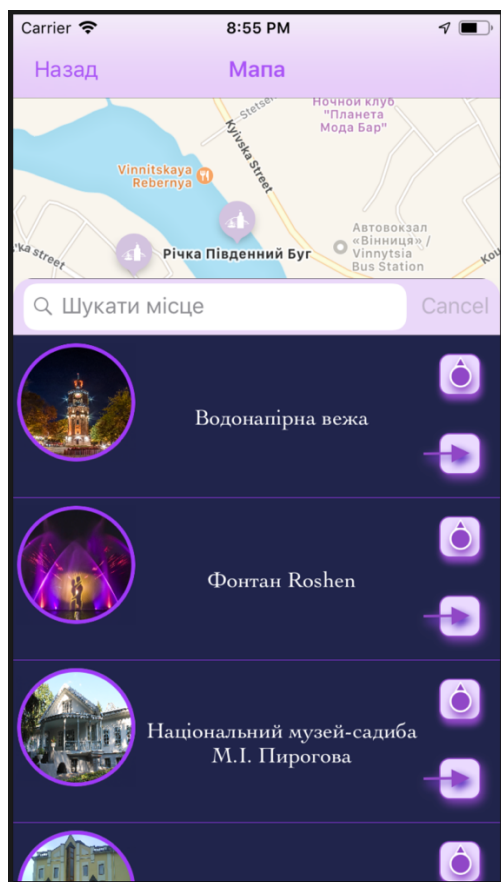


Рисунок 2.4.8 - Вікно «Мапа», пошук місця

3.5 Техніко-економічне обґрунтування

Вихідні дані для підрахунку:

- Ступінь новизни задач – "В" – використання типових проектних рішень за умови їх змін.

- Група складності – 1.

- Узагальнені дані вхідної та вихідної інформації для системи

«Електронний путівник для міста Вінниця» у табл.2.3.

Таблиця 2.3. Узагальнені дані

Вид інформації	Позначення	К-сть наборів даних
Змінна інформація	ЗІ	m=6
Нормативно-довідкова інформація	НДІ	n=4
Банк (база) даних	БД	p=1
Обробка в режимі реального часу	РЧ	так
Забезпечення телекомунікаційної обробки даних і управління віддаленими об'єктами	ТОУ	ні

Витрати часу на додаток, який використовується для управління інформацією, а саме на виготовлення передпроектного дослідження Т1 і технічного завдання Т2, будуть наступні табл. 2.4.

Таблиця 2.4. Визначення витрат часу

Вид системи	Стадія розробки системи	
	Передпроектне дослідження	Технічне завдання
	В	В
Управління науково-технічною інформацією	T1= 67	T2 =24

• Знаходяться стандартні показники витрати часу для стадій "Технічний проект", "Робочий проект" і "Впровадження".

Вхідною інформацією для визначення є:

- значення кількості форм вихідних даних $V1 = 2$,
- значення кількості форм вихідних даних $V2 = 3$,
- стандартне значення часових витрат для "Технічний проект": $T_{B3} = 76$;
- стандартне значення часових витрат для "Робочий проект": $T_{B4} = 214$;
- стандартне значення часових витрат для "Впровадження": $T_{B5} = 74$.
- стандартне значення часових витрат T_B змінюється за допомогою змінюваних коефіцієнтів всіх стадій розроблення автоматизованої системи.

Розрахунок витрат часу стадії "Технічний проект" (T_3).

Коефіцієнт трудомісткості k_p визначається за формулою:

$$k_p = \frac{1.0 * 7 + 0.72 * 4 + 2.08 * 1}{7 + 4 + 1} = 0.997$$

Таблиця 2.5 Коефіцієнти k_1 , k_2 , k_3 для стадії "Технічний проект"

Вид використаної інформації	Ступінь новизни
	В
k1 (ЗІ)	1.0
k2 (НДІ)	0.72
k3 (БД)	2.08

Таблиця 2.6. Коефіцієнт новизни даного проекту, k_0 для системи

«Електронний путівник для міста Вінниця»

Стадія розробки системи	Вид обробки	Ступінь новизни
		В
Технічний проект	РЧ	1.26
Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

Коефіцієнт ступеню новизни роботи, k_0 , що враховує вигляд обробки інформації для трьох стадій розробки системи визначається з урахуванням того, що виконання відбувається в реальному часі.

Витрати часу для стадії "технічний проект" T_3 :

$$T_3 = T_{БЗ} * k_{п} * k_0 = 66 * 0.886 * 1.25 = 95.616;$$

Розрахунок витрат часу для "Робочий проект" (T_4) для системи

Для визначення використання часу на стадії "робочий проект" використовують формулу, де $k_{п}$ – коефіцієнт, що враховує вигляд використаної інформації і визначається за формулою:

$$K_{п} = \frac{1.2 * 7 + 0.65 * 4 + 0.54 * 1}{7 + 4 + 1} = 0.962$$

Таблиця 2.7. Коефіцієнти k_1, k_2, k_3 для стадії "Робочий проект".

Інформаційний вигляд, який використовувався	Група складності алгоритму	Ступінь новизни
		В
k1 (ЗІ)	1	1.1
k2 (НДІ)	1	0.58
k3 (БД)	1	0.48

Коефіцієнт, що враховує вид обробки інформації на стадії "Робочий проект" табл. 2.6. Коефіцієнт складності контролювання інформації, як вхідної так і вихідної і визначається на стадії "Робочий проект" і "Впровадження".

$$k_c = 1.00$$

Витрати часу T_4 вимірюються в людино-днях:

$$T_4 = T_{B4} * k_n * k_o * k_c = 215 * 0.851 * 1.21 * 1.00 = 162.014;$$

Поправочні коефіцієнти в таких самих значеннях як при підрахуванні T_4 .

Щоб знайти стадію визначення загальних витрат часу на "Впровадження" T_5 (люд-днів) виконують підрахунки за формулою:

$$T_5 = T_{B5} * k_n * k_o * k_c = 74 * 0.852 * 1.1 * 1.00 = 76.201;$$

Виходить що, витрати праці людей на складання цієї системи дорівнюють:

$$T_{\Sigma} = 60 + 32 + 75.617 + 162.014 + 76.200 = 460.035_{\text{(люд-дн)}}.$$

Для цієї роботи кількість днів буде 530 із 7- годинним робочим днем, тому щоб розробити даний проект було виділено Φ , днів:

$$\Phi = \frac{530}{7} = 75$$

Для кваліфікаційного проекту $\Phi = 75$ днів. Тоді визначаємо кількість місяців із розрахунку 25 робочих днів. Кількість місяців на розробку, M :

$$M = \frac{\Phi}{25} = \frac{75}{25} = 3$$

Отже, для виконання такого проекту потрібно таку чисельність виконавців \mathcal{C} , виконавців, обраховується за:

$$Ч = \frac{T\Sigma}{\Phi} = \frac{570.046}{75} \approx 8$$

Якщо візьмем значення оплати праці виконавця 17500 грн, то загальна оплата складе:

$$V'_1 = Ч * М * З_{Ппр} = 8 * 3 * 17500 = 420000 \text{ грн.}$$

Витрати, пов'язані з розробкою програми на ПК Розрахунок річного фонду часу роботи ПК.

Річний фонд використання ПК в часу у годинах відповідає числу робочих у році для оператора (в середньому 5год/міс + 6 роб.днів/рік).

$$T_{ПК} = 3723 - (6 * 8 + 5 * 12) = 3615 \text{ год.}$$

Оскільки під час виконання дипломного проекту (роботи) студент в середньому витрачає 450 год. машинного часу, то величина фонду часу ПК дорівнює

$$T'_{ПК} = 3615 * \frac{450}{3723} = 436.9 \text{ год}$$

Поточні витрати на експлуатацію V1

ЦР – ринкова вартість ПК, орієнтовно складає 17000 грн., КУН – коефіцієнт, що враховує витрати на установку і налагодження ПК і дорівнює:

$$Ц_{ПК} = Ц_r * (1 + K_{ун}) = 17000 * (1 + 0.12) = 19040 \text{ грн.}$$

Амортизаційні відрахування використання ПК, З_{АМ}, норма амортизаційних відрахувань, яка для ПК дорівнює НА = 5:

$$З_{АМ} = \frac{19040}{5} = 3808 \text{ грн.}$$

Витрати на електроенергію, споживану ПК, визначаються :

$$З_{\text{ЕЛ}} = Р_{\text{ПК}} * T'_{\text{ПК}} * Ц_{\text{ЕЛ}} * A = 0.032 * 436.9 * 2.1 * 0.9 = 26.42 \text{ грн}$$

ЗР – витрати на поточний ремонт і технічне обслуговування ПК визначаються як 6% від балансової вартості ПК, ЦПК.

$$З_{\text{Р}} = Ц_{\text{ПК}} * 0.06 = 19040 * 0.06 = 1142.4 \text{ грн.}$$

ЗМАТ – непрямі витрати, пов'язані з експлуатацією ПК, визначаються як 5% від балансової вартості ПК ЦПК.

$$З_{\text{МАТ}} = Ц_{\text{ПК}} * 0.05 = 19040 * 0.05 = 952 \text{ грн.}$$

Таким чином, маємо, що заробітна плата обслуговуючого персоналу:

$$З_{\text{ОП}} = 3360 \text{ грн, } З_{\text{АМ}} = 896 \text{ грн, } З_{\text{ЕЛ}} = 26.42 \text{ грн,}$$

Поточні витрати на експлуатацію V_1'' , грн, визначаються:

$$V_1'' = З_{\text{ОП}} + З_{\text{АМ}} + З_{\text{ЕЛ}} + З_{\text{Р}} + З_{\text{МАТ}} = 3360 + 896 + 26.42 + 1142.4 + 952 = 9289 \text{ грн}$$

Отже, загальні витрати на розробку програмного забезпечення комп'ютерної системи розраховуються за формулою і складуть:

$$V = V' + V_1'' = 420000 + 9289 = 429289 \text{ грн}$$

Витрати на придбання і установку ПК V_2

Витрати на придбання і установку ПК (V_2):

$$V_2 = Ц_{\text{ПК}} = 19040 \text{ грн.}$$

Якщо немає потреби в купівлі ПК, то ці витрати дорівнюють "0".

Витрати на підготовку приміщення V_3 .

Ці витрати залежать від стану приміщення, де буде встановлюватися ПК.

Так як пристосоване приміщення є, тому:

$$V_3 = 0 \text{ грн.}$$

Витрати на навчання персоналу V_4

В середньому навчання персоналу триватиме 3 дня, тому можна вважати, що:

$$V_4 = 2550_{\text{грн.}}$$

Загальна вартість розробки і впровадження системи

Загальна вартість розробки і впровадження системи V_{Σ} , вираховується:

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4 = 420000 + 19040 + 0 + 2550 = 450879_{\text{ грн.}}$$

Оскільки норма амортизаційних втрат для комп'ютерних систем $N_A = 5$, то щоб обрахувати річний економічний ефект потрібно скористатись цією формулою:

$$V_p = \frac{450879}{5} = 90175.8_{\text{грн.}}$$

Таблиця 6. Основні джерела прибутку від впровадження комп'ютерної системи і порядок його підрахунку.

№	Джерела прибутку	Σ (сумма)
1	Скорочення витрат на друк матеріалів	30483 грн
2	Оптимізація пошуку роботи	23298 грн
3	Зменшення витрат на робоче місце	17658 грн
		71739 грн

Коефіцієнт економічної ефективності можна визначити, таким обчисленням:

$$K_{\text{ЕФ}} = \frac{Pr}{V_p} = \frac{71739}{90175.8} = 0.74$$

Термін окупності розробки можна визначити таким чином:

$$T_{\text{ОК}} = \frac{1}{K_{\text{ЕФ}}} = \frac{1}{0.74} = 1,16$$

З даного розрахунку виходить що термін окупності буде 1 рік і 2 місяці.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Безпека праці

Під час роботи на комп'ютерах працівники постійно знаходяться під впливом електромагнітного поля. Робота на комп'ютері характеризується впливом на організм людини наступних шкідливих факторів:

- підвищений рівень шуму;
- електромагнітне випромінювання;
- електростатичне поле;
- тривале сидяче положення;
- навантаження на зір;
- перевантаження кистьових суглобів;
- можливість захворювань органів дихання;
- алергії.

Особливості роботи користувача ПК характеризуються :

- високою напруженістю зору;
- одноманітною позою;
- малою руховою активністю;
- значним нервово - емоційним навантаженням;
- роботою з великою кількістю інформації.

Для забезпечення, найкращих умов для ефективної і безпечної роботи потрібно створити такі умови праці, що будуть комфортними і максимально зменшують вплив даних шкідливих факторів. Необхідно, щоб перераховані шкідливі фактори узгоджувалися з установленими правилами і нормами.

Шум - це сукупність звуків, що несприятливо впливають на організм людини, що заважають його роботі і відпочинкові.

Вплив навіть незначних рівнів шуму при напруженій розумовій діяльності виражається у зниженні розумовій працездатності, швидкій втомлюваності, послабленні уваги, появі головного болю.

Шум шкідливо впливає на нервову систему і знижує швидкість і точність сенсомоторних процесів, зростає число помилок при рішенні інтелектуальних задач. Шум впливає на увагу людини і викликає негативні емоції.

Основним джерелом шуму в приміщеннях, де знаходяться ПК, є устаткування для кондиціонування повітря, друкована і копіювальна техніка, а в самих ПК вентилятори систем охолодження. Рівень шуму на робочому місці не повинний перевищувати 50 Дб. Цей рівень шуму не є шкідливим для людини, він не викликає шумової хвороби, але викликає передчасну втомлюваність і роздратування. Нормовані рівні шуму забезпечуються шляхом використання звукобірних матеріалів для облицювання приміщень.

Основними мірами боротьби із шумом є:

- усунення або ослаблення причин шуму в самому його джерелі в процесі проектування й експлуатації устаткування;
- ізоляція джерел шуму від навколишнього середовища засобами звукоізоляції і звукопоглинання;
- раціональне планування виробничого устаткування.

Згідно з нормованими параметрами в діапазоні частот 60 кГц - 300Мгц є напруженості Е і Н електромагнітного поля. При частоті від 60 Кгц до 3Мгц напруженість електричного поля може складати до 50В/м, а напруженість магнітного поля до 5А/м.

При роботі комп'ютера генерується електромагнітне випромінювання дуже низької частоти. Поблизу комп'ютера напруженість електромагнітного поля складає від 4 до 7 мілігаусс. Напруженість магнітного поля вище 4 мілігаусс є шкідливою для людини. Таке випромінювання є причиною аномалій при вагітності і викликає зміну на клітинному рівні. Негативно позначається на людському організмі збільшення кількості позитивно заряджених іонів у повітрі. Медичні дослідження, проведені в США, показують, що довгострокове перебування в деіонізованій атмосфері впливають на метаболізм і приводить до зміни біохімічної реакції в крові на клітинному рівні, що нерідко приводить до стресів. При роботі з монітором на

ньому накопичується заряд статичної електрики, що приводить до виникнення електростатичного розряду.

Для профілактики несприятливого впливу електромагнітного поля на користувача необхідно:

- не переобтяжувати приміщення значною кількістю робочих місць з ПК;
- не застосовувати на місці якому працюєш великої кількості радіоелектронних засобів;
- залишати виключеним комп'ютер , якщо на ньому вже не працюють.

Тривале перебування в електростатичному полі може стати причиною бронхо - легневих захворювань, порушень серцево - судинної та нервової систем, уражень шкіри. Такий вплив електростатичного поля є тому, що воно має здатність притягувати пил, бруд та інші частинки які присутні в повітрі навколо монітору.

Для захисту від цього потрібно робити наступне:

- поставити статичні нейтралізатори електрики;
- дотримуватись 55% відносної вологості в приміщенні;
- використовувати побутові зволожувачі;
- застелити підлогу антистатичним лінолеумом і проводити щоденне вологе прибирання;
- протирати екран комп'ютера та предмети біля нього серветкою просоченої антистатичним препаратом.

Для підтримки нормованих значень концентрації позитивних і негативних, іонів у приміщеннях з ПК рекомендується встановлювати кондиціонери, пристрої іонізації повітря або проводити природне провітрювання тривалістю не менш 10 хвилин після кожних 2 годин роботи.

4.2 Умови праці

Найбільш значним фізичним фактором є виробничий мікроклімат, що характеризується рівнем температури і вологості повітря, а також швидкості руху повітря.

Використовувані ПК не вимагають створення особливих мікрокліматичних умов для роботи і нормально функціонують у межах припустимих для людини значень температури і вологості.

Мікрокліматичні умови в приміщенні з ПК повинні задовольняти вимогам у відповідності:

- температура навколишнього середовища в холодний період року 20 - 22 С, у теплий період 22 - 25 С;
- приблизно 40-60% це відносна вологість повітря ;
- повітря повинно рухатись зі швидкістю 0,1 - 0,2 м/с;
- вміст пилу - максимум 0.0001 кг/м при розмірі часток максимум 3 мкм.

Для того , щоб праця була здоровою та продуктивною необхідно забезпечити чистоту повітря . Атмосферне повітря у своєму складі містить у процентному відношенні:

Азот 78,8%

Кисень 20,25%

аргон, неон і інші інертні гази 0,93%

вуглекислий газ 0,03%

Повітря такого складу найбільш сприятливе для подиху людини.

Розглянуте в даному дипломному проекті устаткування і робочі станції в процесі роботи не виробляють ніяких шкідливих речовин. У такий спосіб повітряне середовище в приміщенні, де вони використовується, шкідливих впливів на організм людини не робить і задовольняє вимогам І категорії робіт, згідно .

Всі оптимальні норми , відносної вологості і швидкості руху повітря та температури в зоні на якій працюють на виробничих приміщеннях в нормах і приведені, у таблиці 4.1.

Таблиця 4.1 -Норми температури та вологості

Сезон року	Категорія робіт	Температура, град С	Відносна вологість, %	Швидкість руху повітря, м/з
Холодний Період	I	22 - 24	40 - 60	0,1
Теплий Період	I	23 - 25	40 - 60	0,1

Неприпустимо розташування ПК у підвалах та цокольних приміщеннях. Підлога повинна мати рівне, неслизьке покриття та має бути зручною для різних видів прибирання. Найкращим вважається покриття підлоги антистатичним лінолеумом.

Для забезпечення нормованих значень мікроклімату, вмісту шкідливих речовин, в приміщенні необхідністю є обладнання системаю опалення, кондиціонування повітря, а також вентиляцією.

Виробничі приміщення для роботи з ПК не повинні межувати з такими приміщеннями, в яких рівень шуму і вібрації не в межах норми . Також неприпустимо розташовувати вибухопожежнебезпечні приміщення категорії А і Б, а також «мокрими» процесами виробництва які якимось чином межують з приміщеннями, де знаходяться ПК, а також над ними і під ними.

Місця призначені для роботи з ПК мусять розташовуватися в окремих приміщеннях. Площа на одного працівника повинна становити не менш ніж 6 м^2 , об'єм - не менш ніж 20 м^3 .

Освітлення служить одним з найважливіших факторів, що впливають на продуктивність праці. Рационально улаштоване освітлення на робочих місцях операторів є істотним показником високого рівня культури праці, невід'ємною частиною наукової організації праці й естетики виробництва.

Вимоги до рациональної освітленості виробничих приміщень зводяться до наступного:

- правильний вибір джерел світла і систем освітлення;

- для робочих поверхонь потрібно забезпечити правильний рівень освітленості;

- усунення відблисків та сліпучої дії світла
- дбати про рівномірне освітлення.

Прийнятний рівень освітлення в приміщенні можна знайти, якщо послідовно вирішити дві задачі:

а) визначити необхідний для оператора рівень освітлення робочого місця зовнішніми джерелами світла.

б) якщо необхідний рівень освітленості виявиться неприйнятним для інших, операторів, що працюють у даному, приміщенні, необхідно знайти спосіб збереження необхідного контрасту зображення іншими засобами. Наприклад, можна розподілити світловий потік з урахуванням розташування робочих місць і засобів відображення, інформації.

При проектуванні й організації робочого місця користувача ПК варто ужити заходів по запобіганню прямих і відбитих відблисків.

Прямі відблиски можна зменшити кожним з наступних способів: застосовувати відбите освітлення; користуватися декількома джерелами освітлення меншої потужності замість одного сильного; використовувати засоби екранування прямого світла від очей оператора.

Відбиті відблиски можна зменшити такими способами:

- використовувати розсіяне світло;
- застосовувати матові поверхні;
- розташовувати джерела прямого світла так, щоб кут спостереження

оператором робочої площі не збігався з кутом падіння на неї променів світла від ліхтаря.

Важливою задачею є вибір виду освітлення (природне або. штучне) і відповідно до цього - вибір типу виробничого приміщення (з вікнами або без вікон).

Найбільше сприятливо для людини природне освітлення. При природному висвітленні продуктивність праці робітників вище, ніж при штучному. Для

достатнього природного освітлення приміщень з ПК площа вікон повинна складати не менш 25% від площі приміщення.

Однак, варто враховувати, що застосування природного світла має багато недотягнень: світло, яке надходить, зазвичай, тільки з одної сторони, а також не рівна освітленість в просторі тощо.

Для усунення цих недоліків необхідно застосовувати додаткові пристосування. Застосування подвійного світла (сполучення природного і штучного освітлення) фізіологічно мало ефективно і негативно діє на зір, сприяє передчасному стомленню.

Застосування штучного освітлення допомагає уникати багатьох з розглянутих недоліків і створювати оптимальний світловий режим. Однак, застосування приміщень без вікон створює в ряді випадків у людей почуття скрутності і непевності. Особливо сильно це виявляється в приміщеннях малого обсягу. У великих приміщеннях даний недолік практично відсутній, тому тут кращий застосування штучного освітлення можна застосовувати газорозрядні лампи.

Згідно з величиною освітленості для системи загального освітлення люмінесцентними лампами потрібно щоб вона була в горизонтальній площині не нижче 300 лк. З урахуванням зорової роботи високої точності величина освітленості може бути збільшена до 500 лк.

Щоб виключити попадання відбитих відблисків в очі користувачів ПК, фактура робочої поверхні необхідно щоб була матова або наполовину матова. Коефіцієнт відбиття має бути в нормі: для стін 0,5-0,6; стелі 0,7- 0,8; підлоги 0,3 - 0,5 та інших поверхонь 0,4 - 0,5.

Для фарбування стін в приміщеннях з ПК потрібно використовувати світлі малонасичені кольори. Надто темні чи світлі кольори за екраном ПК втомлюють органи зору. Найкраще підходять нейтральні сіро - зелені кольори, вони не тільки сприятливо впливають на зір, а й знімають загальну втому.

У приміщеннях, де встановлене комп'ютерне устаткування, створені умови, що задовольняють даним вимогам.

Розрахунок освітлення проводиться для наступних його характеристик:

- стеля з білою поверхнею; / таблиця 5.1/
- стіни з зі світлою поверхнею; / таблиця 5.1/
- підлога з темною поверхнею; / таблиця 5.1/
- ширина приміщення 5 м;
- довжина приміщення 10 м;
- висота приміщення 2,5 м;
- встановлюються світильники серії ЛВО 4x18, (4 лампи в світильнику);
- світловий потік лампи $\Phi_{л} = 1150$ лм;
- норми освітленості $E = 300$ лк на рівні 0,8 м від підлоги.

Для розрахунку освітлення необхідно розрахувати площу приміщення за формулою:

$$S = a \cdot b, [m^2]$$

де a - ширина приміщення,

b - довжина приміщення,

$$S = 10 \cdot 5 = 50 (m^2).$$

Далі визначаємо індекс приміщення згідно формули:

$$\varphi = \frac{S}{(h_1 - h_2) \cdot (a + b)},$$

де h_1 - висота приміщення ;

h_2 - висота робочої поверхні (0,8 м),

$$\varphi = \frac{50}{(2,5 - 0,8) \cdot (10 + 5)} = 25,5$$

Визначаємо потрібну кількість світильників за формулою:

$$N = \frac{E \cdot S \cdot K_3 \cdot 100}{U \cdot n \cdot \Phi_{л}} \quad [\text{шт.}]$$

де E - необхідна освітленість горизонтальної площини (300 лк),

S - площа приміщення m^2 ,

K_3 - коефіцієнт запасу ($K_3 = 1,4$),

U - коефіцієнт використання установки яка освітлює,

Фл - однієї лампи світловий потік (1150 лм).

n - число ламп в одному світильнику.

Визначаємо коефіцієнт використання U, виходячи із значень коефіцієнтів віддзеркалення і індексу приміщення.

Значення коефіцієнтів віддзеркалення приведені в таблиці 4.2.

Таблиця 4.2 -Коефіцієнти віддзеркалення

Поверхня	Коефіцієнт віддзеркалення
Площина з матеріалів з високою відбиваністю	80
Площина з білою поверхнею	70
Площина зі світлою поверхнею	50
Площина з сірою поверхнею	30
Площина темно -сірою поверхнею	20
Площина з темною поверхнею	10

Значення коефіцієнта використання вибирається згідно таблиці 4.3.

Таблиця 4.3 - Значення індексу приміщення

Коефіцієнти віддзеркалення поверхонь приміщення	Стеля	50	50	70	70	50	50	30	0
	Стіни	50	30	50	30	50	30	30	0
	Підлога	30	10	20	20	10	10	10	0
Індекс приміщення	0.6	36	28	35	29	33	28	28	23
	0.8	42	33	40	33	38	32	31	26
	1.0	47	37	44	37	41	36	35	29
	1.25	52	41	49	42	45	40	39	33
	1.5	57	45	53	46	49	44	43	37
	2.0	63	51	58	52	54	49	48	42
	3.0	76	65	70	65	64	61	60	55
	5.0	89	75	81	78	74	72	71	68

Враховуючи коефіцієнти віддзеркалення поверхонь приміщення та його індекс вибираємо, що коефіцієнт використання U дорівнює 70. Визначаємо необхідну кількість світильників:

$$N = \frac{300 \cdot 50 \cdot 1,4 \cdot 100}{70 \cdot 4 \cdot 1150} = 6,5 = 7 \text{ [шт.]}$$

Усі технічні міри можливо розділити на дві групи.

Перша група - забезпечує захист від удару електричним струмом людей у ситуації з дотиком струмоведучих частин. До них має відношення:

- стан ізоляції, її контроль, а також контроль ділянок живильної мережі та електротехнічних пристроїв;
- захисні огороження та блок;
- Розташування яке буде оптимальним для устаткування, щоб забезпечити проміжок між струмоведучими частинами;
- Світова та звукова сигналізація для безпеки, плакати що попереджують, нанесення умовних позначень;

- Захищення від того щоб не було переходу високої напруги на сторону низької ;

- застосування понижених напруг 42, 36 і 12 В;

- застосування індивідуальних захисних ізолюючих засобів.

Друга група забезпечує захист від ударом електричним струмом при дотику до корпусу електроустановки у ситуації пробою ізоляції струмоведучих частин. До них має відношення:

- захисне заземлення;

- захисне занулення;

- захисне відключення;

- подвійна ізоляція;

- застосування розділових трансформаторів.

Загальновідомо, що від стану електричної ізоляції струмоведучих частин має залежність надійність і довговічність роботи електротехнічного устаткування. Ушкодження ізоляції в багатьох випадках є одною з головних причин електричних травм, аварій і пожеж. Фізичний зміст ізоляції, як захисної міри, зводиться до того, щоб обмежити струм, що протікає по тілу людини, до тієї величини що буде безпечна.

Надійна ізоляція під впливом багатьох факторів і забезпечується застосуванням, конкретного її типу (робочої, посиленої і подвійної), що відповідають ізоляційних матеріалів, раціональною конструкцією електроустаткування, оптимальними умовами виробничого середовища і правильною організацією профілактики в процесі експлуатації.

Зазвичай, електротехнічне устаткування має робочу ізоляцію, яка повинна витримувати гранично можливі навантаження в умовах експлуатації це: електричні механічні, і теплові навантаження.

Захисне заземлення - це спеціальне з'єднання з землею не струмоведучих металевих частин електричного устаткування , блискавковідводів , розрядників та апаратури. Призначенням захисного заземлення на меті виступає зниження до безпечної величини напруги на корпусі щодо землі, яка виникає на не

струмоведучих частинах електроустановок у випадку якщо станеться замикання на корпус при ушкодженні ізоляції провідників, які несуть робочий струм живлення апаратури.

Отже, при наявності заземлення з опором розтіканню струму малої величини, різко стає меншим струм, який протікає через тіло людини, яка торкнулася пошкодженого заземленого корпусу. До аналогічного висновку можна дійти, якщо розглянути значення опору заземлення корпусів електроустановки при харчуванні від мережі перемінного струму.

4.3 Ергономіка робочого місця

Монітор має відповідати таким технічним вимогам як:

- світіння щоб було не менш яскравим ніж 100 кд/м^2 ;
- для кольорового дисплея необхідний мінімальний розмір світлової крапки не більше $0,1 \text{ мм}$;
- контрастність зображення знака потрібен не менше $0,8$;
- частота кадрового розгорнення не менше 7 кГц ;
- кількість крапок не менше 640 ;
- на екрані висота символів не менше ніж $3,8 \text{ мм}$;
- відстань від очей до екрана необхідна бути порядку $600 - 800 \text{ мм}$;
- яскравість світіння не менш 100 кд/м^2 ;
- розмір який може бути мінімальний світлової крапки не більш $0,1 \text{ мм}$;
- контрастність зображення знаку не менш $0,8$;
- частота кадрового розгорнення не менш 7 кГц ;
- кількість крапок не менш 640 .

В моніторі необхідністю є те, щоб він був обладнаний поворотною площадкою, яка дозволяла б робити його переміщення в горизонталі і вертикалі у межах $130-220 \text{ мм}$ а також мав змогу змінювати на $10-15$ градусів кут нахилу екрана.

Найбільш сприятливе співвідношення яскравості між екраном, його ближнім оточенням та далеким (периферійна зорова зона) складає $5:2:1$. Співвідношення тло-буква - $5:1$. Більш знижений контраст між тлом екрана та

знаком сприяє стомленню зорового аналізатора. Розрізняють прямий контраст, який розраховується для темного об'єкта на світлому тлі, і зворотний контраст, який розраховується навпаки для світлого об'єкта на темному тлі це тому, що суб'єктивна оцінка яскравості сприйманого сигналу залежить від яскравості тла.

Критичною частотою мелькання (КЧМ) називають частоти появи світлового сигналу, при якій він сприймається безупинно як подразник .

Залежить ця частота від розмірів, яскравості та конфігурації зображення. При нормальних умовах ця КЧМ приблизно дорівнює 15-25 Гц, але якщо відбувається зорове стомлення ця величина знижується.

Місце для роботи користувача ПК для виконання робіт сидячи має відповідати вимогам стандартів.

Просторова організація робочого місця повинна забезпечувати оптимальну робочу позу з такими вимогами:

- ступні ніг повинні знаходитись на підлозі чи на підставці для ніг;
- стегна бути в горизонтальній площині;
- плечі - вертикальні;
- кут ліктьового суглобу 70 - 90°;
- зап'ястя зігнуті під кутом не більше 20° від горизонтальної площини;
- нахил вперед голови в межах 15 - 20° до вертикалі.

Головні параметри робочого місця мають бути регульованими тому що цим забезпечується можливість створення оптимальних умов праці людині з урахуванням антропометричних характеристик.

Головні параметри робочого місця, спорядженим персональним комп'ютером:

- сидіння висотою 400 - 500 мм;
- рекомендовані розміри стола: висота 725 мм; ширина 600 - 1000 мм; глибина 800 - 1000 мм;

- стіл мусить мати простір для ніг у висоту не менше 600 мм, ширина не менше 500 мм, глибина на рівні колін не менше 450 мм, на рівні витягнутих ніг - не менше 650 мм;

- на робочому місці повинна бути підставка для ніг, це рекомендується для усіх видів робіт, зв'язаних із довготривалим збереженням положення сидячи ;

- розташування екрану повинно бути зручним для зорового спостереження у вертикальній площині під кутом $\pm 30^\circ$ від лінії зору працівника.

При проектуванні розміщення робочих місць у загальному залі, необхідно враховувати потребу людини в тому щоб зберегти свій персональний простір й зручної дистанції, що залежить від характеру взаємин і від зовнішньої ситуації.

Розміщення робочих місць рекомендується ставити на відстані 215 - 400 см. (далека фаза соціальної дистанції).

Присутність іншої особи на цій ділянці не є чинником що заважає, як на ближчих дистанціях, отже людина може продовжувати працювати, незважаючи на присутність іншої людини. При цьому спілкування не складає труднощів. При проектуванні здвоєних робочих місць повинно опускатись розміщення працівників на відстані 120 - 215 см, щоб відповідало близькій фазі соціальної дистанції.

Системні блоки встановлюються на робочому місці з урахуванням легкої досяжності до рознімань і органів керування на тильній стороні. Принтер зазвичай, розміщають по праву сторону від користувача. Текст розміщений на моніторі, має бути видний операторові в перебуванні його в основній робочій позі. Поруч від принтера в спеціальних відсіках зберігається папір та інші необхідні приналежності.

Сполучні кабелі прокладаються в спеціальних каналах. Пристрій каналів мусить бути таким, щоб сполучні рознімання не були перешкодою витягові кабелів.

Для маніпулятора типу «миша» в праву сторону від користувача на стільниці повинна бути вільна площадка, яка за розміром і формою має бути така ж як і поверхня екрана.

4.4 Пожежна безпека

Для більшості приміщень, обладнаних ПК, встановлена категорія пожежної небезпеки В - (горючі і негорючі рідини, тверді горючі і негорючі речовини і матеріали (у тому числі пил і волокна). Дивлячись на високу вартість електронного устаткування, і ще на категорію пожежної небезпеки приміщень, у яких воно розміщується, будинку для приміщення, оснащених ПК, мають бути I ступеня вогнестійкості по ДБН.

У виробничому приміщенні потрібно щоб були дотримані така правила безпеки:

- входи та виходи з приміщення, а також доступ до засобів пожежогасіння мусять бути постійно вільні;

- устаткування, яке знаходиться в експлуатації, має бути справне і щораз перед початком роботи проходити перевірку;

- по закінченні робіт потрібно оглянути приміщення, знеструмити електромережу та закрити це приміщення.

- Кількість евакуаційних виходів з будинків з кожного поверху і із приміщень повинне нараховувати не менше двох.

- ширину евакуаційного виходу (двері) установити не менш 0.8 м. Пристрій кручених сходів, розсувних і піднімальних дверей, які обертаються і турнікетів на шляхах евакуації не повинно бути. Також забороняється розміщення на сходових клітках будь-які приміщення, прокладати технологічні комунікації та також улаштовувати виходи підйомників і вантажних ліфтів. На евакуаційних шляхах улаштовують як природне, так і штучне аварійне освітлення.

Пожежні крани встановлюють у коридорах, на площадках сходових кліток, у входів, тобто в доступних, помітних місцях. Ручні вуглекислотні вогнегасники

встановлюють у приміщеннях, обладнаних ПК, з розрахунку один вогнегасник на 40 - 50 кв. м площі, але не менш двох у приміщенні.

Для виявлення початкової стадії загоряння й оповіщення служби пожежної охорони використовують системи автоматичної пожежної сигналізації (АПС). Вони можуть самостійно пускати в хід установки пожежогасіння, поки пожежа не досягла великих розмірів.

Об'єкти крім АПС необхідно обладнати установками стаціонарного автопожежогасіння. Доцільно застосовувати установки газового гасіння пожеж, дія яких заснована на швидкому заповненні приміщення вогнегасною газовою речовиною, у результаті чого знижується зміст кисню в повітрі.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досліджено процес пошуку інформації про пам'ятки у місті Вінниця.

Було проведено системний аналіз даної області та створено функціональні моделі AS-IS та TO-BE. На основі цього зроблено висновок, що нинішній рівень автоматизації даного процесу незадовільний та поставлено задачі на автоматизацію.

Розглянуті задачі автоматизації у Розділі 1 були успішно впроваджені. Впроваджено функцію швидкого пошуку туристичного місця; його основна інформація, графік роботи, вартість, також створено навігаційну мапу, яка допоможе скоротити час для того щоб дістатись цього міста. Створений функціонал реалізовано у вигляді мобільного додатку, це дозволяє будь-якому місці з інтернет мережею мати доступ до путівника. Електронний засіб пришвидшив та спростив процес знаходження інформації про пам'ятки і також навігацію до них.

Користувач буде мати основний електронний путівник при подорожі та можливість швидко знайти потрібне місце або переглянути інформацію про нього.

Використання туристами такого електронного путівника дозволить пришвидшити і удосконалити процес пошуку інформації, ознайомитись з інформацією про пам'ятку, прокласти маршрут до потрібного місця.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Apple XCode [Електронний ресурс] - Режим доступу до ресурсу: <https://developer.apple.com/xcode/>.
2. Офіційний сайт Stack Overflow [Електронний ресурс] - Режим доступу до ресурсу: <https://stackoverflow.com/research/developer-survey-2015#tech-super>. Когаловский М.Р. Энциклопедия технологий баз данных. 2002. — 800 с.
3. The Swift [Електронний ресурс] Режим доступу до ресурсу: <https://developer.apple.com/library/watchos/documentation/Cocoa/Conceptual/CoreData/index.html>.
4. Абдикеев Н.М. Проектирование информационных систем (современные технологии): Учебное пособие / Н.М. Абдикеев. - М.: КОС - ИНФ, Рос. экон. акад., 2003. - 140 с.
5. Кен Хендерсон. Професійне керівництво з CoreData: структура та реалізація. 2006. — С. 1056.
6. Башмаков А.И. Интеллектуальные информационные технологии: Учеб. пособие / А.И. Башмаков, И.А. Башмаков. - М.: Изд -во МГТУ им. Н.Э. Баумана, 2005. - 304 с.
7. Офіційна документація фреймворку MapKit [Електронний ресурс] // ANON - Cupertino: Apple Inc., - Режим доступу до ресурсу: https://developer.apple.com/library/ios/documentation/MapKit/Reference/MapKit_Framework_Reference/.
8. Гриньов А. В. Організація та управління на підприємстві. - Харків: Вид. дім "ИНЖЕК", 2004. - 329 с.
9. Бойчик І. М. Економіка підприємства: Навч. посібник. - К.: Атіка, 2004. 480 с.
10. Геврик Є О. Охорона праці. - К.: Ельга; Ніка-Центр, 2003. - 280 с.
11. Жидецький В. Ц. Основи охорони праці. - Львів: Афіша, 2002. - 320 с.
12. Пушкін В. Зарядка для очей ISBN: 978-5-699-50082-6

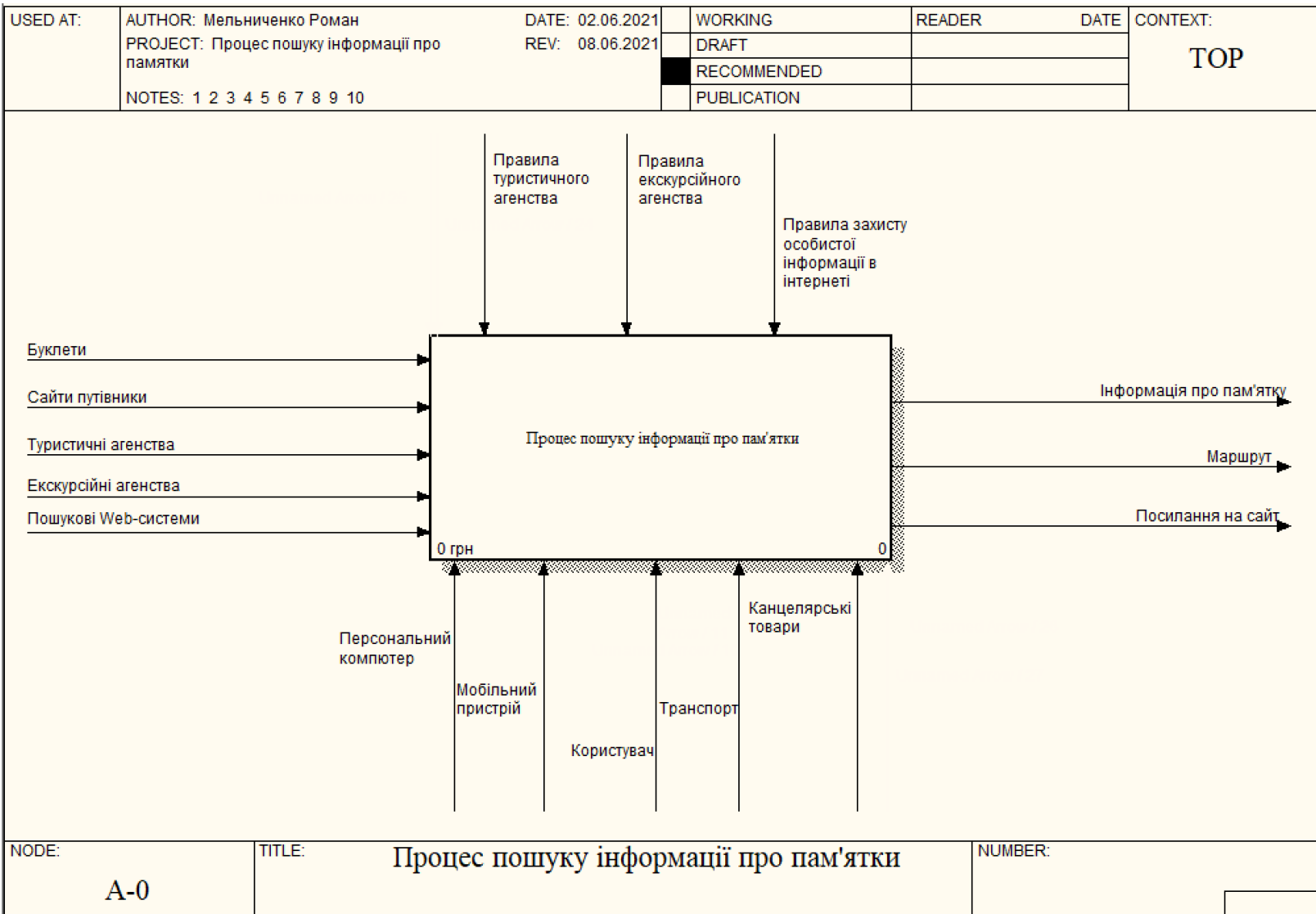


Рис. А.1. Функціональна схема моделі AS-IS.

NODE: A-0	TITLE: Процес пошуку інформації про пам'ятки	NUMBER:
---------------------	--	---------

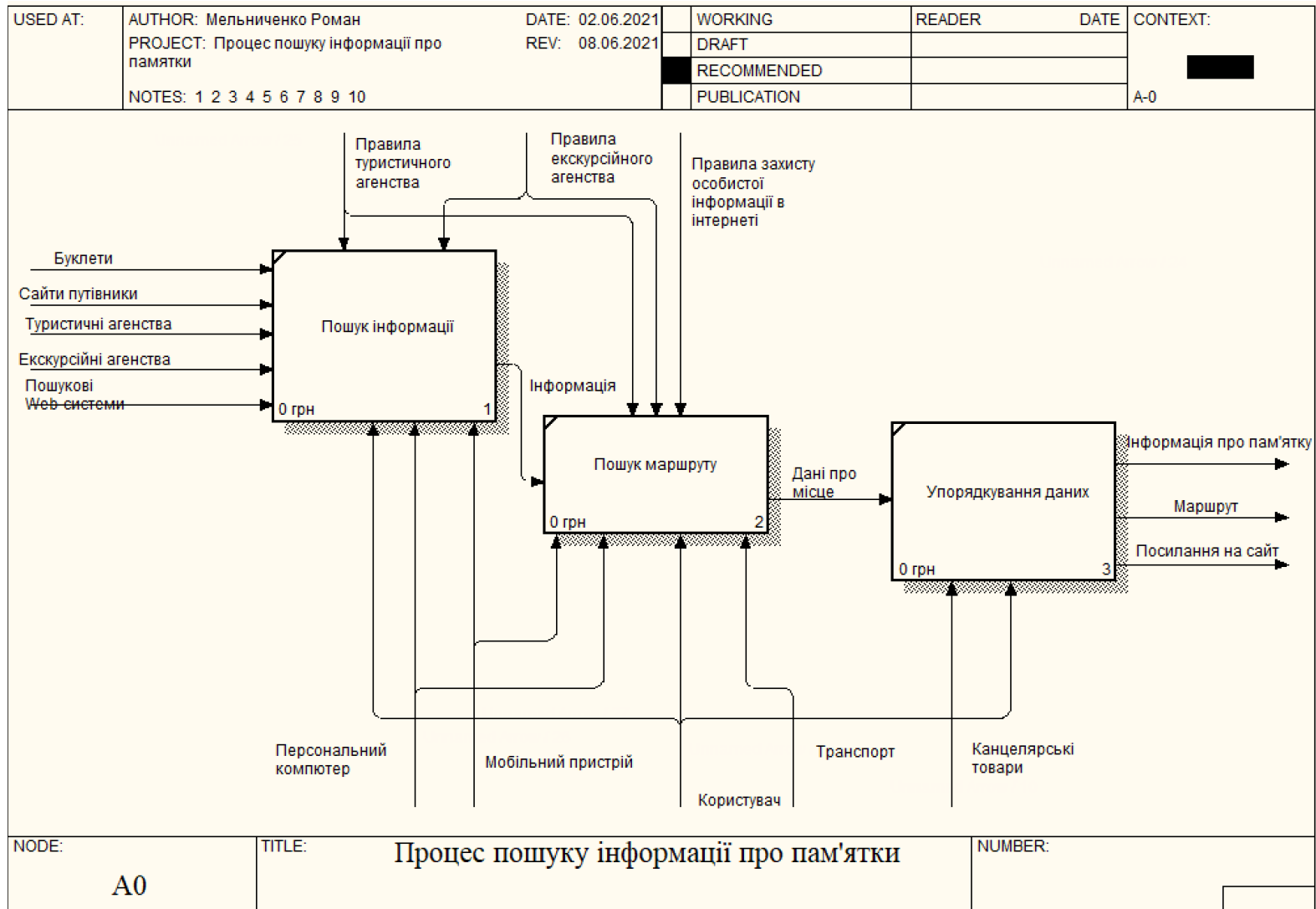


Рис. А.2. Перший рівень декомпозиції моделі AS-IS.

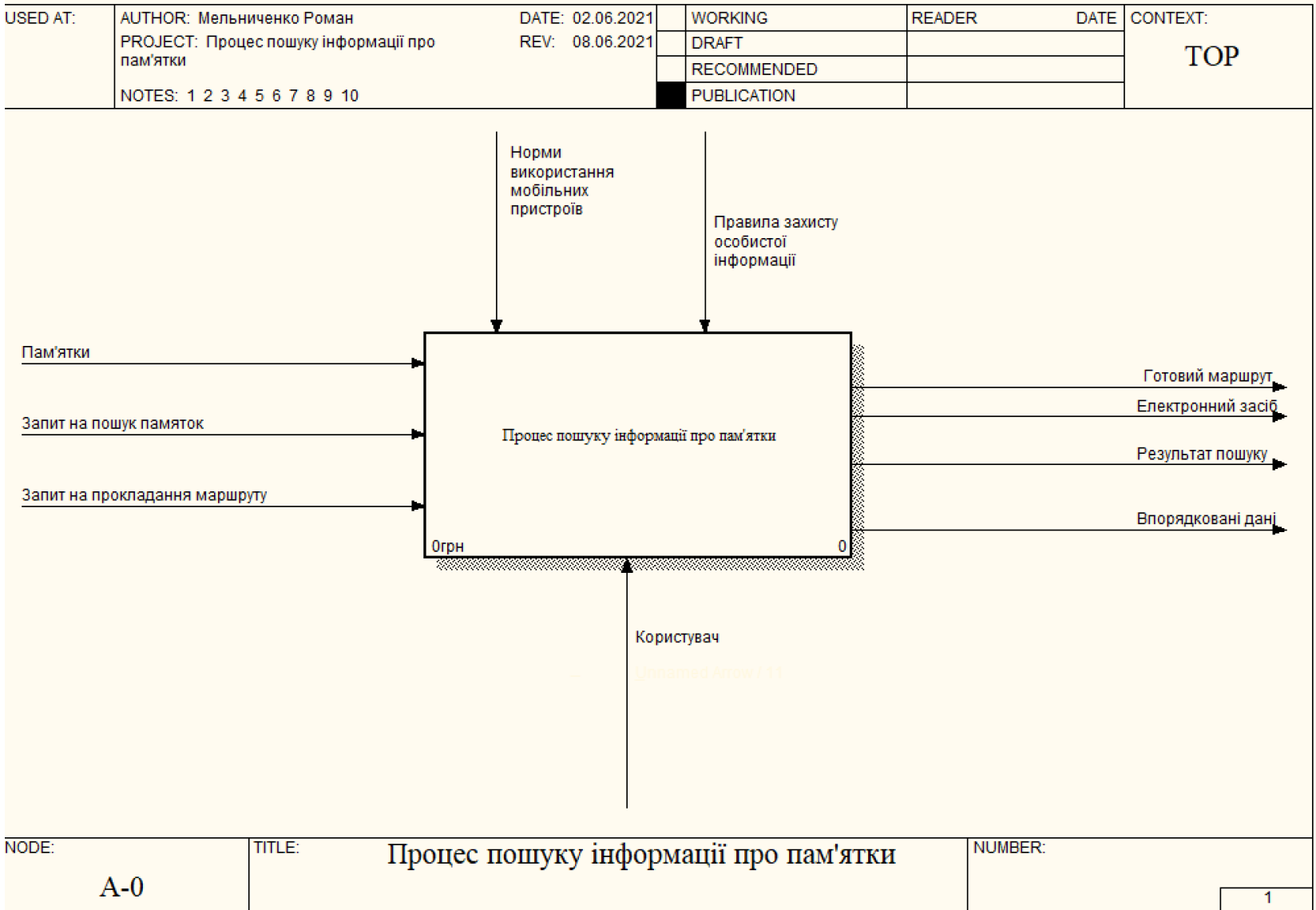


Рис.Б.1 Функціональна модель ТО-ВЕ.

USED AT:	AUTHOR: Мельниченко Роман	DATE: 02.06.2021	WORKING	READER	DATE	CONTEXT: A-0
	PROJECT: Процес пошуку інформації про пам'ятки	REV: 08.06.2021	DRAFT			
	NOTES: 1 2 3 4 5 6 7 8 9 10		RECOMMENDED			
			PUBLICATION			

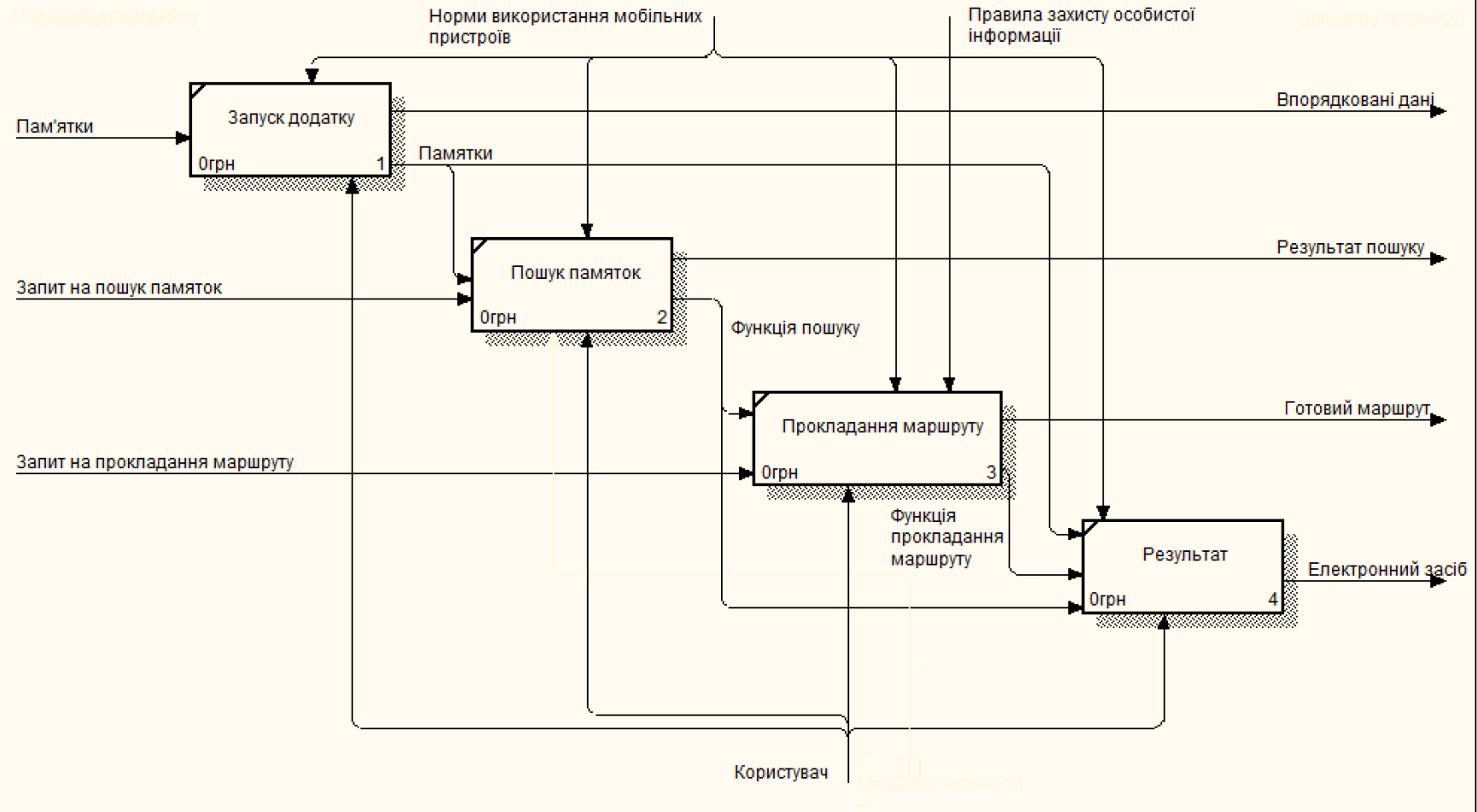


Рис.Б.2 Перший рівень декомпозиції діаграми TO-BE.

NODE: A0	TITLE: Процес пошуку інформації про пам'ятки	NUMBER: 2
--------------------	--	---------------------

ДОДАТОК В «ЛОГІЧНА СХЕМА БАЗИ ДАНИХ»

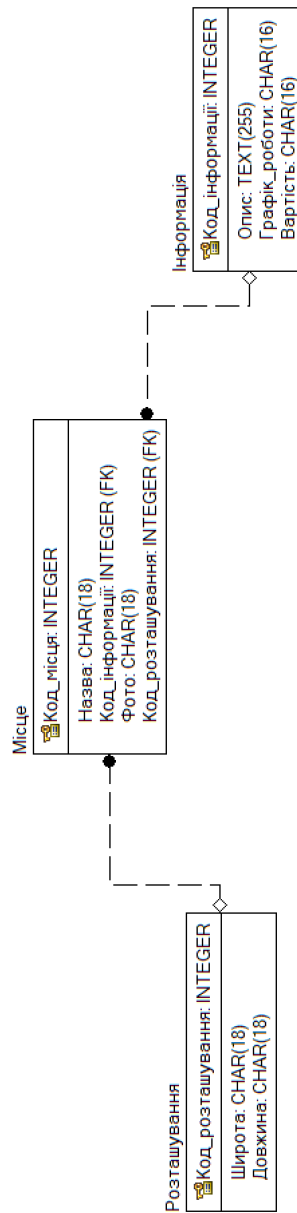


Рис.В.1 Логічна схема бази даних.

ДОДАТОК Г «ФІЗИЧНА СХЕМА БАЗИ ДАНИХ»

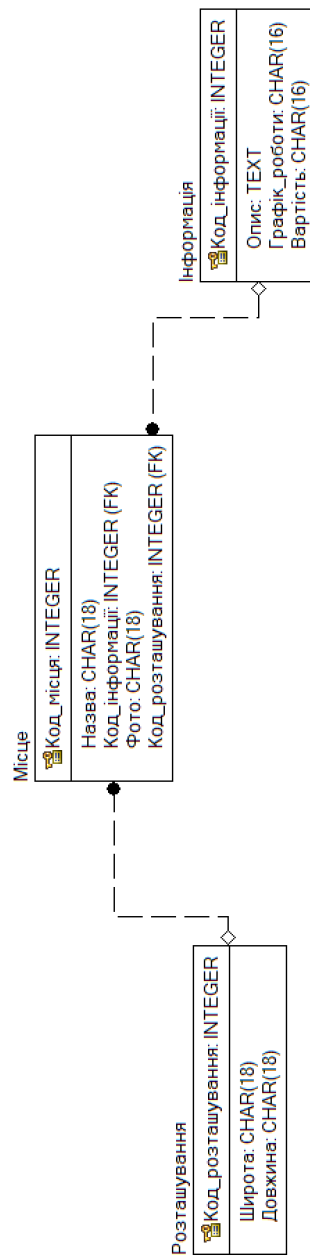


Рис.Г.1 Фізична схема бази даних.

ДОДАТОК Д «ЗГЕНЕРОВАНА БАЗА ДАНИХ У COREDATA»

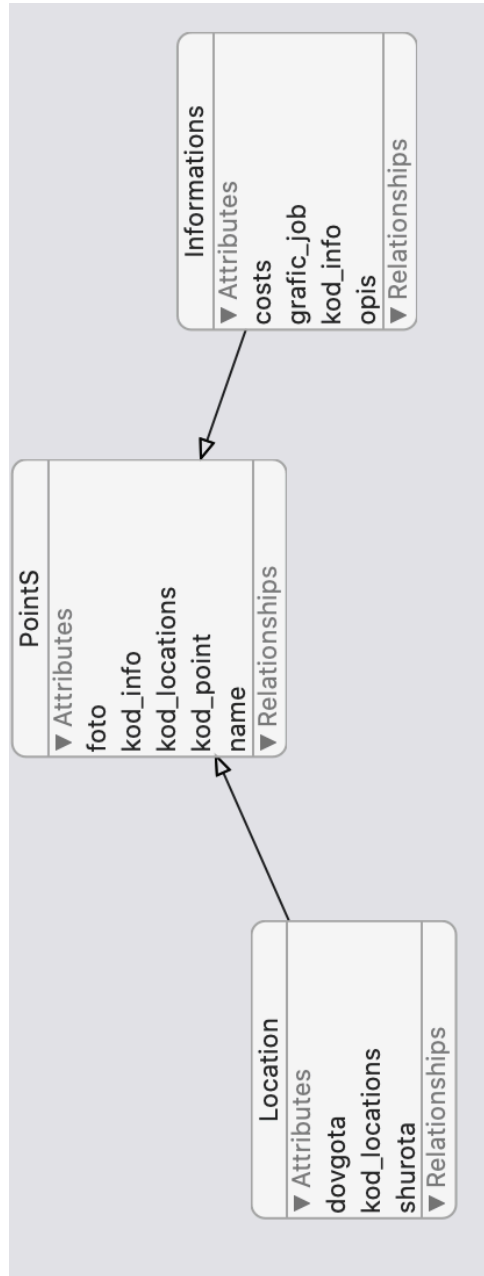


Рис.Д.1 Схема згенерованої БД у CoreData

ДОДАТОК Е «ЗНІМКИ ЕКРАНУ ДОДАТКУ»



Рис.Е.1 Сторінка «Про пам'ятку»

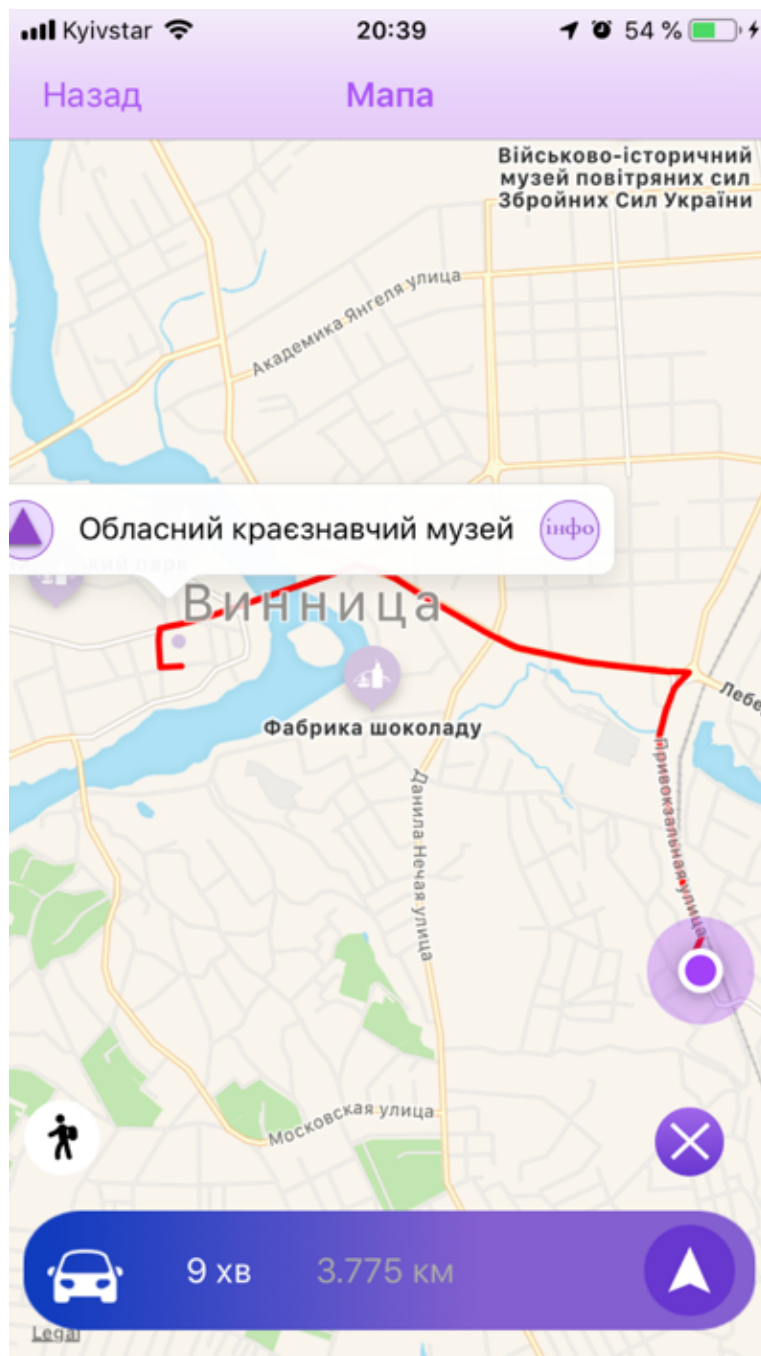


Рис.Е.2 Сторінка «Мапа»

ДОДАТОК Є «ФРАГМЕНТИ КОДУ ПРОГРАМИ»

1. Код методу для налаштування вигляду анотації

```
func mapView(_ mapView: MKMapView, viewFor annotation:
MKAnnotation) -> MKAnnotationView? {
    guard let annotation = annotation as? Artwork else {
return nil }
    let identifier = "marker"
    var view: MKMarkerAnnotationView
    if let dequeuedView =
mapView.dequeueReusableAnnotationView(withIdentifier:
identifier)
    as? MKMarkerAnnotationView {
        dequeuedView.annotation = annotation
        view = dequeuedView
    } else {
        view = MKMarkerAnnotationView(annotation: annotation,
reuseIdentifier: identifier)
        view.canShowCallout = true
        view.calloutOffset = CGPoint(x: -5, y: 5)
        let mapsButton = UIButton(frame: CGRect(origin:
CGPoint.zero, size: CGSize(width: 30, height: 30)))
        // Control State
        mapsButton.setBackgroundImage(UIImage(named:
"MapButton2"), for: UIControl.State())
        view.rightCalloutAccessoryView = mapsButton
        let pointButton = UIButton(frame: CGRect(origin:
CGPoint.zero, size: CGSize(width: 30, height: 30)))
        pointButton.setBackgroundImage(UIImage(named:
"MapButton3"), for: UIControl.State())
        view.leftCalloutAccessoryView = pointButton
        view.glyphImage = UIImage(named: annotation.imageName)
        view.markerTintColor = objcLMP.UIColorFromRGB(rgbValue:
0xD3C3DF)
    }
    return view
}
```

2. Код за допомогою якого працює вікно *mapa*, він прораховує оптимальний маршрут:

```
func rout(firstCoordinate: CLLocationCoordinate2D,
secondCoordinate: CLLocationCoordinate2D, transType:
Transp) {
// show item in point
let firstItem = MKMapItem(placemark:
MKPlacemark(coordinate: firstCoordinate))
let secondItem = MKMapItem(placemark:
MKPlacemark(coordinate: secondCoordinate))
// request
let request = MKDirections.Request()
request.source = firstItem
request.destination = secondItem
request.requestsAlternateRoutes = true
// Путь
switch transport {
case .avto:
request.transportType = .automobile
let directions = MKDirections(request: request)
directions.calculate { (response, error) in
guard let response = response else{
NSLog("Error of requesting
\ (error.debugDescription) ")
return
} // маршрут есть, то берем первый маршрут]
if response.routes.count > 0 {
let route = response.routes.first!
//distance
if route.distance > 1000 {
self.distanceLabel.text = "\ (route.distance/1000) км"
} else {
self.distanceLabel.text = "\ (route.distance) мет"
}
//time
let seconds: Int =
lroundf(Float(route.expectedTravelTime))
let hour: Int = seconds / 3600
let mins: Int = (seconds % 3600) / 60
if hour == 0 {
self.timeLabel.text = "\ (mins) хв"
} else {
self.timeLabel.text = "\ (hour) год \ (mins) хв"
```

```

    }
    // рисуем путь над дорогой
    self.mapView.addOverlay(route.polyline, level:
    .aboveRoads)
    // концентрация над двома точками
let rect = route.polyline.boundingMapRect
self.mapView.setRegion(MKCoordinateRegion(rect), animated:
true)
    }
    }
    case .walking:
    request.transportType = .walking
    let directions = MKDirections(request: request)
    directions.calculate { (response, error) in
    guard let response = response else{
    NSLog("Error of requesting
    \(error.debugDescription)")
    return
    } // маршрут есть, то берем первый маршрут]
    if response.routes.count > 0 {
    let route = response.routes.first!
    //distance
    if route.distance > 1000 {
    self.distanceLabel.text = "\(route.distance/1000) км"
    } else {
    self.distanceLabel.text = "\(route.distance) мет"
    }
let seconds: Int = lroundf(Float(route.expectedTravelTime))
    let hour: Int = seconds / 3600
    let mins: Int = (seconds % 3600) / 60
    if hour == 0 {
    self.timeLabel.text = "\(mins) хв"
    } else {
    self.timeLabel.text = "\(hour) год \(mins) хв"
    }
    // рисуем путь над дорогой
    self.mapView.addOverlay(route.polyline, level:
    .aboveRoads)
    // концентрация над двома точками
let rect = route.polyline.boundingMapRect
self.mapView.setRegion(MKCoordinateRegion(rect), animated:
true)
    }
    }

```