

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

## **Пояснювальна записка**

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: Розробка методу та програмного засобу для VR 3D відображення історичних пам'яток

Виконав: студент II курсу,

групи 2ПІ-18м

спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Гончарук Д.В.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Черноволик Г.

О.

(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. КН Арсенюк І. Р.

(прізвище та ініціали)

ВНТУ – 2019  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра програмного забезпечення  
Освітньо-кваліфікаційний рівень - магістр  
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Романюк О. Н.

“ \_\_\_\_ ” \_\_\_\_\_ 2019 року

## ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Гончаруку Денису Валентиновичу

1. Тема роботи – розробка методу та програмного засобу для VR 3D відображення історичних пам'яток.

Керівник роботи: Черноволик Галина Олександрівна, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від “ \_\_\_\_ ” \_\_\_\_\_ 2019 року № \_\_\_\_

2. Строк подання студентом роботи

---

3. Вихідні дані до роботи: Операційна система – Linux; середовище розробки – Visual Studio Code; мови програмування – JavaScript, NodeJS.

4. Зміст розрахунково-пояснювальної записки: аналіз технологій відображення реальності з використанням VR, аналіз інтерфейсу взаємодії у VR, аналіз існуючих аналогів, опис підходів до застосування технології віртуальної реальності у веб-додатках, розробка методу моніторингу та отримання даних про геолокацію, розробка методики конвертації 3D моделей, проектування моделей веб-ресурсу, проектування БД, опис архітектури та даних програми, розробка інструкції користувача, тестування роботи програмного додатку, економічна частина.

5. Перелік графічного матеріалу: тема, автор, науковий керівник магістерської кваліфікаційної роботи; мета, об'єкт та предмет дослідження; аналіз предметної області; порівняльний аналіз аналогів; структура програмного додатку; алгоритм програмного додатку; обґрунтування вибору програмного середовища розробки; обґрунтування вибору мови програмування; результати роботи.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Черноволик Г. О., к.т.н, доцент кафедри ПЗ		
4	Бальзан М. В., к.е.н., доцент кафедри ЕПВМ		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасного стану питання та постановка задач дослідження		Вик.
2	Розробка методу відображення 3D моделей у веб-додатку на основі отриманої із соцмережі геолокації		Вик.
3	Опис реалізації рішення та тестування розробленого програмного ресурсу		Вик.
4	Економічна частина		Вик.

Студент \_\_\_\_\_  
( підпис )

**Гончарук Д.В.**  
(прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_  
( підпис )

**Черноволик Г. О.**  
(прізвище та ініціали)

## **АНОТАЦІЯ**

Магістерська кваліфікаційна робота присвячена визначенню останньої відзначеної користувачем геолокації шляхом отримання відповідної інформації з його Instagram-акаунту, для подальшого відображення користувачеві 3D-моделей історичних пам'яток відносно отриманої області виведеної на карту, а також підвищенню швидкодії шляхом конвертації розроблених «важких» 3D-моделей будівель в json-формат, а також збільшення зручності в користуванні шляхом створення клієнтоорієнтованого web-ресурсу.

Програмний додаток реалізований за допомогою середовища розробки Visual Studio Code, мовами розробки JavaScript та NodeJS.

## **ANNOTATION**

Master's qualification work is devoted to determining the last user-marked geolocation by obtaining relevant information from his Instagram account, for further display to the user of 3D models of historical monuments in relation to the obtained area displayed on the map, improving performance by converting developed "heavy" 3D models of buildings into json format, as well as increased ease of use by creating a client-oriented web resource.

The software application is implemented using the development environment Visual Studio Code, in the language of development - JavaScript, NodeJS.

## ЗМІСТ

\_Тос26949800

ВСТУП.....	8
1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ.....	14
1.1 Класифікація технологій відображення реальності.....	14
1.2 Класифікація способів відображення з використанням VR.....	19
1.2.1 Властивості VR.....	19
1.2.2 Типи VR.....	20
1.2.3 Типи устаткування для VR-відображення.....	21
1.2.4 Області застосування VR-технологій.....	24
1.3 Інтерфейс взаємодії у VR.....	25
1.4 Аналіз аналогів.....	27
1.5 Висновки до розділу 1.....	29
2 РОЗРОБКА МЕТОДУ ВІДОБРАЖЕННЯ 3D МОДЕЛЕЙ У ВЕБ-ДОДАТКУ НА ОСНОВІ ОТРИМАНОЇ ІЗ СОЦМЕРЕЖІ ГЕОЛОКАЦІЇ.....	30
2.1 Підходи до застосування технології віртуальної реальності у веб-додатках.....	30
2.2 Розробка методу моніторингу та отримання даних про геолокацію.....	33
2.3 Розробка методики конвертації 3D моделей.....	36
2.4 Проектування моделей веб-ресурсу.....	41
2.4.1 Концептуальна модель.....	41
2.4.2 Логічна модель.....	42
2.4.3 Фізична модель.....	44
2.5 Проектування БД.....	51
3 ОПИС РЕАЛІЗАЦІЇ РІШЕННЯ ТА ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАСОБУ.....	52
3.1 Опис архітектури та даних програми, вимоги до ПЗ і АЗ.....	52
3.2 Опис засобів реалізації.....	54
3.2.1 Обґрунтування вибору середовища розробки.....	55
3.2.2 Обґрунтування вибору мови програмування.....	57

3.3 Розробка інструкцій із використання додатку.....	60
3.3.1 Інструкція з підключення пристрою віртуальної реальності.....	60
3.3.2 Інструкція користувача.....	67
3.4 Тестування роботи програмного додатку.....	69
4 ЕКОНОМІЧНА ЧАСТИНА.....	72
4.1 Оцінювання комерційного потенціалу розробки.....	72
4.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько–технологічної роботи.....	76
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.....	80
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності.....	82
4.5 Висновки до розділу 4.....	85
ВИСНОВКИ.....	86
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	88
ДОДАТКИ.....	91
Додаток А. Технічне завдання.....	92
Додаток Б. Акт впровадження.....	96
Додаток В. Лістинг.....	97
Додаток Г. Ілюстративний матеріал.....	107

## ВСТУП

**Обґрунтування вибору теми дослідження.** Соціальна мережа — соціальна структура, утворена індивідами або організаціями. Вона відображає різноманітні зв'язки між ними через різноманітні соціальні взаємовідносини. Але, в Україні, на відміну від західних країн, поки недооцінюють наскільки впливовою може бути дана структура в соціальному чи бізнес аспекті.

Більшість свого часу ми проводимо у різних соціальних мережах. Соціальні мережі настільки сильно увійшли в наше повсякдення та соціальну культуру, що ми практично проживаємо частину свого життя там і навпаки, реальне життя переноситься та документується у соціальній мережі. Зараз можна практично повністю вивчити людину за одним лише профайлом у соцмережі. [12]

Ще одним свіжим віянням останніх років є віртуальна реальність, яка виникла сама собою і неочікувано, звісно не без допомоги сучасних технологій, викликала пошвавлене зацікавлення у суспільства. Віртуальна реальність занурює людину в тривимірне оточення за допомогою різноманітних спеціальних шоломів, окулярів або інших систем, призначених для цього, тому часто використовується термін iVR (immersive VR — віртуальна реальність із зануренням).

Ці технологія на сьогоднішній день перетворилася в корисний засіб, що застосовуються в декількох областях, наприклад:

- навчання в віртуальних симуляторах є на порядок ефективнішим за реальну практику: можна з високою точністю керувати імітованим середовищем, переглядати повтори та безбоязно відпрацьовувати реальні маніпуляції в безлічі потенційно небезпечних ситуацій;

- у всіх провідних автовиробників є VR-системи для тестування дизайну та ергономіки продуктів, які ще не вийшли у виробництво, що дає змогу швидше перебирати різні варіації у порівнянні з реальними макетами. Це ж



застосовується і у виробництві катерів, літаків, тракторів, виробничих ліній, і навіть фабрик;

- цифрові моделі виглядають дуже правдоподібно: можна з усіх боків розглянути свій майбутній будинок або оцінити міське планування задовго до початку будівництва;

- VR — корисний засіб для дослідження ринку в області роздрібно́ї торгівлі: до прикладу, можна наживо поглянути на зовнішній вигляд свого магазину до його будівництва або перенесення, простежити за рухом відвідувачів і за напрямком їх погляду;

- віртуальна реальність — це хороший спосіб лікування фобій: при остраху висоти можна перенестися на змодельований стрімчак, відчути та перебороти свій страх;

- також віртуальна реальність є підґрунтям для подальшого розвитку туристичних центрів. За допомогою цієї технології можливі віртуальні тури та демонстрації.

Останній пункт є дуже важливим і цікавим напрямом, що потребує детального дослідження та, як наслідок, вдосконалення. У тому темпі, в якому розвивається наше місто, розробка програмних продуктів, що дозволять краще з ним ознайомитися, будуть доречними та перспективними. А отже, розробка, орієнтована на використання у сфері туризму, у навчальних цілях (до прикладу, під час уроків історії), на людей, які цікавляться історією та пам'ятками архітектури для свого саморозвитку, та просто на активних користувачів соцмереж (яких зараз переважна більшість), матиме великий попит.

Учасник віртуального туру є лише пасивним глядачем. Але головна перевага таких віртуальних турів — їх доступність для всіх користувачів мережових технологій та відсутність необхідності будь-яких істотних додаткових витрат. Цільова аудиторія для таких віртуальних турів є надзвичайно широкою, зокрема малозабезпечені верстви населення. Фактори доступності й дешевизни також стали визначальними для створення веб-ресурсу відображення історичних пам'яток.

Віртуальні історичні пам'ятки являють собою інтегративну культурну форму, що характеризується структурною та функціональною різноманітністю. З одного боку, це нова технологія безконтактної інформаційної взаємодії користувача з музейним середовищем, комп'ютерна система, яка забезпечує певні візуальні ефекти. З іншого — віртуальне відображення 3D пам'яток як інформаційний ресурс має сутність медіа: це і засіб масової комунікації, і нова форма видавництва. Крім того, це вагомий елемент сучасного культурного середовища, що сприяє розвитку творчих здібностей, культурному обміну та спілкуванню користувачів.[18]

Створений програмний засіб надаватиме користувачам безоплатний доступ до визначних та цікавих місць. Він є каналом поширення культурних цінностей і власне долучення населення до культури.

Поширені трактування технічних можливостей віртуальної реальності зазвичай передбачають значно вищу активність учасника.

Упровадження комп'ютерних технологій набуло в туризмі особливого значення, завдяки чому можна говорити про початок так званої дигітальної революції в туристському секторі, яка відкриває для мандрівників нові можливості.

Футурологи прогнозують, що гаджети, які нагадуватимуть нинішні шоломи віртуальної реальності, стануть звичайним явищем і під час віртуальних подорожей, які за сприйняттям людиною на рівні свідомості та фізичних відчуттів майже не відрізняються від реальних. [19]

Переважає більшість фахівців туристичної індустрії впевнені, що розвиток віртуального туризму стане одним із напрямів туристичної індустрії. Нині ж, віртуальні тури в туристичній галузі слугують, передусім, засобами реклами і просування.

Віртуальні тури — це своєрідне переміщення в часі — у майбутні та минулі реальні подорожі. Якщо людина планує подорож, їй цікаво поглянути на визначні пам'ятки, які вона зможе скоро побачити і вирішити чи варто прокладати свій маршрут через деякі з них. Але цікавіший ефект віртуальних

турів — оновлення вже пережитих і, можливо, забутих вражень та їх посилення.

Згідно з дослідженням організації Harris Interactive, одним із найважливіших факторів під час вибору місця відпочинку або відвідування є саме візуальна інформація. 73% опитуваних відповіли, що вибирають місце відпочинку або туризму в Інтернеті. 69% вважають найважливішою інформацією, що необхідна для прийняття остаточного рішення, є саме візуальна — фотографії, віртуальні тури або відео. 59% опитаних назвали 3D-сферичні панорами та віртуальні тури корисними під час вибору місця відвідування або відпочинку. [9]

Таким чином, віртуальність стає дедалі більшою реальністю. Віртуальний досвід допомагає наблизитися до реальності й дозволяє проходити такими місцями і побачити такі речі, які не є доступними в іншому разі.

**Зв'язок роботи з науковими програмами, планами, темами.** Робота виконувалась згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

**Мета та завдання дослідження.** Метою роботи є покращення точності визначення останньої відзначеної користувачем геолокації шляхом отримання відповідної інформації з його Instagram-акаунту, для подальшого відображення користувачеві 3D-моделей історичних пам'яток відносно отриманої області виведеної на карту, підвищення швидкодії за рахунок конвертації розроблених «важких» 3D моделей будівель в json-формат, а також збільшення зручності в користуванні шляхом створення клієнтоорієнтованого web-ресурсу.

Основними завданнями дослідження є:

- визначити сутність та значення сучасних методів представлення 3D об'єктів, їх переваги та недоліки;
- проаналізувати наявні методи та технології моніторингу геоданих користувачів;
- розробити метод моніторингу та отримання даних про геолокацію із соцмережі для відображення 3D моделей у веб-додатку;

- розробити методику конвертації 3D моделей;
- спроектувати концептуальну, логічну та фізичну моделі веб-застосунку й описати принцип роботи розроблених засобів системи;
- спроектувати базу даних, що міститиме 3D-моделі історичних пам'яток;
- розробити інструкцію користувача із підключення та використання ресурсу.

**Об'єкт дослідження** — процес побудови та відображення 3D-моделей історичних пам'яток із застосуванням технології віртуальної реальності на основі отриманих координат для створення веб-ресурсу.

**Предмет дослідження** — методи моніторингу та отримання даних про останню відзначену геолокацію, отриману через мережу Instagram.

**Методи дослідження** — теорія 3D моделювання для побудови математичних моделей об'єктів, стандарт графічного інтерфейсу OpenGL для вбудовування в браузер повноцінної 3D графіки, комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних досліджень, методи відображення віртуальної реальності для рендеру 3D моделей.

Наукова новизна отриманих результатів.

1) Подальшого розвитку отримав метод моніторингу та отримання даних про останню відзначену геолокацію, який відрізняється від існуючих використанням географічних координат, отриманих через мережу Instagram, що дозволяє розподіляти на карті усі доступні у визначеному радіусі 3D-об'єкти з оптимальною швидкодією.

2) Розроблено методику конвертації створених «важких» 3D моделей будівель, яка відрізняється від існуючих застосуванням json-формату, що дозволяє підвищити продуктивність веб-ресурсу.

**Практичне значення одержаних результатів** полягає в тому, що на основі теоретичних досліджень і отриманих наукових результатів розроблено алгоритм та програмні модулі для клієнтоорієнтованого web-ресурсу з визначенням останньої відзначеної користувачем геолокації та подальшого відображення 3D-моделей історичних пам'яток відносно отриманої області.

**Впровадження.** Впровадження результатів досліджень підтверджуються відповідними актами та використовуються на такому підприємстві: компанія «ТОВ Екзістек» для забезпечення працівників організації ресурсом, що гарантуватиме цікаве дозвілля, культурний розвиток та ознайомлення з архітектурними пам'ятками рідного міста, для можливості удосконалення існуючих та створення нових методів та втілення ідей компанії для незалежної реалізації продукту на різних ринках.

**Особистий внесок здобувача.** Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто.

**Апробація.** Результати роботи було представлено на щорічній конференції НТКП ВНТУ (м. Вінниця, 2018) та на Міжнародній науково-практичній Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ» (9-10 грудня 2019 року).

**Публікації.** Основні результати досліджень було опубліковано у тезах доповіді [12].

**Структура та обсяг роботи.** Магістерська кваліфікаційна роботи складається зі вступу, чотирьох розділів, висновків, списку літератури, що містить 31 найменувань, 4 додатки. Робота містить 38 ілюстрацій, 8 таблиць.

# 1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

## 1.1 Класифікація технологій відображення реальності

Доповнена реальність (augmented reality, AR) — результат додавання в поле сприймання будь-яких сенсорних даних для доповнення відомостей про навколишнє середовище та покращення сприйняття інформації.

Віртуальна реальність (virtual reality, VR) — це технічно створений світ, що передається людині через органи чуття: зір, слух, дотик та інші. Віртуальна реальність наслідує як вплив, так і реакції на нього. Віртуальна реальність (VR) означає сприйняття на комп'ютерах речей, які насправді не існують. Із цього простого визначення ідея не звучить особливо ново.

Інакше кажучи, віртуальна реальність — це:

1) Правдоподібно: справді потрібно відчувати, що перебуваєш у своєму віртуальному світі (на Марсі чи де завгодно), і продовжуєш вірити в це, інакше ілюзія віртуальної реальності зникне.

2) Інтерактивна: коли ви рухаєтесь, світ VR повинен рухатися разом із вами. Ви можете переглянути 3D-фільм і транспортуватися до Місяця чи до морського дна, але це не є інтерактивним у будь-якому сенсі.

3) Генерована комп'ютером: Це важливо ,тому що лише потужні машини, що мають реалістичну 3D-комп'ютерну графіку, досить швидкі, щоб створити правдоподібні, інтерактивні, альтернативні світи, які змінюються в режимі реального часу, по мірі їх переміщення.

4) Досліджуваний: світ VR повинен бути достатньо великим і детальним, щоб можна було його вивчити. Яка б реалістична картина не була, вона показує лише одну сцену, з однієї точки зору. Книга може описати величезний і складний «віртуальний світ», але реально досліджувати його можна лише лінійним шляхом, саме так, як описує його автор.

5) Занурення: щоб бути правдоподібним та інтерактивним, VR повинен залучати як тіло, так і розум.

Для повного відчуття VR потрібні три речі. По-перше, правдоподібний і насичений деталізацією віртуальний світ для дослідження; комп'ютерна модель або імітація, іншими словами. По-друге, потужний комп'ютер, який може в режимі реального часу визначати, що ми робимо, і відповідно коригувати свій досвід (тому те, що ми бачимо чи чуємо, змінюється так само швидко, як ми рухаємось — як і у реальній реальності). По-третє, апаратне забезпечення, пов'язане з комп'ютером, повністю занурює нас у віртуальний світ, коли ми блукаємо. Зазвичай нам потрібно вдягнути так званий головний дисплей (HMD) з двома екранами та стереозвуком, і надіти одну або декілька сенсорних рукавичок. Крім того, ми могли б пересуватися всередині кімнати, обладнаної динаміками об'ємного звуку, на які зовні проєкціюються змінні зображення. [1]

Віртуальна реальність та доповнена — це кардинально різні речі. Їх принципова різниця полягає у тому, що віртуальна створює абсолютно новий світ, а доповнена, в свою чергу, тільки додає окремих штучних елементів до сприйняття реального світу.

Доповнена реальність — це сприйнята змішана реальність (mixed reality), яка створюється з використанням «доповнених» за допомогою комп'ютера елементів сприйняття реальності. Тобто терміни «доповнена» і «змішана» реальність трактуються як синоніми.

По-перше, в 1994 році Пол Мілграм і Фуміо Киширо дали визначення змішаній реальності (mixed reality, MR) як «...все між крайнощами віртуального континууму (VC), де віртуальний континуум простягається від повної реальності до повністю віртуального оточення з доповненими реальністю та віртуальністю всередині нього».

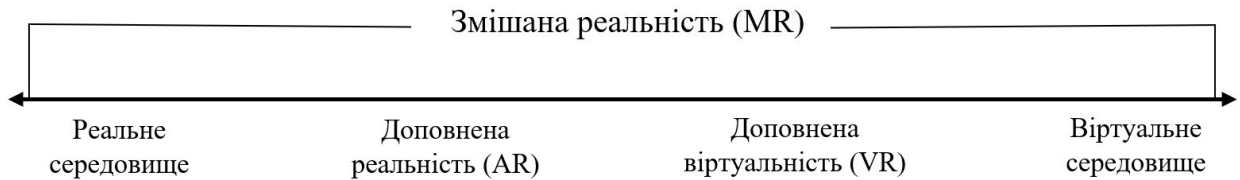


Рисунок 1.1 — Схематичне зображення видів реальності

Тобто, доповнена реальність даній ідеї стає лише частиною змішаної реальності, яку ще й називають гібридною реальністю (hybrid reality).

По-друге, при описі змішаної реальності рідко використовується більш загальна концепція: комп'ютерно-опосередкована реальність (computer-mediated reality), або просто опосередкована реальність. Множина віртуальної реальності являє собою горизонтальну вісь на площині опосередкованої реальності. Вертикальна вісь відповідає множині опосередкування (її також називають медіальною віссю). [32] До цієї множини, зокрема, належить знижена реальність (diminished reality), що реалізується за допомогою шоломів або окулярів, які блокують рекламу з реального світу або замінюють її на корисну інформацію.



Рисунок 1.2 — Схематичне зображення зв'язків між видами реальностей



На щорічній конференції Google I/O, що проходила в травні 2018 року в Маунтін-В'ю, штату Каліфорнія, глава підрозділу AR і VR Клей Бейвор зазначив: «VR/MR/AR/RR — не окремі визначені речі. Це — зручні ярлики для різних точок спектру». (Абревіатура RR означає real reality — реальну дійсність. Напевно, цей термін потрібно вводити офіційно, враховуючи велику кількість реальностей, породжених інформаційними технологіями).

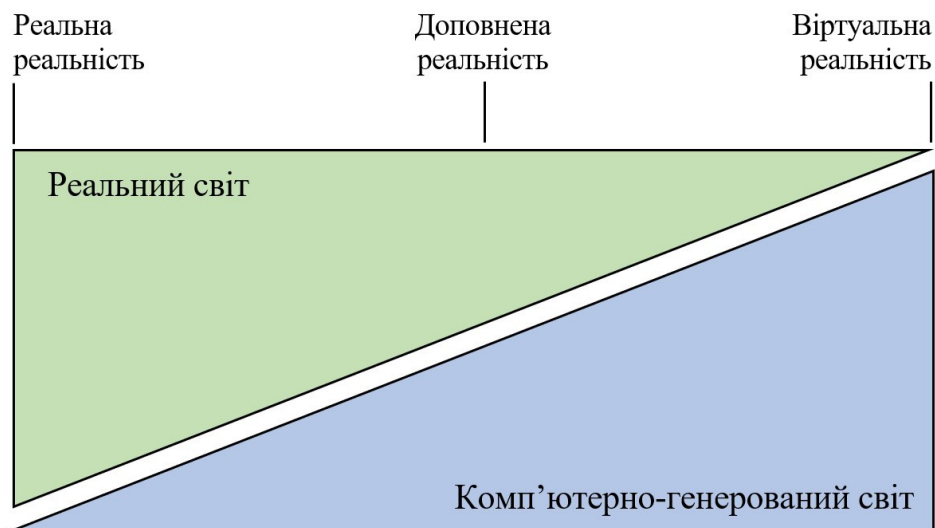


Рисунок 1.3 — Спектр імерсивних інформаційних технологій

До свого висловлювання, Клей Бейвор додає: «Якщо VR і AR — дві точки спектру, то що ми маємо називати спектром? Ось кілька ідей: імерсивні обчислення, обчислення з присутністю, фізичні обчислення, перцепційне обчислення, змішана реальність або імерсивна реальність. Ця технологія зароджується, і потрібно ще довго вибирати визначення, але зараз давайте назвемо це імерсивними обчисленнями (immersive computing)».

Але скептики вважають, що якщо дивитись в будь-якому напрямку — це не означає зануритися у віртуальну реальність. Маркетологи, які застосовують термін VR по відношенню до 360°-камер — недбало поводяться з лексикою.

В лабораторії Японського інституту RIKEN створили «систему заміщення реальності», що з'єднує в свідомості людини вигадку та реальність таким чином, що їх буде неможливо відрізнити один від одного. На даному етапі технології даний термін можна вважати синонімом до віртуальної реальності, але в скорому майбутньому може бути створена. [4]

Simulated reality (модельована реальність) — це гіпотетична реальність, яка нічим не відрізняється від реальності. У змодельованій реальності можуть існувати розумні сутності, які можуть або не можуть усвідомлювати, що вони живуть у симуляції. Це сильно відрізняється від сучасної концепції (технологічно доступної віртуальної реальності). VR легко відрізнити від дійсності, оскільки занурені в неї учасники ніколи не сумніваються у тому, вони переживають. Моделювати реальність, навпаки, буде складно та неможливо відділити від того, що реально відбувається.

Багато термінів, що позначають один і той же, викликано тим, що кожна компанія придумує власну назву для технологій доповненої реальності. Деякі користувачі вважають, що всі «змішані» та «гібридні» реальності можна частково, називати AR. Інші ж, вважають їх різними. Наприклад, термін «змішана реальність», запропонована ще в 1994 році, згодом відійшов на другий план, поступившись місцем доповненій і віртуальній реальностям. Але в останні роки компанія Microsoft почала активно використовувати «змішану реальність» для просування на ринку свого продукту.

А тепер деякі експерти (і постачальники обладнання) поділяють найбільш вживані терміни в такий спосіб:

Доповнена реальність — проектування будь-якої цифрової інформації (зображення, відео, текст, графіка і т.д.) поверх екрану будь-яких пристроїв. У результаті реальний світ доповнюється штучними елементами та новою інформацією. Може бути реалізована за допомогою додатків для звичайних смартфонів і планшетів, окулярів доповненої реальності, стаціонарних екранів, проєкційних пристроїв та інших технологій.[20]

Змішана реальність — проектування тривимірних віртуальних об'єктів або голограм на фізичний простір. Дає змогу переміщатися навколо віртуального об'єкта, оглядати його з усіх боків і, при необхідності, всередині. Вимагає, як правило, спеціального обладнання (очок або шоломів). Віртуальні образи привносяться в фізичний простір, візуалізуються та розташовуються відповідно предметів реальності так, щоб вони сприймалися як реально

існуючі. Людина продовжує взаємодіяти з фізичним світом, в якому, водночас, присутні віртуальні об'єкти, іноді які практично неможливо відрізнити від реальних.

Віртуальна реальність — занурення людини в задалегідь змодельований світ і часткове (а в перспективі повне) її ізолювання від світу фізичного. Для цього використовуються шоломи віртуальної реальності й інші спеціальні пристрої. Розробники прагнуть імітувати взаємодію з створюваною ними псевдореальністю шляхом впливу на всі наявні у людини органи чуття. Вже освоєно вплив на зір і слух, ведуться роботи зі створення рукавичок і навіть костюмів, які забезпечують відчуття дотику об'єктів у віртуальній реальності.

Мабуть, саме ці три терміни і будуть найбільш популярними протягом найближчих декількох років.

## **1.2 Класифікація способів відображення з використанням VR**

Інтерес до VR сильно виріс за останні 2-3 роки і продовжує зростати, з'являється все більше різного обладнання та технологій, а головне — нових ідей, для реалізації яких потрібні розробники.

У віртуальній реальності за допомогою комп'ютера генерується тривимірне середовище, з яким користувач може взаємодіяти, повністю або частково в нього занурюючись.

### **1.2.1 Властивості VR**

У процесі створення віртуальної реальності треба звертати увагу на деякі особливості, хоча усі відразу зустріти практично неможливо:

- правдоподібність — підтримання у користувача відчуття правдоподібності того, що відбувається;
- інтерактивність — забезпечення співдії з середовищем;
- машинно генерується — засновування на потужному АЗ;

- доступність для вивчення — надавання здатності опрацьовувати великий деталізований світ;
- створення ефекту присутності — залучення до процесу не лише мозку, але й тіла користувача, впливаючи на найбільш можливу кількість органів відчуття.

### 1.2.2 Типи VR

До основних типів віртуальної реальності можна віднести:

1) VR з ефектом повного занурення.

Цей тип має на увазі наявність трьох чинників:

- правдоподібна симуляція світу з високим ступенем деталізації;
- високопродуктивний комп'ютер, здатний розпізнавати дії користувача та реагувати на них в режимі реального часу;
- спеціальне обладнання, поєднане з комп'ютером, яке забезпечує ефект занурення в процесі дослідження середовища.

2) VR без ефекту занурення.

Не кожному і не завжди потрібне буде повне занурення в альтернативну реальність. До типу «без занурення» відносяться симуляції з якісним зображенням, звуком і контролерами, що в ідеальному варіанті транслюються на широкоформатний екран. Також в цю категорію потрапляють такі проекти, як археологічні 3D-реконструкції старовинних поселень або моделі будівель, які архітектори створюють для демонстрації своєї роботи клієнту. Усі перераховані вище приклади не відповідають стандартам VR в повній мірі, але дозволяють відчувати модельований світ на кілька рівнів глибше, ніж інші засоби мультимедіа, а тому зараховуються до віртуальної реальності.

3) VR зі спільною інфраструктурою.

Сюди можна віднести «віртуальні світи» на кшталт Second Life і Minecraft. Єдина властивість, з перерахованих вище, якої їм не вистачає для повного комплекту — створення ефекту присутності: такі світи не

забезпечують повного занурення. Проте, у віртуальних світах добре прописано взаємодію з іншими користувачами, чого часто не вистачає продуктам так званої «справжньої» віртуальної реальності.

Віртуальні світи використовуються не лише в ігровій індустрії: завдяки таким платформам, як 3D Immersive Collaboration і Open Cobalt можна організовувати робочі та навчальні 3D-простори — це називається «спільна робота з ефектом присутності».

Створення можливості одночасної взаємодії в співтоваристві та повного занурення зараз є одним із важливих напрямків розвитку VR.

#### 4) VR на базі інтернет-технологій.

Фахівці в області комп'ютерних наук розробили спосіб створення віртуальних світів в Інтернеті, використовуючи технологію Virtual Reality Markup Language, аналогічну HTML. Вона на якийсь час була обділена увагою і зараз вже вважається застарілою, але з огляду на зростаючий інтерес Facebook до VR, в майбутньому віртуальна реальність обіцяє ґрунтуватися не лише на взаємодії, але й на інтернет-технологіях. [3]

### 1.2.3 Типи устаткування для VR-відображення

#### 1) Шоломи й окуляри / Head Mounted Display, HMD



Рисунок 1.4 — Зовнішній вигляд VR-окулярів

Такі пристрої складаються з двох невеликих екранів, розташованих навпроти кожного ока, шор, що запобігають потраплянню зовнішнього світла, і стереонавушників. Екрани показують злегка зміщені один відносно одного стереоскопічні зображення, забезпечуючи реалістичне 3D-сприйняття. У шоломах також містяться вбудовані акселерометри та датчики положення. Здебільшого просунуті VR-шоломи досить громіздкі, але останнім часом з'явилася тенденція до створення спрощених легковажних варіантів (в тому числі картонних, як на рисунку 1.4), які зазвичай призначені для смартфонів з VR-додатками.

Шоломи для віртуальної реальності діляться на три типи:

- для комп'ютера — працюють у зв'язці з ПК або консолями: Oculus Rift, HTC Vive, Playstation VR.

- для мобільних пристроїв — називаються гарнітурами та працюють у зв'язці зі смартфонами, виглядають як тримач з лінзами: Google Cardboard, Samsung Gear VR, YesVR.

2) Незалежні окуляри віртуальної реальності — самостійні пристрої, що працюють під управлінням спеціальних або пристосованих ОС: Sulong Q, DeePoon, AuraVisor.

3) Кімнати / Cave Automatic Virtual Environment, CAVE



Рисунок 1.5 — Зовнішній вигляд кімнат віртуальної реальності

Альтернатива для тих, хто не хоче зіпсувати зачіску — зображення в даному випадку транслюються не в шолом, а на стіни приміщення, часто представляють собою дисплеї MotionParallax3D (хоча для більш повного UX в деяких таких кімнатах потрібно надягати 3D-окуляри або навіть комбінувати CAVE і HMD). Існує думка, що VR-кімнати набагато кращі VR-шоломів: більш високий дозвіл, немає необхідності тягати на собі громіздкий пристрій, в якому деяких навіть заколисує, і самоідентифікація відбувається простіше завдяки тому, що користувач має можливість постійно бачити себе. Проте, придбання такої кімнати, ясна річ, вийде набагато дорожче, ніж купівля шолома.[21]

#### 4) Інформаційні рукавички / Datagloves



Рисунок 1.6 — Зовнішній вигляд VR-рукавичок

Для задоволення інстинктивної потреби користувача помацати руками те, що він знаходить для себе цікавим у процесі вивчення середовища, було створено рукавички з сенсорами для захоплення рухів кистей і пальців рук. Технічне забезпечення такого процесу варіюється — можливе використання оптоволоконних кабелів, тензометричних або п'єзоелектричних датчиків, а також електромеханічних пристроїв (таких як потенціометри).

#### 5) Джойстики (геймпади) / Wands



Рисунок 1.7 — Зовнішній вигляд VR-геймпаду

Спеціальні технічні засоби для взаємодії з віртуальним середовищем, які включають в себе датчики положення та рухів, а також кнопки та колесо прокрутки, як у миші. У наш час вони все частіше стають бездротовими, щоб уникнути незручностей і нагромаджень даних при приєднанні до комп'ютера. [6]

#### **1.2.4 Области застосування VR-технологій**

**Навчання.** Віртуальна реальність використовується для імітації навчального та тренувального середовища у заняттях, що потребують попередньої підготовки (до прикладу, керування літаком та навіть операції).

**Наука.** VR дає можливість удосконалити та прискорити вивчення молекулярного й атомного світу: занурившись у віртуальне середовище, вчений зможе використовувати частинки так, ніби вони є частинками пазлу.

**Медицина.** Окрім допомоги у підготовці лікарів-хірургів, технологія також корисна під час самих операцій, оскільки використовуючи спеціальне обладнання, можна контролювати рухи робота, одночасно маючи змогу краще контролювати процес.



Промисловий дизайн та архітектура. Щоб не будувати дорогі моделі техніки можна створити віртуальну модель, яка дозволить не тільки вивчити проект зсередини, але й перевірити його технічні характеристики.

Ігри та розваги. Наразі це найвідоміший і найширший варіант використання VR-додатків: він включає ігри, а також фільми, віртуальний туризм та різні події. Як вже було зазначено, VR інтегрується у різні сфери життя, і з міфу про наукову фантастику він перетворився на віртуальну, але, все ж, реальність.[22]

### **1.3 Інтерфейс взаємодії у VR**

Основою інтерфейсу сучасних застосувань віртуальної реальності є саме графічний контент, що змінюється у відповідь на зміну положення голови користувача. У деяких випадках даний інтерфейс доповнюється звуковою інформацією. Наразі немає стандартних вбудованих широкорозповсюджених технологій для створення тактильних відчуттів, запахів тощо, хоча в даному напрямку проводяться розробки. Але найголовнішим напрямком, в якому здійснюється робота над VR-інтерфейсом, це створення так званого «room-scale VR». Дана технологія полягає в реакції середовища віртуальної реальності не лише на положення голови, а й на переміщення тіла користувача у просторі [4]. Цей підхід реалізується встановленням у приміщення, де буде здійснюватися робота з VR додатком, камер та/або інфрачервоних датчиків, які фіксують переміщення користувача, передають зображення обчислювальному вузлу, який визначає зміну положення тіла та передає результуючі дані додатку VR, що в сукупності забезпечує реакцію середовища VR в режимі реального часу. Основними недоліками даної технології є перш за все те, що для її роботи необхідно придбання та встановлення додаткової апаратури, яка на даний момент ще досить коштовна, а також виникає складність у використанні даної технології, оскільки вона вимагає певного вільного простору в декілька квадратних метрів, бо інакше буде користувач буде зіштовхуватися з об'єктами

реального світу. Саме через вищезгадані недоліки більшість сучасних VR-застосувань розроблено для стаціонарної парадигми VR, де фіксуються зміни лише положення голови. Взаємодія користувача з VR-середовищем відбувається за допомогою VR-контролерів, а для випадку, якщо їх немає, було прийнято стандартний інструмент — курсор, що знаходиться в центрі зображення й виконує функцію вибору об'єкту (натиснення кнопки миші). Цей курсор спрацьовує, якщо буде протягом деякого часу затриманий на об'єкті VR-середовища. [23]

Віртуальна реальність у веб з точки зору веб-технологій — це один зі способів взаємодії користувача з мультимедійною інформацією, а також іншими людьми в режимі реального часу з використанням мережевих технологій. Цей спосіб цілком незвичний для Internet, оскільки зазвичай користувачу доводиться працювати з плоским двовимірним графічним інтерфейсом, який на даному етапі досягнув високого рівня зручності та задоволення потреб. Лише в деяких випадках може використовуватися 3D-контент, але лише як демонстраційний матеріал для чітко визначених та обмежених цілей, наприклад, об'ємна демонстрація товару в інтернет-магазині. Віртуальна реальність же, на відміну від звичного способу керування, дає змогу повністю зануритися у тривимірний світ, який оточує користувача за допомогою 3D-моделей. Взаємодія у цьому випадку буде відрізнятися від стандартної для веб, оскільки тут немає таких пристроїв введення-виведення, як миша та клавіатура. На зміну їм приходять більш інтуїтивні засоби, такі як рух голови та керування об'єктами VR за допомогою спеціальних маніпуляторів. І шолом, і маніпулятори містять датчики руху, тому можуть ефективно реалізовувати інтуїтивно-зрозумілий інтерфейс та більш швидко та гнучку роботу. Варто також розрізнити віртуальну реальність та звичайні плоскі об'єкти інтерфейсу користувача, які приведені до псевдотривимірного вигляду [5]. Досить часто такими є вікна VR-браузерів, які фактично є панелями, і майже нічим не відрізняються від вікон десктоп-браузерів, за виключенням того, що вони тепер змінили своє положення. Наразі, це єдиний варіант

реалізації роботи з браузерами, і пов'язано це з тим, що, по-перше, він є звичним інтерфейсом для користувачів, які будуть переходити на нову платформу, а по-друге, ще немає розробок щодо нової моделі інтерфейсу, який зміг би задовольнити потреби оптимальнішим чином [6]. На даний момент здійснюється спільна робота консорціуму W3C, Mozilla, Google, а також багатьох інших розробників та дизайнерів, які працюють в напрямку VR у web, над розробкою стандарту для такого інтерфейсу взаємодії. Таким чином, досить скоро користувачі зможуть відкрити для себе новий спосіб роботи в Інтернет.

#### 1.4 Аналіз аналогів

Як основний аналог до розроблюваного ресурсу вважаю доцільним виділити сервіс від Google — Google Arts & Culture. Для цього, було зведено в таблицю (табл. 2.1) основні критерії, за якими будуть порівнені дані ресурси:

Таблиця 2.1 — Порівняльна таблиця розроблюваного ресурсу з аналогом

№	Критерії	Мій web-сервіс	Google Arts & Culture
1	Інформаційність	-	+
2	Якість 3D відображення	+	+
3	Швидкодія	+	-
4	Продуктивність	+	-
5	Підтримка	+	+
6	Зручність	+	+
7	Простота до використання	+	-
8	Геолокація	+	+

1) Інформаційність — оскільки розроблюваний ресурс є вузькоспеціалізованим, тому виникає певна проблема з інформаційністю. Google Art & Culture — це онлайн-платформа, використовуючи яку,

громадськість може отримати доступ до зображень високої чіткості та художніх робіт, які зберігаються у музеях — партнерах ініціативи. Проект було запущено компанією Google у партнерстві з 17-ма міжнародно-відомими музеями, включно з Галереєю Тейт, Лондон; Музеєм мистецтва Метрополітен, Нью-Йорк та Галереєю Уффіці, Флоренція.

Платформа дає змогу користувачам здійснювати віртуальні тури галереями музеїв, знайомитись з фізичною та контекстуальною інформацією про художні роботи та створювати свої власні віртуальні колекції. Функція «walk-through» проекту використовує технологію Google Street View.

2) Google A&C для 3D відображення використовує власний сервіс Poly. В нього є API, яке може використовувати кожен розробник. Для розроблюваного додатку, мною було використано власні методи для 3D відображення на веб-ресурсі, оскільки у Poly є ряд суттєвих недоліків: швидкодія та продуктивність не є кращими сторонами даного сервісу. Під час відкриття відрендереного зображення починається велике навантаження на CPU та швидко заповнюється оперативна пам'ять. Внаслідок цього з'являється певна проблема з пристроями, на яких дані моделі зможуть запуснитись. На розроблювальному ресурсі ситуація повністю протилежна: оскільки моделі перетворюються в json формат, швидкодія та продуктивність стає значно вищою. Це дозволяє запускати створений веб-ресурс з різних десктопних та мобільних пристроїв. [24]

3) Підтримка у двох додатків достатньо розвинена. Google Arts & Culture — це відкритий проект. До нього залучаються цілі музеї та люди зі всього світу. До створеного мною сервісу можна отримати підтримку у міста, оскільки на даний воно значно розвивається у культурному плані і подібні ідеї вітаються, як ніколи раніше. Для цього потрібно лише кілька 3D-дизайнерів, системного та web-спеціалістів. Цим проектом можна підняти історичну популярність міста.

4) Через велике різноманіття ідей відображення картин та моделей є певне інформаційне перевантаження. Це спричинило проблему у простоті

використання сервісу, водночас як на розробленому додатку такої проблеми немає.

5) В якості візуалізації карти для Google Arts & Culture, Google використовує власний сервіс — Google Maps. Я використовую карту з відкритим доступом. У якості методу визначення навігації використовую Instagram, HTML5 - navigator.

Звичайно, змагатись із сервісом від Google важко, але у створеного мною додатку є переваги по ряду критеріїв, таких як: простота та зручність у використанні, продуктивність, швидкодія. У решті розглянутих критеріїв чієїсь явної переваги немає. Проблему з інформативністю, буде однозначно вирішено у наступних, удосконалених та випущених мною, версіях додатку.

## **1.5 Висновки до розділу 1**

Необхідність у дослідженні та розвитку даної технології є дуже великою. Так, у сучасному світі Інтернет є засобом обміну будь-якою мультимедійною інформацією, а також інформацією, інтерактивною по відношенню до користувача. Веб вміщує у собі всі сучасні наявні різновиди мультимедійних даних, але не передбачає способів взаємодії з ними як з об'єктами реального світу. Віртуальна реальність вирішує цю проблему. Крім того, що вона надає можливість переглядати VR-медіа та взаємодіяти з об'єктами подібно до матеріальних. Таким чином, відбувається покращення як якості контенту, так і інтерфейсу користувача.

Технологія створення віртуальної реальності в мережі вже зараз має якісні напрацювання, які обіцяють стрімкий та перспективний розвиток. Технологія продовжує стрімко розвиватися та набирати популярність, незважаючи на велику кількість складнощів. Маючи необхідні передумови та потужності для реалізації розробок та рішень, віртуальна реальність у веб в найближчому майбутньому завоює великий сегмент ринку та стане однією з найбільших індустрій в ІТ в цілому та серед Інтернет-технологій зокрема.

Отож, з огляду на всі вищезазначені аргументи, можна зробити висновок, що дослідження технології створення віртуальної реальності у веб є дуже перспективним напрямком, який однозначно варто досліджувати.

## **2 РОЗРОБКА МЕТОДУ ВІДОБРАЖЕННЯ 3D МОДЕЛЕЙ У ВЕБ-ДОДАТКУ НА ОСНОВІ ОТРИМАНОЇ ІЗ СОЦМЕРЕЖІ ГЕОЛОКАЦІЇ**

Запропонована система поєднуватиме у собі інструмент для:

- 1) демонстрації доповненої реальності та аналізу даних;
- 2) для персоналізації контенту;
- 3) для демонстрації.

Даний вид продукту є продуктом, що набирає популярності на ринку, і тим самим пропонує вакантні персональні місця на ринку послуг та технологій, разом із доступністю та гнучкістю веб-додатку, оскільки саме веб-додаток може гарантувати найлегший та найшвидший доступ до розроблюваного ресурсу. Це помітно виділяє цей тип ресурсу з-поміж мобільних чи десктопних додатків. [28]

Веб-ресурс не потребує додаткових встановлень, чи прив'язки до платформ чи сервісів. Це може допомогти в охопленні максимальної аудиторії користувачів. Доступ до ресурсу скорочується до наявності пристрою з доступом до інтернет та мобільного браузеру, а конвертування адреси веб-сайту, скажімо у QR код, робить доступ до ресурсу окрім легкого, ще й швидким. Дана тематика роботи та й сам додаток може виступати в подальшому у вигляді платформи, яку можуть використовувати туристичні центри міст чи приватні туристичні компанії.

### **2.1 Підходи до застосування технології віртуальної реальності у веб-додатках**

На теперішньому етапі розвитку технологій віртуальної реальності для веб-застосунків умовно виокремлюють методи до розробки додатків. Це розділення засноване на контенті віртуальної реальності, що було використано. Відповідно виділяється перша група програм, які використовують підготований

VR-вміст, та другу — що генерує інтерактивний 360°-вміст у реальному часі роботи додатку, використовуючи потужність інструментів розробника.

Це розділення досить умовне, оскільки для першого випадку, коли веб-додаток використовує готовий вміст VR та показує його користувачеві, для виконання цього завдання необхідно використовувати спеціальний інтерфейс, який взаємодіє з обладнанням і дає змогу правильно відображати вміст на пристрої віртуальної реальності. [8]

Але тоді як другий підхід, що характеризується створенням середовища VR під час застосування, досить відрізняється від попереднього, оскільки передбачає використання допоміжних спеціалізованих інструментів для цього завдання і, відповідно, забезпечує повністю іншу архітектуру додатків.

В рамках магістерської роботи необхідно проаналізувати наявні методи та технології моніторингу геоданих користувачів з їх подальшим використанням, а також можливість реалізації веб-ресурсу, який зможе, безпосередньо у браузері, за допомогою карти відображати певні 3D об'єкти, з додатковою інформацією про них. У нашому випадку — це історичні реконструкції об'єктів для VR.

Також потрібно подбати про масштабованість системи у рамках підходу до організації бази даних, а також про кросбраузерність та максимальну швидкодію при передачі даних, адже додаток є саме веб-сервісом. Для вирішення поставленої задачі потрібно ознайомитись та вивчити наукову літературу з технологій відтворення 3D-моделей. Розглянути підходи до вирішення задачі та проаналізувати можливі методи вирішення.

Для реалізації поставленої задачі необхідно забезпечити оптимальну швидкодію при розподіленні на карті усіх доступних об'єктів з урахуванням геоданих користувача та побудувати 3D об'єкти поєднуючи їх з картою. Попередньо дані потрібно отримати від користувача з профайлу що авторизувався у соціальній мережі. Вибрати тип пристрою, за допомогою якого буде відбуватись перегляд 3D моделей. Налаштувати роботу з ним. [29]



Принцип роботи алгоритму взаємодії користувача з ресурсом зображено на рисунку 2.1.

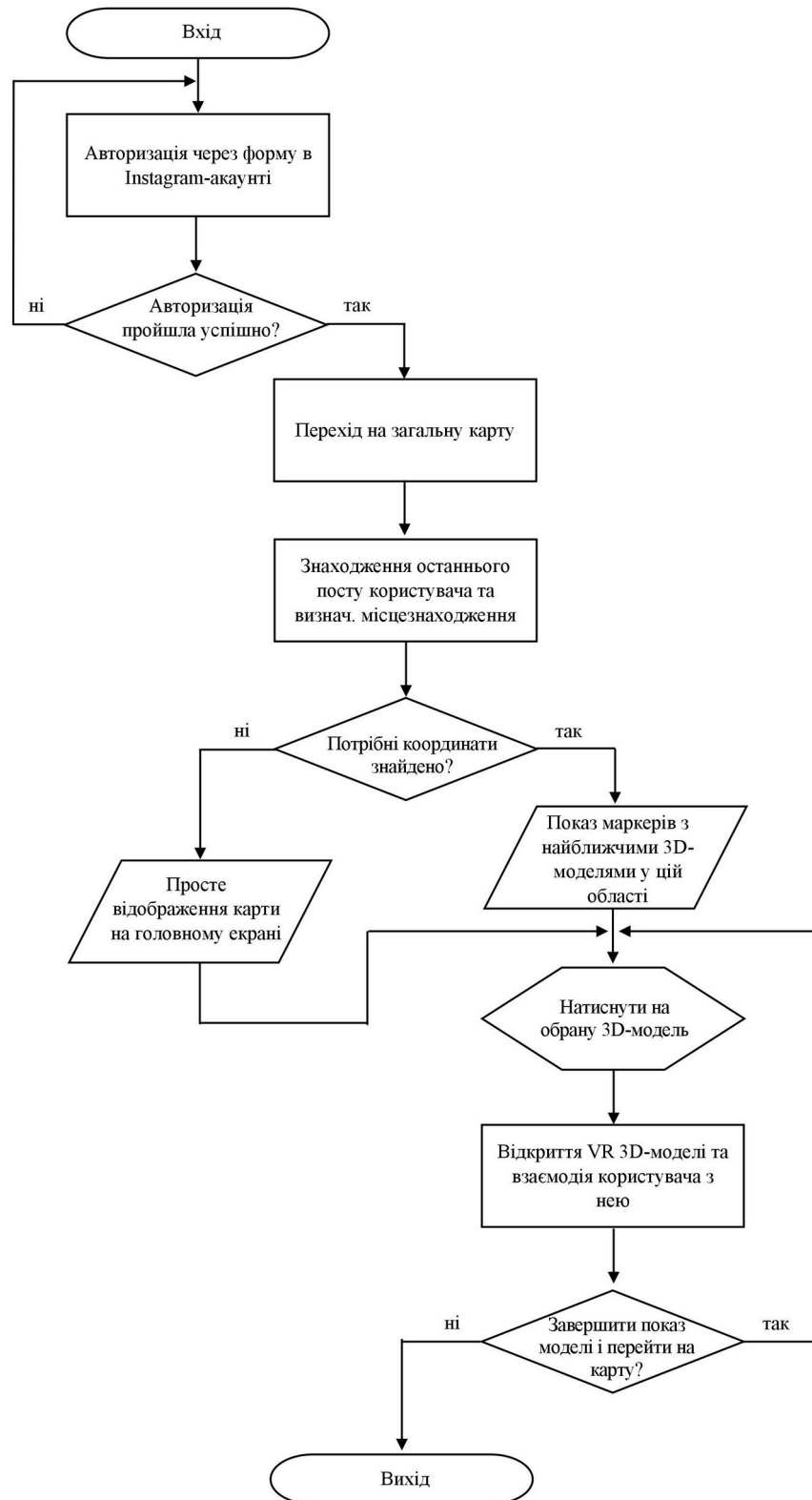


Рисунок 2.1 — Алгоритм взаємодії користувача з ресурсом

В рамках тестування програми перевірити правильність відображення об'єктів на карті за допомогою окулярів або шолома віртуальної реальності.

## 2.2 Розробка методу моніторингу та отримання даних про геолокацію

Геолокація дозволяє визначати географічне розташування відвідувачів веб-сайту. І це не означає просто країну або навіть місто, в якому знаходиться відвідувач, геолокація може звузити інформацію про місцезнаходження людини до міського кварталу або навіть визначити її точні координати, якщо вона зайшла на сторінку зі смартфона.

Таблиця 2.1 — Мінімальні системні вимоги

Браузер	IE	Firefox	Chrome	Safari	Opera	Safari IOS	Android
Мінімальна версія	9	3.5	5	5	10.6	3.2	2.1

Більшість нових можливостей JavaScript були частиною первісної специфікації HTML5, а потім були відокремлені, коли специфікацію передали у відання організації W3C. Але геолокація ніколи не була частиною HTML5, а просто повністю сформувалася приблизно в той же самий час.

Принцип роботи геолокації. Веб-сторінка хоче отримати дані про місцезнаходження користувача. Відвідувач може дозволити надавати ці дані або ж не надавати їх. Така поведінка браузера Chrome не є власною ініціативою його розробників, а офіційним правилом стандарту геолокації, які вимагають, щоб кожен веб-сайт, який намагається отримати дані про місцезнаходження користувача, отримав для цього його дозвіл.

Щоб обчислити місцерозташування користувача, браузер заручається допомогою постачальника послуг позиціонування (location provider), наприклад, для Chrome це сервіс Google Location Services. Позиціонування є завданням не із легких, і постачальник застосовує кілька різних підходів для її вирішення. [30]

У випадку настільного комп'ютера з фіксованим (тобто не безпроводним) підключенням, метод простий, але дає не дуже точних результатів. Коли хтось підключається до інтернету, дані з його комп'ютера або локальної мережі направляються (через кабель, виділену лінію або комутоване телефоном з'єднання) на потужний мережевий апаратний пристрій, який, у свою чергу, направляє їх у мережу Інтернет. Цей апаратний пристрій має однозначний IP-адрес, тобто числовий код, який ідентифікує його для інших учасників мережі. Цей пристрій також має фізичну поштову адресу. Постачальник використовує ці два типи інформації для визначення географічного розташування відвідувача веб-сторінки. Спочатку обчислюється IP-адреса пристрою, через яку здійснюється підключення, а потім визначається його фізична адреса. Відповідно, такий непрямий підхід дозволяє визначити точне місце розташування не користувача, а тільки його інтернет-провайдера.

Позиціонування за допомогою IP-адреси є неточним способом геолокації. Якщо є найкраще джерело даних, постачальник розташування буде використовувати це джерело. Проте часто навіть такі неточні результати є корисними.

Якщо відвідувач зайшов на сторінку з нетбука або смартфона по бездротовому підключенню, постачальник розташування використовує найближчі точки бездротового доступу. В ідеальному випадку постачальник розташування вибирає дані з величезної бази даних, щоб визначити точне місцезнаходження цих точок доступу, а потім використовує отриману інформацію, щоб визначити місцерозташування користувача методом тріангуляції.

А в разі підключення з мобільного телефону застосовується подібний метод тріангуляції, але за допомогою сигналів від різних антен мобільного зв'язку. Ця швидка, порівняно ефективна процедура зазвичай дозволяє визначити місце розташування відвідувача з точністю до 1 км.

Нарешті, багато мобільних пристроїв оснащено спеціальними апаратними засобами GPS (Global Positioning Service — глобальна система навігації та

визначення положення), що дозволяє визначати місцезрештування таких пристроїв із точністю всього лише до кількох метрів. Але цей метод геолокації має свій недолік — він повільніший і споживає більше енергії, що важливо для пристроїв, які працюють на акумуляторах. Крім цього, він не особливо добре працює в містах з великими та високими будинками через відбиття сигналу від будівель.

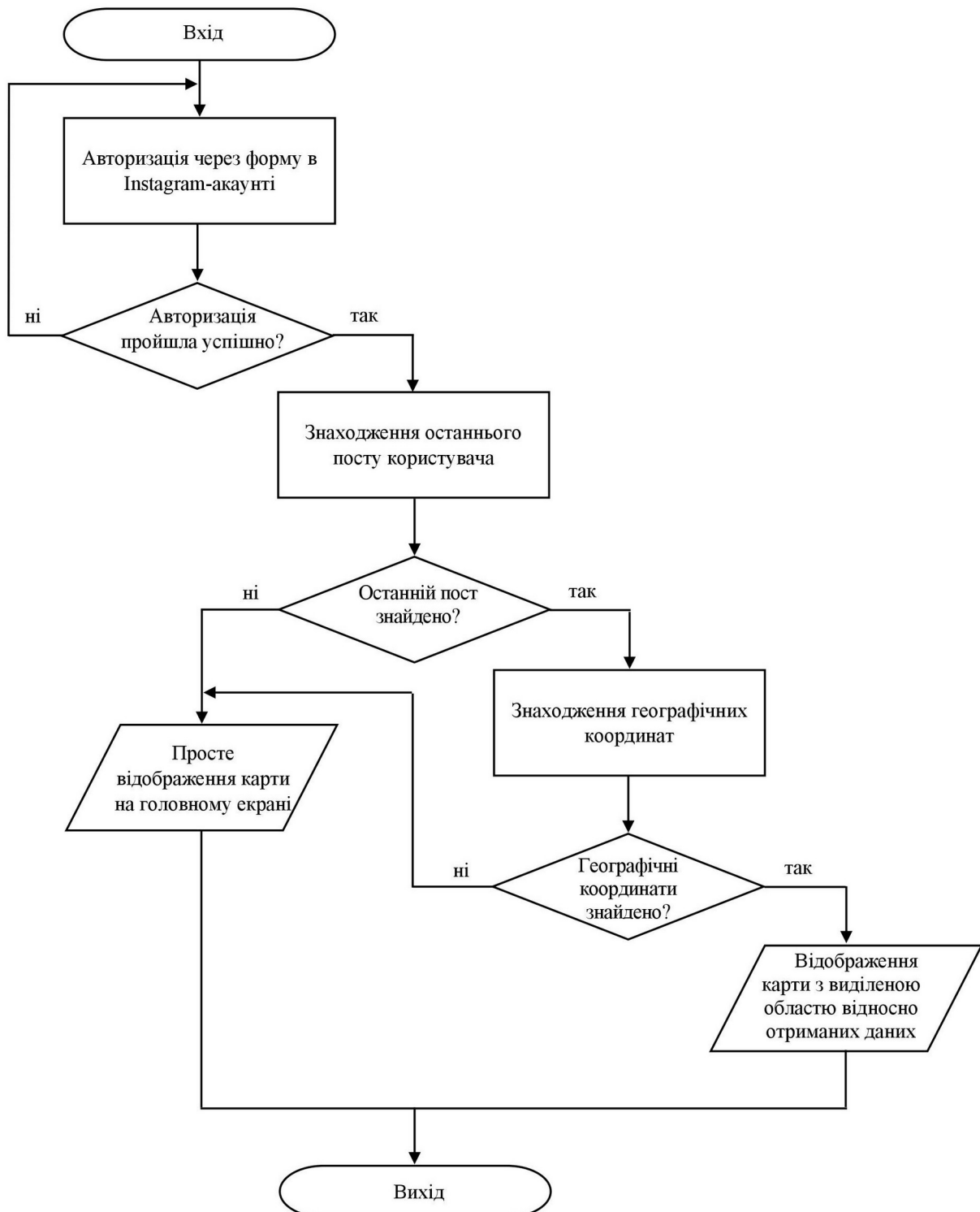


Рисунок 2.2 — Алгоритм отримання даних про останню відзначену геолокацію

Звичайно ж, можливі й інші способи визначення місцезнаходження. Ніщо не перешкоджає використовувати для цього інші джерела інформації, такі як дані від RFID-пристроїв (Radio Frequency Identification — радіочастотна ідентифікація), дані від пристроїв Bluetooth, файли cookies з інформацією від картографічного сайту на зразок Google Maps і т.п. [9]

З усього цього можна зробити наступний висновок: незалежно від способу підключення до інтернету, навіть якщо відвідувач веб-сайту сидить за настільним комп'ютером, можливість геолокації дозволяє визначити його місцезнаходження з більшою або меншою точністю. А якщо він виходить в інтернет з мобільного телефону або з пристрою, оснащеного апаратним забезпеченням GPS, його місце розташування може бути визначено з точністю швидше більшою, ніж меншою.

### **2.3 Розробка методики конвертації 3D моделей**

JSON або JavaScript Object Notation — це мінімальний, читабельний формат для структуризації даних. Він використовується головним чином для передачі даних між сервером та веб-додатком, як альтернатива XML.

Дві основні частини, що складають JSON — це ключі та значення. Разом вони складають пару:

- 1) Ключ: це завжди рядок, укладений у лапки.
- 2) Значення: може бути рядком, числом, булевим виразом, масивом або об'єктом.
- 3) Пара ключ/значення: пара ключових значень слідує певному синтаксису, а за ключем слідує двокрапка та наступне значення. Пари розділяються комами.

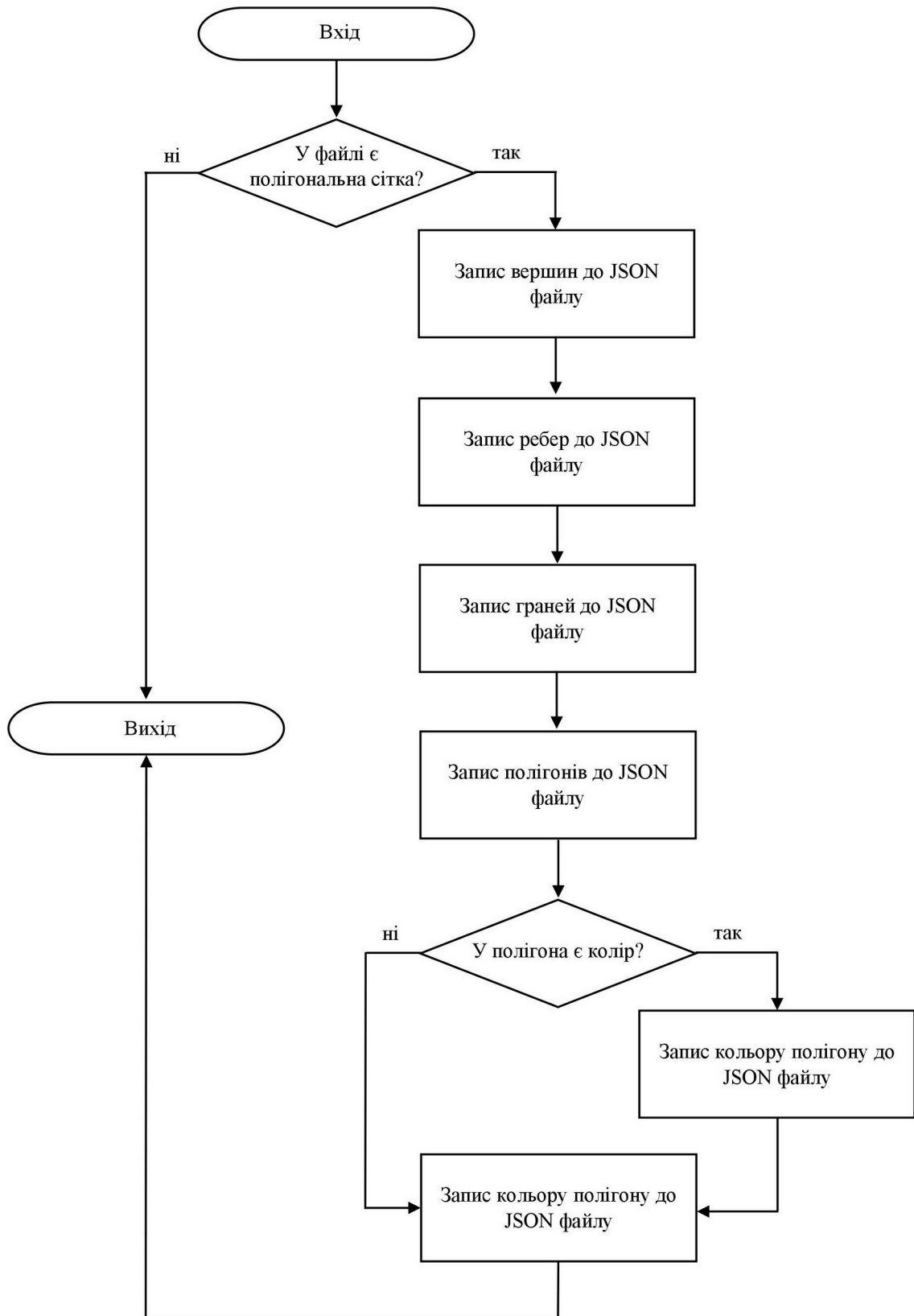


Рисунок 2.3 — Алгоритм конвертації створених «важких» 3D моделей будівель

Для прикладу, нижче один рядок із зразка JSON:

«foo» : «bar»

Цей приклад — пара ключ/значення. Ключ — «foo», а значення — «bar».

Типи даних:

- масив — асоціативний масив значень;
- булеві значення — вірно чи хибно;
- число — ціле число;
- об'єкт — асоціативний масив пар ключ/значення;
- рядок — кілька простих текстових символів, які зазвичай утворюють слово;

- масив — позначається квадратними дужками, усе, що знаходиться всередині квадратних дужок, є частиною масиву.

```
"foo" : {
  "bar" : "Hello",
  "baz" : [ "quuz", "norf" ]
}
```

Об'єкт позначається фігурними дужками. Все, що знаходиться всередині фігурних дужок, є частиною об'єкта.

«foo» і відповідний об'єкт — пара ключ/значення.

```
"foo" : {
  "bar" : "Hello"
}
```

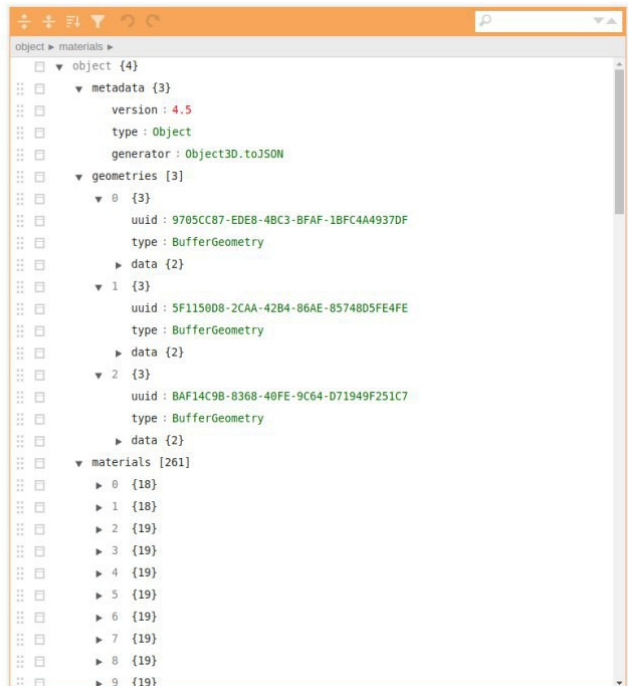
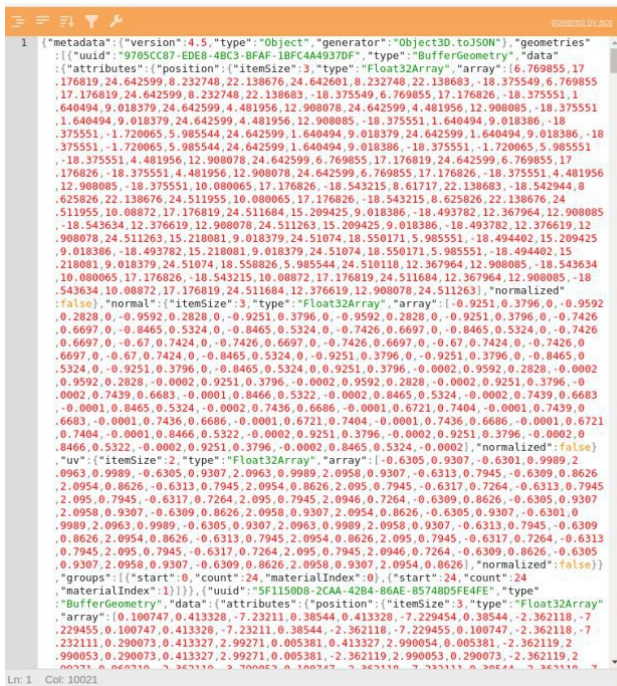


Рисунок 2.4 — Приклад сконвертованої 3D-моделі у JSON

Формат обміну даними — це текстовий формат, який використовується для обміну даними між різними платформами та операційними системами. JSON — найпопулярніший і найлегший формат обміну інформацією для веб-додатків.

У JavaScript можна легко проаналізувати дані JSON, отримані від веб-сервера, використовуючи `JSON.parse()` метод. Цей метод аналізує рядок JSON і будує значення JavaScript або об'єкт, описаний рядком. Якщо даний рядок недійсний JSON, то буде отримано синтаксичну помилку.

Припустимо, отримано від веб-сервера таку послідовність, кодовану JSON:

```
{"name": "Peter", "age": 22, "country": "United States"}
```

Тепер можна просто використовувати `JSON.parse()` метод JavaScript, щоб перетворити цей рядок JSON в об'єкт JavaScript і отримати доступ до окремих значень за допомогою крапки нотації (`.`), наприклад:

```
// Store JSON data in a JS variable
var json = '{"name": "Peter", "age": 22, "country": "United States"}';

// Converting JSON-encoded string to JS object
var obj = JSON.parse(json);

// Accessing individual value from JS object
alert(obj.name); // Outputs: Peter
alert(obj.age); // Outputs: 22
alert(obj.country); // Outputs: United States
```

Об'єкти та масиви JSON також можуть бути вкладені. Об'єкт JSON може довільно містити інші об'єкти JSON, масиви, вкладені масиви, масиви об'єктів JSON тощо. Наступний приклад покаже, як розібрати вкладений об'єкт JSON і витягнути всі значення в JavaScript:

```
/* Storing multi-line JSON string in a JS variable
using the new ES6 template literals */
var json = `
  "book": {
    "name": "Harry Potter and the Goblet of Fire",
    "author": "J. K. Rowling",
    "year": 2000,
```



```

    "characters": ["Harry Potter", "Hermione Granger", "Ron Weasley"],
    "genre": "Fantasy Fiction",
    "price": {
      "paperback": "$10.40", "hardcover": "$20.32", "kindle": "$4.11"
    }
  }
};

```

```

// Converting JSON object to JS object
var obj = JSON.parse(json);

// Define recursive function to print nested values
function printValues(obj) {
  for(var k in obj) {
    if(obj[k] instanceof Object) {
      printValues(obj[k]);
    } else {
      document.write(obj[k] + "<br>");
    }
  }
};

```

```

// Printing all the values from the resulting object
printValues(obj);

```

```

document.write("<hr>");

```

```

// Printing a single value
document.write(obj["book"]["author"] + "<br>"); // Prints: J. K. Rowling
document.write(obj["book"]["characters"][0] + "<br>"); // Prints: Harry Potter
document.write(obj["book"]["price"]["hardcover"]); // Prints: $20.32

```

Іноді об'єкт або значення JavaScript з коду потрібно перенести на сервер під час спілкування з Аґах. Для цього JavaScript пропонує метод `JSON.stringify()`, який перетворює значення JavaScript у рядок JSON, як показано нижче:

```

// Sample JS object
var obj = {"name": "Peter", "age": 22, "country": "United States"};

// Converting JS object to JSON string
var json = JSON.stringify(obj);
alert(json);

```

Вихід із наведеного прикладу буде виглядати приблизно так:

```

{"name": "Peter", "вік": 22, "country": "Сполучені Штати"}

```

Хоча об'єкти JavaScript і JSON виглядають досить схоже, але вони не зовсім однакові. Наприклад, в JavaScript імена властивостей об'єкта можуть

бути укладені в одиначні лапки ('...') або подвійні лапки ("..."), або можна взагалі опустити їх. Але в JSON усі назви властивостей мають бути укладені у подвійні лапки.

## 2.4 Проектування моделей веб-ресурсу

Розвиток та вдосконалення сучасних систем у різних сферах людської діяльності неподільно пов'язане з розвитком технологій моделювання процесів функціонування складних систем. Якість моделей, що використовуються на етапі проектування, залежить насамперед від якості створеної системи. [31] Оптимізація параметрів керування, визначення ефективності роботи системи за заданими параметрами й інші завдання вирішуються методами аналітичного та імітаційного моделювання. Імітаційне моделювання, на відміну від аналітичних моделей, дозволяє відтворити процес функціонування системи та детальніше керувати ним.

### 2.4.1 Концептуальна модель

Кожен веб-додаток обов'язково повинен мати клієнтську частину, де буде реалізовано користувацький інтерфейс, значну частину функціоналу та бізнес-логіку (рис 2.1).

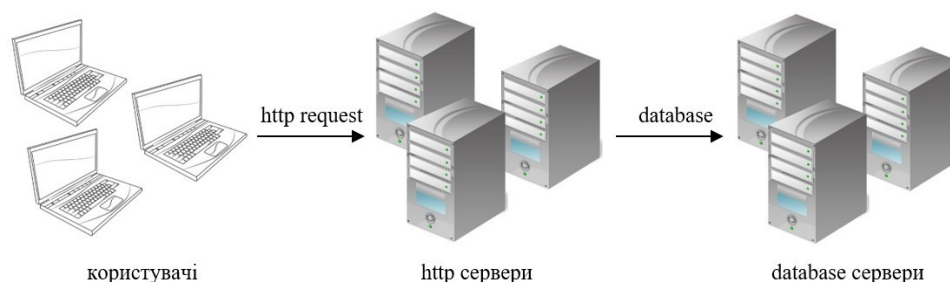


Рисунок 2.5 — Схематичне зображення зв'язку клієнт-сервер

Для розроблюваного веб-ресурсу до клієнтської частини входить ще й адмінанель для менеджменту даних та безпосереднього розпаралелення адмінського та користувацького функціоналу.

Такий підхід гарантує:

- скорочення завантаженого користувачем коду;
- робить більш гнучкі та незалежні один від одного інтерфейси адмінпанелі й клієнта;
- не створює надлишкових залежностей коду чи ресурсів.

Серверна частина даного веб-додатку виконуватиме функції передачі даних за запитами від клієнта на основі даних, отриманих здійсненням запитів до сервера мережі Instagram з ключем авторизації, задля отримання даних з профайлу та подальшого «спілкування» з базою даних.

Основною вимогою до 3D об'єктів має бути те, що вони повинні конвертуватись у такий формат даних, який було б можливо швидко передавати по мережі до клієнта. У тому числі, необхідно подбати про їхнє зберігання у базі даних задля уможливлення пошуку між об'єктами та маніпуляцією з ними (до прикладу, як видалення чи додавання нових).

Під час проектування бази даних, а саме її вибору, потрібно підлаштуватись під той формат даних, який буде прийнятним для зберігання 3D моделей, а також матиме можливість каталогізувати дані.

#### 2.4.2 Логічна модель

При побудові архітектури веб-додатку необхідно максимально зменшити залежність між структурними одиницями (рис. 2.2).

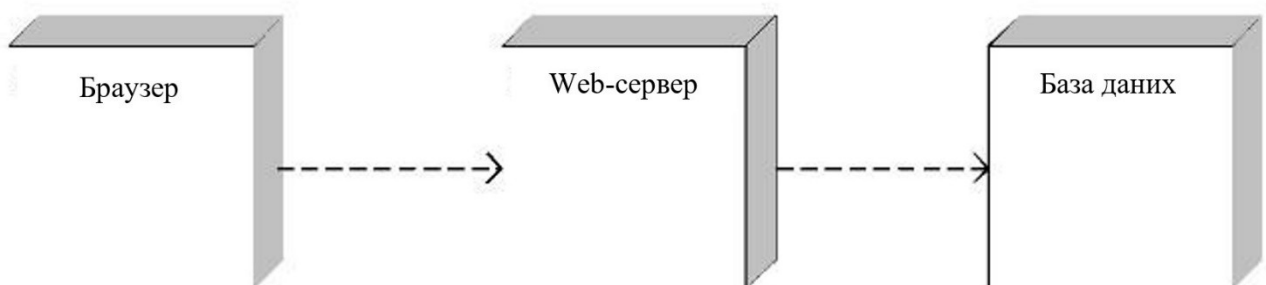


Рисунок 2.6 — Вигляд залежності між структурними одиницями додатку

У загальному випадку додаток складається з трьох структурних одиниць:

- модуль, що працює під управлінням браузера;
- модуль, що працює під управлінням web-сервера;
- база даних.

Для досягнення максимальної незалежності між структурними одиницями, необхідно, щоб кожна структурна одиниця оперувала лише необхідним їй набором даних.

Для ясності та однозначного розуміння потрібно уточнити визначення деяких основних термінів.

Браузер — це прикладне програмне забезпечення для перегляду web-сторінок.

HTML — це стандартна мова розмітки документів.

Web сервер — це програмне забезпечення, здатне приймати HTTP запити від клієнтів, обробляти їх та надсилати відповідь відповідно до стандарту протоколу.

База даних — це представлена в об'єктивній формі сукупність незалежних матеріалів, упорядкованих таким чином, щоб їх можна було знайти й обробити ЕОМ. [10]

Розроблювана у магістерській роботі програма написана на JavaScript і виконується «Браузером» шляхом «спілкування» з Web-сервером через AJAX, отримуючи відповіді у відповідності з певним протоколом.

«Браузер» інтерпретує лише HTML-код, а самі перетворення відбуваються вже за допомогою XSLT-перетворень на стороні Web-Сервера.

Опираючись на попередні архітектурні рішення потрібно виділити конкретні технології та платформи з якими варто працювати.

Клієнтська частину краще за все виконувати на скриптовій мові JavaScript, оскільки вона динамічна та об'єктно-орієнтована, а також використовувати реалізацію стандарту ECMAScript. Підхід REST буде використовуватися в комунікації між сервером і клієнтом. REST — це підхід до архітектури мережевого протоколу, що забезпечує доступ до інформаційних ресурсів.

Дані мають передаватися у невеликих типових форматах (наприклад, HTML, XML, JSON). Протокол REST (як і HTTP) повинен підтримувати функцію кешування, бути незалежним від мережевого рівня та зберігати інформацію про стан між парами запит-відповідь. Підтверджується, що такий підхід забезпечує масштабованість системи та дозволяє їй розвиватися з новими вимогами. [2]

На стороні сервера, Node.js слід використовувати як платформу, оскільки вона призначена для запуску високопродуктивних веб-додатків, написаних на JavaScript. Особливості платформи не обмежуються створенням серверних сценаріїв для веб, тому платформу можна також використовувати для створення стандартних клієнтських та серверних мережевих додатків. Двигун V8, розроблений Google, використовується для забезпечення виконання JavaScript.

### **2.4.3 Фізична модель**

Маючи уявлення про типи VR засобів і платформ, можна виділити конкретну цільову аудиторію пристроїв. Як опорну технологію доцільно обрати Google Cardboard. В чому переваги цієї платформи?

Google Cardboard — це доступна відповідь компанії на досвід віртуальної реальності (VR): простий переглядач, в який користувачі можуть вставити свої смартфони. Він підходить до екранів до шести дюймів, і компанія зазначає, що працює із більшістю окулярів. Google Cardboard — це пристрій, виготовлений із картону, резинки, магнітів та декількох лінз. [7]

Також є безліч комп'ютерних програм, спеціально розроблених для роботи з глядачем. За даними Google, «більшість додатків Cardboard працюють з Android 4.1+ та найновішими смартфонами iOS».

У підсумку виходить дешева з точки зору придбання або виробництва самотужки платформа, що не прив'язана до конкретної операційної системи.

Необхідно детальніше розібрати деякі етапи розробки моделі, задля отримання результату, зображеного на рисунку 2.3.



Рисунок 2.7 — Приклад зовнішнього вигляду готового 3D-об'єкта

Приклад розробки та експорту об'єкта розглядатиметься у 3D-редакторі Blender.

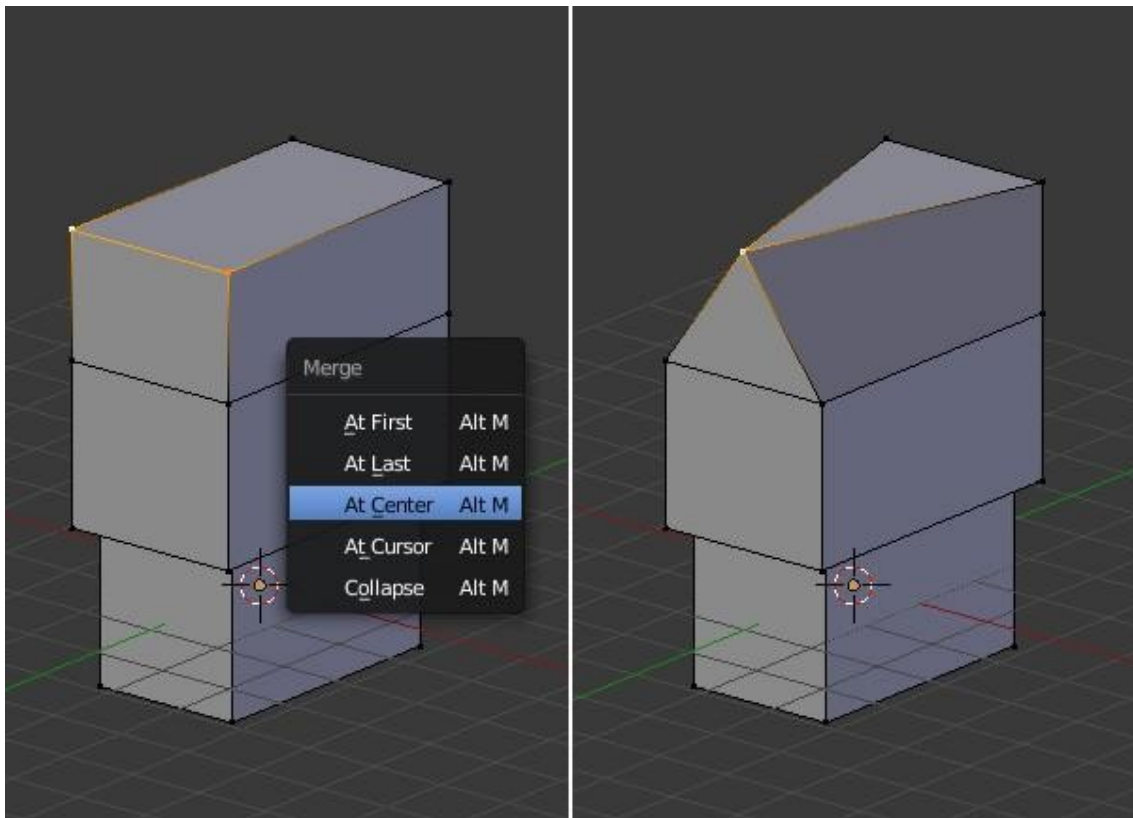


Рисунок 2.8 — Початок побудови 3D-моделі

Будування моделі ведеться з розрахунком як на її самостійний показ, так і показ на імпровізованій сцені. А отже, наявність додаткових елементів декору навколишнього рельєфу можна опустити.

Попередні дії виконуються задля зменшення розміру даних, що будуть передаватись і відмальовуватись на стороні клієнта. Із цією ж метою потрібно зберігати баланс у кількості полігонів і деталізації деяких елементів (рис. 2.5).

Після моделювання, текстуризування та виконання UV-розгортки необхідно підготувати модель до експорту (рис. 2.6).

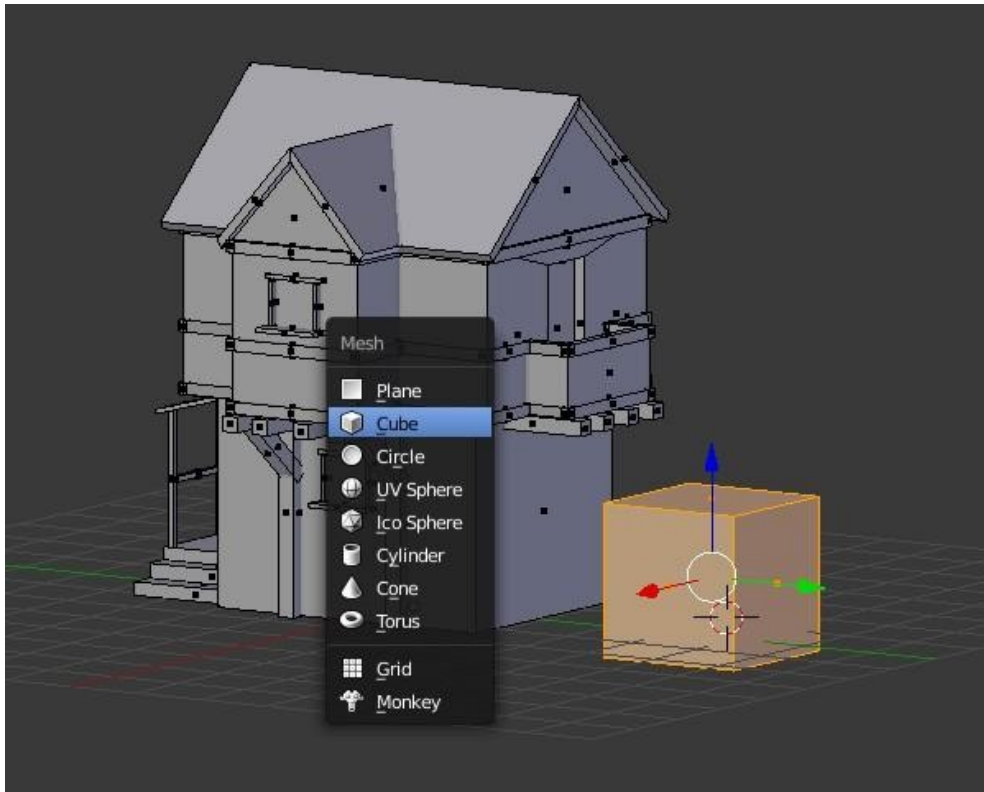


Рисунок 2.9 — Вигляд низькополігональної моделі

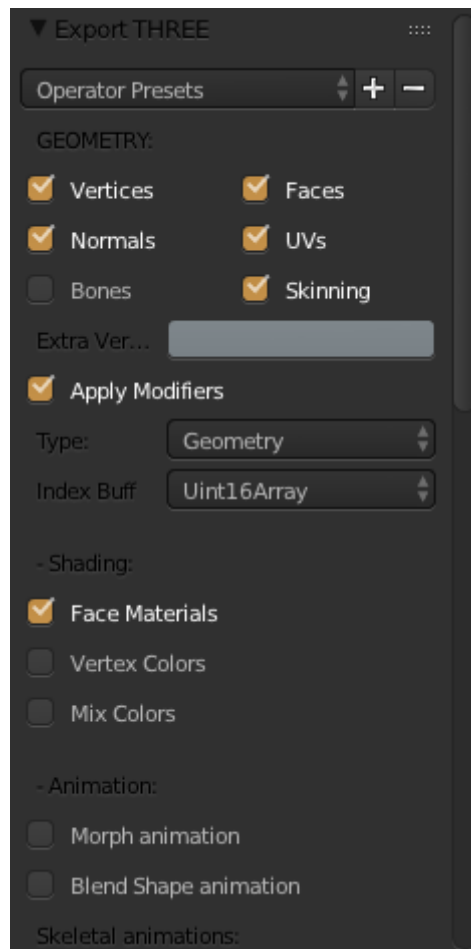


Рисунок 2.10 — Вигляд вікна налаштування експорту



При налаштуванні експорту потрібно вказати, що саме необхідно експортувати (вершини, ребра, матеріали, текстури, розгортки і т.п.).

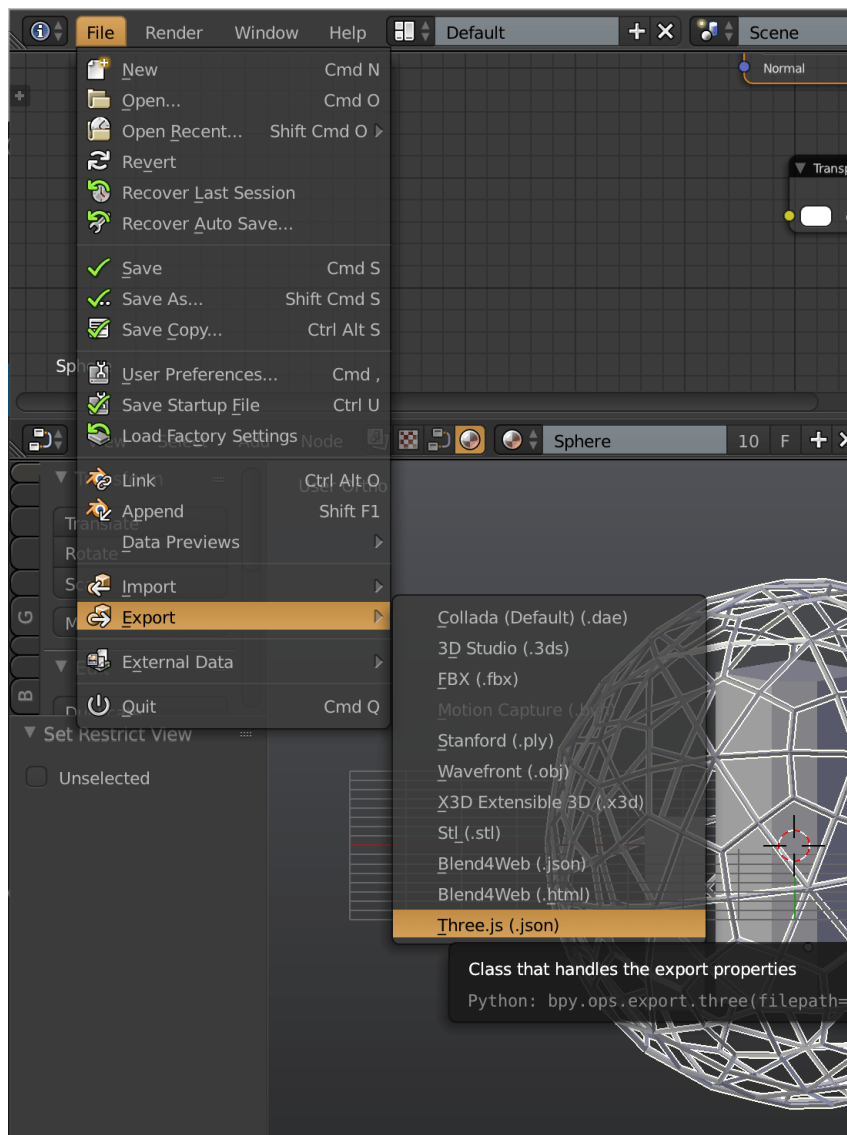


Рисунок 2.11 — Вигляд вікна експорту моделі



Рисунок 2.12 — Представлення успішно завантаженої та налаштованої сцени

У разі успішного експорту та передачі його в заздалегідь налаштовану сцену фреймворка, пристрій може бути готовий до перегляду моделі. Окрім відображення об'єктів, веб-ресурсу потрібна реалізація менеджменту, транспортування та збереження моделей, представлення їх у доступному вигляді на карті світу.

Для того, щоб віртуальна реальність працювала, потрібно зануритися в інший світ. А це означає блокувати майже все інше (звідси безліч відеозаписів людей, які забувають своє оточення, намагаючись битися з якимось вигаданим ворогом).

Google Cardboard — це конструкція VR, яка дозволяє користувачам блокувати зовнішні візуальні стимули та зосереджуватися на екрані свого пристрою. Після складання пристрою, в нього потрібно вставити мобільний пристрій, відкрити VR-додаток на вибір.[11]

Дві пластикові акрилові лінзи з картону роблять цей досвід більш правдоподібним, наслідуючи спосіб, яким очі обробляють реальний світ (а також спосіб встановлення самих VR-екранів), водночас як бічні клаптики блокують відволікання, щоб очі залишалися наведеними на те, що з'являється на екрані (рис. 2.9).



Рисунок 2.13 — Вигляд процесу використання Cardboard

Google Cardboard блокує периферійне бачення для більш захоплюючого досвіду. Єдиним недоліком (крім помітного у відсутності ремінця) є те, що Google Cardboard нічого не робить для блокування зовнішнього шуму, що може протидіяти повному зануренню у віртуальну реальність.

Потрібно не забувати, що варто робити часті перерви під час використання будь-якого пристрою перегляду VR, щоб уникнути головного болю та перенапруження очей.

Google Cardboard продається від 15 доларів (або 25 доларів для двох глядачів). Крім того, його можна побудувати його самостійно (наприклад, із коробки для піци); Google навіть надає інструкцію як робити окуляри у домашніх умовах (рис. 2.10), просто необхідно буде взяти кілька інших речей, наприклад, лінзи, магніти та резинку, і час, щоб все це скласти разом.



Рисунок 2.14 — Вигляд процесу складання окулярів VR

Складання пристрою Google Cardboard займає лише кілька кроків. Якщо шукати щось із більш привабливим виглядом (або виготовлене з більш міцного матеріалу, або із способом закріпити його на голові), можна придбати інші варіанти для роботи в тандемі з додатками Google Cardboard, у деякі з яких можна розмістити й більші екрани. Вони варіюються приблизно від 5 до 100 доларів.

## 2.5 Проектування БД

Розуміючи з якими типами даних потрібно буде працювати, чудовим вибором стане MongoDB, оскільки це документо-орієнтована система управління базами даних (СУБД) з відкритим сирцевим кодом, яка не потребує описування схем таблиць.

MongoDB має ряд важливих особливостей:

- MongoDB зберігає дані в гнучких, JSON-подібних документах, значення полів яких можуть варіюватися від документа до документа, а структура даних може змінюватися з часом;

- модель документа відображає об'єкти в коді програми, що спрощує роботу з даними;

- спеціальні запити, індексація й агрегація в режимі реального часу забезпечують потужні способи доступу та аналізу даних;

- MongoDB є розподіленою базою даних по своїй суті, тому висока доступність, горизонтальне масштабування та географічний розподіл вбудовані та прості у використанні.[13]

Під час розробки програмного продукту постає необхідність зберігати дані про користувачів та адміністраторів, об'єкти на карті з інформацією про них, а також окремого збереження вимагають 3D-моделі.

Модель документів зберігання даних (JSON/BSON) обрано для магістерської кваліфікаційної роботи, оскільки вона значно простіше кодується та керується (завдяки застосуванню стилю без схем), а внутрішня кластеризація відповідних даних забезпечує додаткове підвищення продуктивності.

### **3 ОПИС РЕАЛІЗАЦІЇ РІШЕННЯ ТА ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАСОБУ**

#### **3.1 Опис архітектури та даних програми, вимоги до ПЗ і АЗ**

Розроблювана програма, реалізована мовою JavaScript на базі платформи Node.js, містить у собі наступні модулі:

- модуль для створення та додавання даних про модель;
- модуль для завантаження моделі;
- модуль для редагування моделі;
- модуль для видалення моделі;
- модуль для відображення моделі засобами VR;
- модуль авторизації з Instagram-сервером;
- модуль збереження даних з останніх постів з Instagram;
- модуль виділення даних з постів Instagram та їх обробка.

На всіх архітектурних рівнях додатку, варто працювати з JSON та JavaScript об'єктами. Дані з постів користувача надсилатимуться у форматі .json. У схожому на JSON форматі зберігаються дані у базі даних MongoDB. Бібліотека Three.js сприймає моделі до рендеру у форматі JSON.

InstagramApi:

Wall posts — тип Array, список усіх постів на стіні;

Post — тип Object, інформація про пост;

Geo — тип Object, інформація про геодані поста;

Coordinates — тип String координати локації з поста.

Three.js:

Model — тип Object.

Open Street Map:

InitializeData — тип Object;

viewData — тип Array;

viewDataItem — тип Number.

Опис бази даних веб-сайту:

1) Collections:

- dataBuilds — колекція локацій;
- dataModels — колекція моделей;
- Users — колекція користувачів;
- Admins — колекція адміністраторів;
- Places — колекція згаданих місць за регіонами.

2) Items:

```
dataBuild {_id: ObjectId,
           nameObj: String,
           idObj:String,
           description: String,
           dataId: ObjectId,
           loc: Object}
```

```
dataModel {_id: ObjectId,
            name: String,
            data: Object.}
```

```
User {_id: ObjectId,
       name: String,
       userId; Number,
       places: Array.}
```

```
Admin {_id: ObjectId,
        pass: String,
        login; string,}
```

```
Place {_id: ObjectId,
        name; String;
        coordinates; Array,
        counter; Number,
        region:String.}
```

Мінімальні програмні та апаратні вимоги до програмного забезпечення:

- інтернет-браузер версії Chrome 49 або Chrome for Android 57;
- підключення до мережі інтернет;
- гіроскоп у мобільному пристрої (опціонально);
- Google Cardboard чи аналоги (опціонально).

Шляхів для еволюції додатку декілька:

- розширення функціоналу з моніторингу даних у мережі Instagram, шляхом аналізу фотографій та геоданих до них;
- моніторинг коментарів та текстових записів на предмет згадувань міст чи подій;
- наповнення бази даних шляхом додавання нових місць чи локацій;
- розробка нових моделей;
- накопичення бази для складання рейтингів місць.

### **3.2 Опис засобів реалізації**

Програму було реалізовано на прототипно-орієнтованій сценарній мові програмування JavaScript.

У якості 3D-інтерпретатора на веб-ресурсі, було використано JavaScript бібліотеку Three.js., що на сьогоднішній день є найбільш функціональною та підтримуваною. У свою чергу, сама бібліотека при рендері зображення опирається на WebGL чи безпосередньо Canvas, залежно від того, чи підтримує підтримує ваш браузер WebGL.

WebGL (web-Graphics Library) — програмна бібліотека для мови програмування JavaScript, що дає змогу створювати на JavaScript 3D-інтерактивну графіку, що функціонує в широкому спектрі сумісних із нею Інтернет-браузерів. За рахунок використання низькорівневих засобів підтримки OpenGL, частина коду на WebGL може виконуватися безпосередньо на відкритих. WebGL — це контекст елемента полотна HTML, який забезпечує API 3D-графіки без використання плагінів. [25]

Three.js приймає модель у форматі, який завантажуються — json. При розробці моделей у 3D-редакторах особливу увагу, окрім безпосередньо розробки моделі, потрібно приділяти саме експорту моделі у json.

Додаткові бібліотеки, такі як: Three.js — яка дозволяє рендерити 3D-моделі, Instagram API, що містить у собі інструменти для доступу до порталу;



Openstreet map API — що дає змогу користуватись відкритою мапою та маніпулювати об'єктами на ній — є необхідними для виконання покладених задач на користувацький інтерфейс.[14]

### **3.2.1 Обґрунтування вибору середовища розробки**

Інтегроване середовище розробки (IDE) — це програмний пакет, який об'єднує основні інструменти, необхідні для написання та тестування програмного забезпечення.

Розробники використовують численні інструменти під час написання, створення та тестування програмного коду. Інструменти розробки часто включають текстові редактори, бібліотеки кодів, компілятори та тестові платформи. Без IDE розробник повинен вибирати, розгортати, інтегрувати та керувати всіма цими інструментами окремо. IDE об'єднує багато таких інструментів, пов'язаних з розвитком, як єдину рамку, додаток або послугу. Інтегрований набір інструментів призначений для спрощення розробки програмного забезпечення та дозволяє виявити та мінімізувати помилки розробки та кодування.[15]

Visual Studio Code поєднує простоту редактора вихідного коду з потужним інструментом для розробників, таким як завершення та налагодження коду IntelliSense.

Перш за все, це редактор, який виходить з вашої мети. Приємний цикл редагування-збирання-відлагодження означає менший час знайомства з оточенням та більше часу на виконання ваших ідей.

Visual Studio Code підтримує macOS, Linux та Windows — таким чином, можна почати втілювати свої задумки у швидкому темпі, незалежно від платформи.

Основні переваги:

- код Visual Studio підтримує роботу з TypeScript, JavaScript, Node.js та Mongo;

- наявні вбудовані відладчик та командна строка;
  - підтримка практично усіх мов програмування;
  - наявність вбудованої бібліотеки елементів коду;
  - автозавершення при вводі коду;
  - додавання в бібліотеку власник сніпетів;
  - підсвічування синтаксису;
  - одночасна робота з декількома проектами;
  - підтримування багатовіконного та двухпанельного режимів;
  - розширення функціоналу за допомогою плагінів;
  - інтеграція з Visual Studio Team Services, GitHub і Git;
  - наявність вбудованих засобів для тестування, збірки, упакування та розгортання додатків;
- публікація створених програмних продуктів в Microsoft Azure (посередництвом Visual Studio Team Services);
  - інтегрована система підказок;
  - командна робота над проектами;
  - широкий набір налаштувань і кросплатформеність.
- Основні можливості редактора:
- підказки IntelliSense: у будь-якому місці коду, натиснувши Ctrl+Space, буде виведено вікно підказки IntelliSense, а при наборі коду редактор буде показувати його автоматично;
  - підказки параметрів: для перевантажених функцій можна листати варіанти реалізації, використовуючи клавіші Up и Down;
  - сніпети коду: VS Code підтримує аббревіатури Emmet, які можна використовувати при редагуванні файлів HTML, Razor, CSS, Less, Sass, XML або Jade;
  - перехід до визначення символу: натиснувши F12 можна перейти до визначення символу, а якщо натиснути Ctrl і провести курсором по символу можна побачити в додатковому вікні;

- перехід до методу або змінної: натиснувши `Ctrl+Shift+O` відкриється перелік методів у файлі, на додачу можна натиснути двокрапку — методи будуть згруповані по типу. Натиснувши `Up` і `Down` треба обрати потрібний метод і курсор в редакторі перейде на строку, де знаходиться вибраний метод;

- перегляд визначення символу: якщо потрібно швидко побачити визначення символу або на референсах, для того, щоб згадати деталі реалізації або використання, можна використовувати пошук або перехід до визначення символів, що не завжди зручно, оскільки переключає контекст. У цьому випадку можна скористатися командами `Reference Search (Shift + F12)` або `Peek Definition (Alt + F12)` і результати пошуку будуть показані в інлайн вікні;

- перейменування методів та змінних: натиснувши `F2` можна змінити ім'я методу або змінної;

- відлагоджування: зберегти конфігураційний файл і вибрати пункт `Launch` у меню, встановити точку зупинки та натиснути `F5` для старту відладки, також відладчик дозволяє переглядати значення поточних змінних;

- контроль версій: VS Code інтегрований із системою контролю версій `git` і підтримує більшість команд, підтримуються локальні та віддалені репозитарії;

- конфлікти: дозволяє виконувати злиття та вирішення конфліктів при роботі з системою контролю версій.

### **3.2.2 Обґрунтування вибору мови програмування**

JavaScript — це одна з найпростіших, найуніверсальніших і найефективніших мов, яка використовується для розширення функціональності веб-сайтів. Служби розробки JavaScript допомагають легко використовувати екранні візуальні ефекти та обробляти й обчислювати дані на веб-сторінках. Мова програмування також допомагає розширити функціональність веб-сайтів, використовуючи сторонні сценарії, серед ряду інших зручних функцій.

Основні переваги JavaScript:

- 1) JavaScript клієнтоорієнтована мова.

# Client Object Model Mechanics

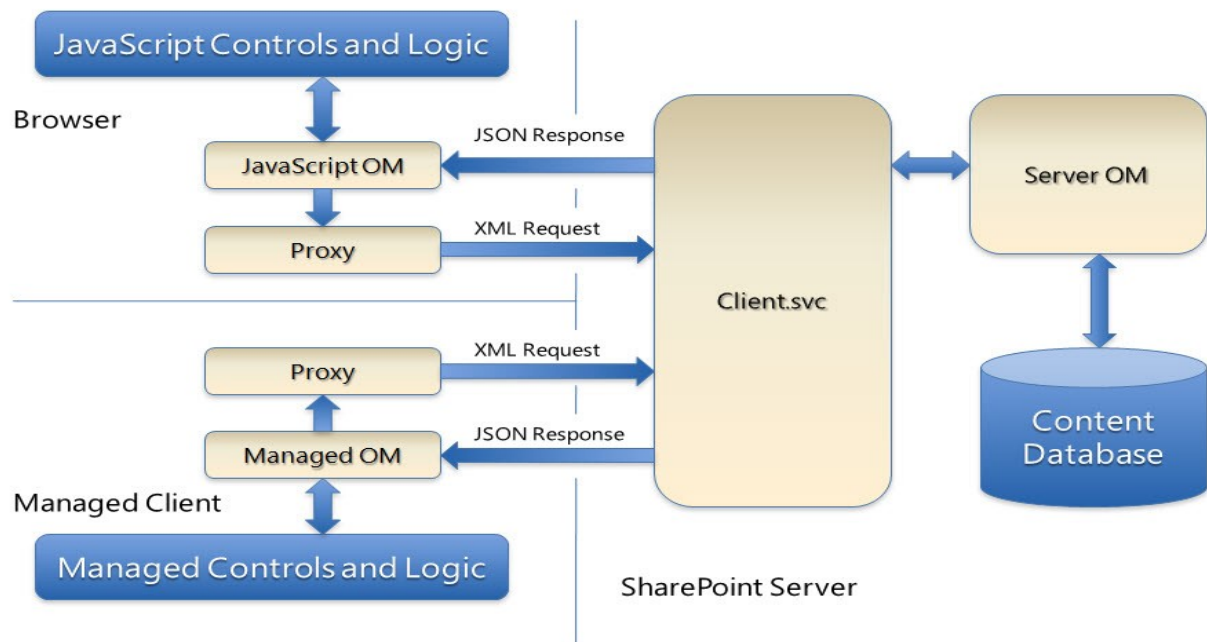


Рисунок 3.1 — Механіка об'єктної моделі клієнта

Код JavaScript виконується на процесорі користувача замість веб-сервера, таким чином він економить пропускну здатність і завантажується на веб-сервері.

2) JavaScript проста для вивчення. Мову JavaScript легка і вивченні, адже вона пропонує синтаксис, близький до англійської. Вона використовує модель DOM, яка надає безліч заздалегідь визначених функціональних можливостей для різних об'єктів на сторінках, що дозволяє легко розробити сценарій для вирішення власної мети.

3) JavaScript порівняно швидкий для кінцевого користувача. Оскільки код виконується на стороні клієнта, результати й обробка завершуються майже миттєво залежно від завдання (завдання в JavaScript на веб-сторінках зазвичай прості, щоб уникнути того, що програмою буде використовуватися надмірна кількість пам'яті), оскільки його не потрібно обробляти у веб-сервері сайту та надсилати назад користувачеві, що споживає локальну, а також пропускну здатність сервера.

4) Розширена функціональність веб-сторінок. Сторонні додатки, по типу Greasemonkey, дають змогу писати фрагменти JavaScript, які можуть виконувати на потрібних веб-сторінках, щоб розширити їхню функціональність. Якщо при користуванні веб-ресурсом виникає бажання увімкнути певну функцію, то її можна написати самостійно, або ж скористатися додатком на зразок зазначеного, для його реалізації на веб-сторінці.

5) Не потрібна компіляція. JavaScript не вимагає процесу компіляції, відповідно компілятор не потрібен. Браузер інтерпретує JavaScript як теги HTML.

6) Легко налагоджувати та тестувати. Розуміння синтаксису JavaScript дуже легке, тому будь-яка людина може його вивчити та використовувати для розробки динамічних та масштабованих веб-сайтів.

7) Платформа незалежна. Будь-який браузер із підтримкою JavaScript може розуміти та інтерпретувати JavaScript-код. Будь-який код JavaScript може бути виконаний на різних видах обладнання, для яких написана програма JavaScript.

8) Мова програмування на основі подій. Будучи мовою на основі подій, різні сегменти коду виконуються щоразу, коли певна подія відбувається в JavaScript. Тобто, сегмент коду виконується, коли користувач натискає кнопку або переміщує курсор миші на об'єкт.

9) Можливості процесуального програмування. Мова JavaScript охоплює всі можливості процесуальної мови. Розгалуження, циклічність, перевірка стану — деякі з тих можливостей, які можна виконати на веб-сторінці. [16]

Найпростіше визначення Node.JS — це середовище виконання JavaScript, яке допомагає у виконанні коду JavaScript на стороні сервера. Цей багатоплатформовий JavaScript з відкритим вихідним кодом допомагає в розробці мережеских додатків у реальному часі.

Переваги Node JS:

1) Легка масштабованість. Node.JS вважається найкращим варіантом для мікро-сервісної архітектури, також Node.JS — делікатна технологія.

Розробникам досить просто масштабувати додатки як по горизонталі, так і по вертикалі. Насправді, легко додати більше мікро-послуг поверх вже існуючих.

2) Більш висока продуктивність. Node.JS запускає код JavaScript через движок Google V8 JavaScript і компілює код безпосередньо в машинний. Цей процес полегшує та прискорює ефективну реалізацію кодів.

3) Єдина мова програмування. Node.JS допомагає написати серверний додаток на JavaScript. Це дозволяє розробникам писати інтерфейсні та фонові веб-додатки на JavaScript з використанням середовища виконання.

4) Свобода розробки додатків. Node.JS пропонує свободу розробки додатків і програмного забезпечення, на відміну від іншої платформи, яка накладає певні керівні принципи.

5) Обробляти кілька запитів одночасно. Node.JS допоможе обробляти різні запити одночасно краще, ніж інші, включаючи Ruby on Rails і Python.

6) Розширена підтримка. Розробники Node.JS отримують розширену підтримку різних інструментів. Наприклад, можна протестувати вихідний код Node.JS, використовуючи Jasmine й інші подібні інструменти.[17]

### **3.3 Розробка інструкцій із використання додатку**

Інтерфейс — найважливіша частина системи, елемент, за допомогою якого відбувається взаємодія системи та користувача. Від інтерфейсу залежить, чи зможе користувач впевнено й ефективно нею керувати.

#### **3.3.1 Інструкція з підключення пристрою віртуальної реальності**

VR-окуляри налаштовуються та підключаються до смартфона відповідно до нижчеописаної інструкції.

Насамперед потрібно викачати потрібний VR-додаток на свій смартфон і запустити його. Найчастіше, можна відразу побачити, що екран смартфона «роздвоївся» — це нормально, лякатися не потрібно.



Рисунок 3.2 — Вигляд екрану смартфона із «роздвоєнням»

Наступним кроком треба встановити свій смартфон у VR-гарнітуру. Залежно від конкретної моделі спосіб установки трішки відрізнятиметься. Наприклад, в окулярах VR-BOX виймається спеціальна планка, де між зажимами на пружинах встановлюється смартфон, після чого планку вставляють назад в окуляри.



Рисунок 3.3 — Вигляд процесу встановлення смартфона у VR-окуляри

Якщо виникають проблеми з зображенням, то можна спробувати відрегулювати лінзи в своїх окулярах. Регулювання є не у всіх моделях, але наприклад в VR-BOX (які буде використано для демонстрації) можна

перемістити лінзи ближче до перенісся, а також відсунути або навпаки, наблизити їх до очей.

Також, якщо є можливість, для більшої зручності та кращого взаємозв'язку між елементами можна використовувати джойстик, але це не є обов'язковим.

Часто зустрічається така проблема, коли екран смартфона поділений неправильно, і одягнувши VR-окуляри, користувач продовжує бачити два окремих екрани. Якщо регулювання лінз тут не допомагає (а найчастіше так і буде), потрібно використовувати спеціальний додаток під назвою Cardboard.

Щоб установити Cardboard пройдіть декілька кроків:

- 1) скачати, встановити та відкрити додаток;
- 2) після запуску додаток визначає модель ваших окулярів і якщо все вірно — натиснути «так», в іншому випадку — «ні, обрати окуляри» (рис. 3.4);

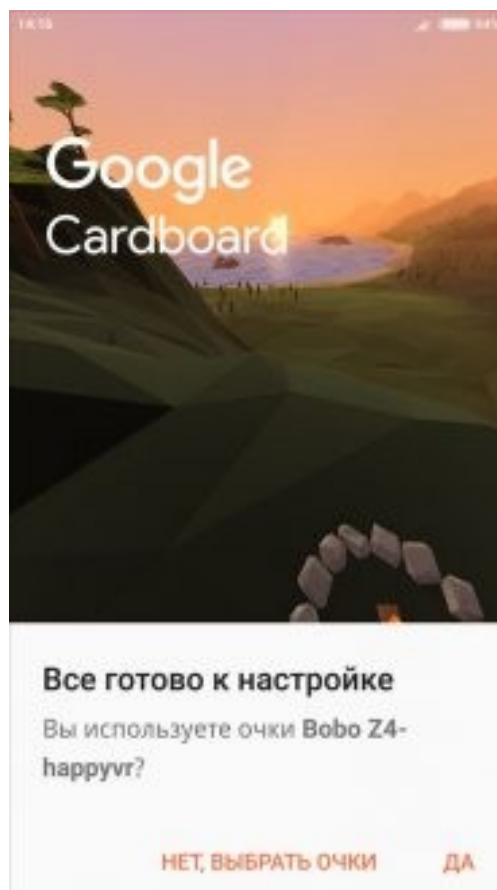


Рисунок 3.4 — Вигляд процесу визначення моделі окулярів



3) надати додатку доступ до камери та просканувати штрих-код з VR-гарнітури, якщо він там звичайно є, у протилежному випадку — знайти на офіційному сайті модель своєї гарнітури та відповідний їй код, після чого програма автоматично налаштує смартфон для роботи з вашими окулярами, тобто підбере потрібні кути та співвідношення поділу екрану (рис. 3.5).

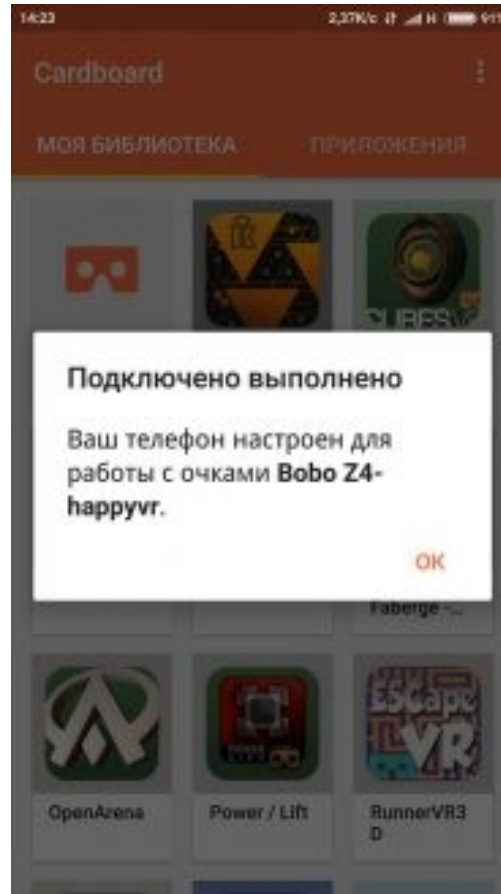


Рисунок 3.5 — Вигляд сповіщення про успішне підключення окулярів до смартфона

Якщо користувач вирішив використовувати джойстик, то наступна інформація та інструкція саме для цього.

Такі джойстики бувають різних моделей і відрізняються один від одного не лише зовні, але й по функціоналу. Частіше за все геймпад йде в комплекті з VR-гарнітурою і виглядає так, як показано на рисунку 3.6.



Рисунок 3.6 — Вигляд Bluetooth-геймпаду

Універсальна інструкція з підключення Bluetooth-геймпаду:

1) увімкнути Bluetooth на смартфоні й одночасно джойстик, натиснувши кнопку Start або Home (рис. 3.7);

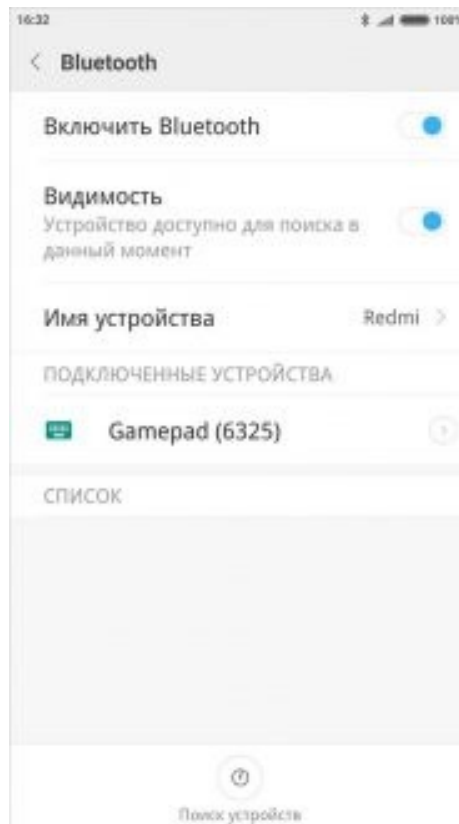


Рисунок 3.7 — Вигляд вікна налаштувань зв'язку смартфон-геймпад

2) зачекати, поки смартфон знайде новий пристрій, і натиснути на нього для сполучення. Після успішного підключення має з'явитися повідомлення, зображене на рисунку 3.8.

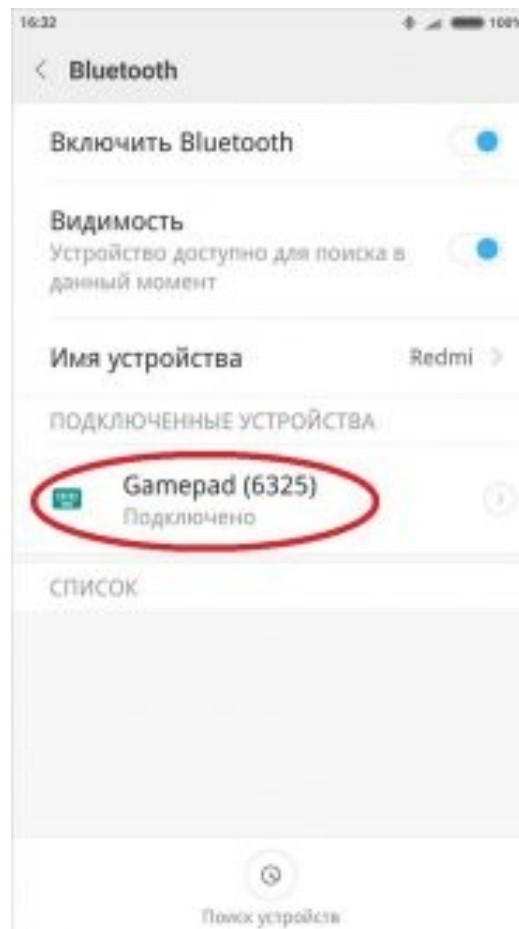


Рисунок 3.8 — Вигляд повідомлення про успішне підключення геймпаду

Важливо! На стандартних геймпадах, що йдуть у комплекті з окулярами, збоку є перемикач (рис. 3.9). Для того, щоб геймпад коректно працював, потрібно поставити його в режим Game. Водночас, режим Key потрібен для управління смартфоном у звичайному стані (наприклад, увімкнути дистанційно аудіоплеєр).



Рисунок 3.9 — Зовнішній вигляд перемикача режимів геймпада

Схеми підключення пристрою віртуальної реальності для подальшого його використання з розробленим програмним продуктом можна зобразити у вигляді узагальненої схеми, зображеної на рисунку 3.10.

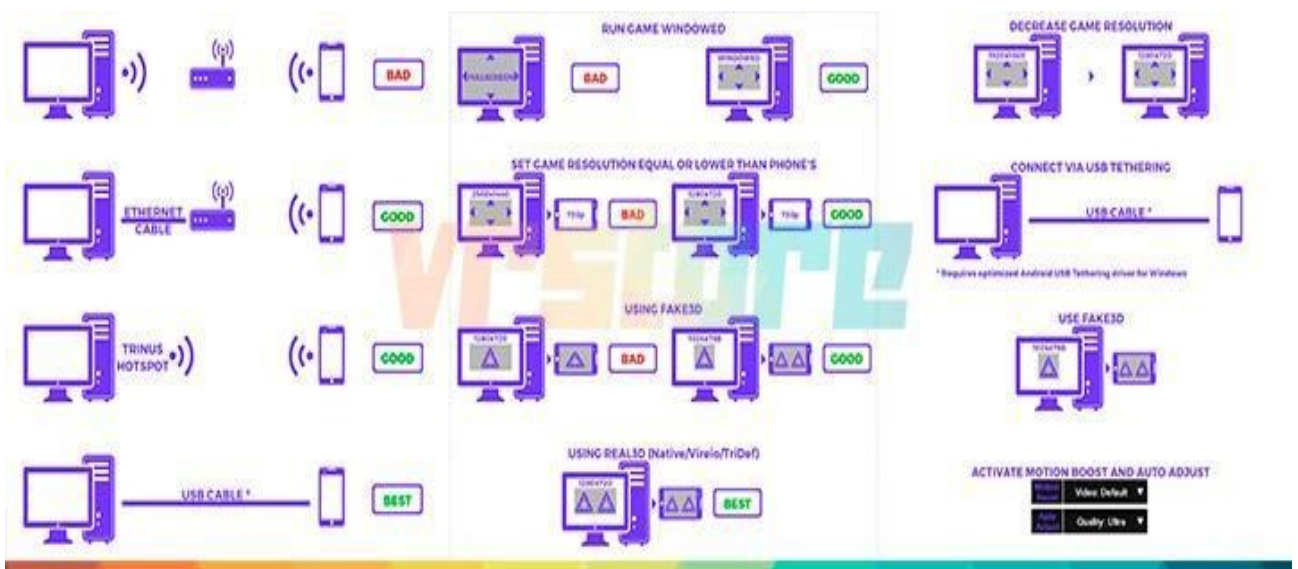


Рисунок 3.10 — Схеми підключення пристрою віртуальної реальності

Коли пристрій налаштований і готовий до роботи з ним, можна приступати до наступного кроку.

### 3.3.2 Інструкція користувача

Для початку роботи безпосередньо з програмним продуктом, користувачеві потрібно відкрити посилання у браузері на смартфоні або комп'ютері, якщо останній оснащено гіроскопом (рис. 3.11).

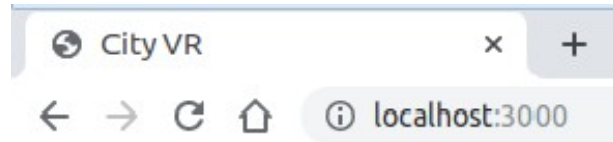


Рисунок 3.11 — Вигляд посилання в адресній стрічці браузера

Після відкриття посилання, на головному екрані користувачеві буде відображено форму авторизації в Instagram-акаунті, що містить поля вводу логіну та паролю (рис. 3.12).

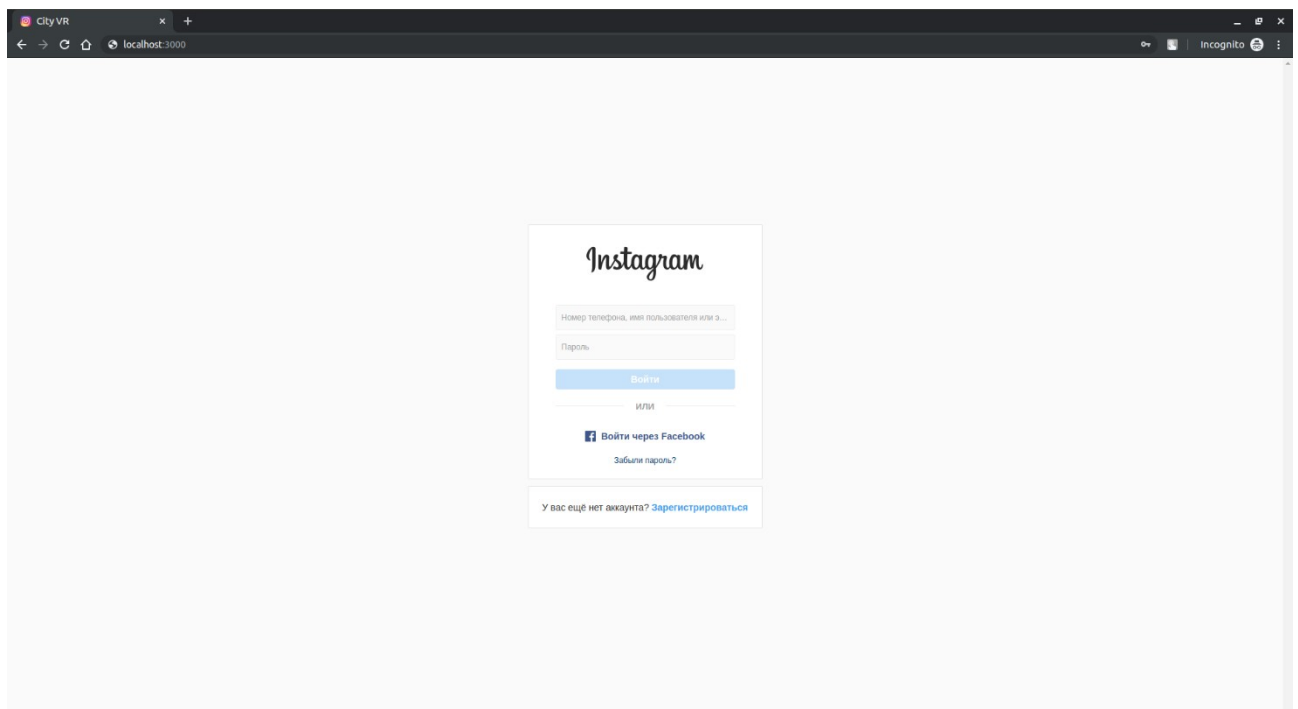


Рисунок 3.12 — Вигляд форми авторизації в акаунті

У разі успішної авторизації, на екрані користувача відобразатиметься карта, з деякою областю і так званим поінтером (рис. 3.13). Область виділення, що розгортається, не випадкова. Аналізується інформація, отримана з

останнього поста під'єданого Instagram-акаунту, що стосується місцезнаходження користувача за позначеною ним геолокацією.

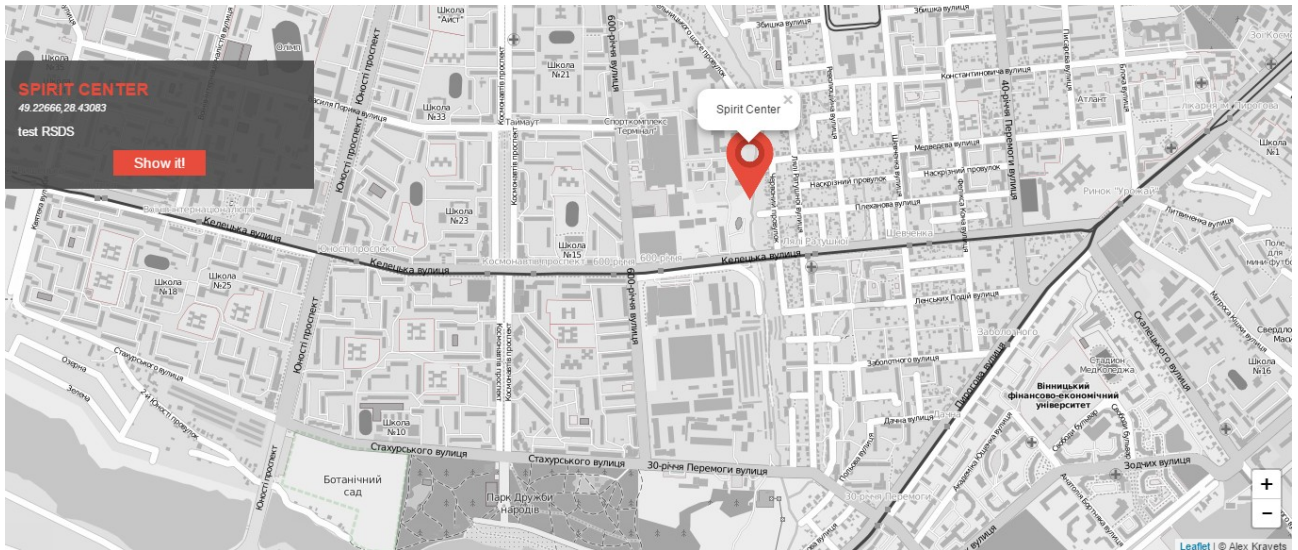


Рисунок 3.13 — Вигляд карти з позначкою останнього місцерозташування користувача

У верхньому лівому куті можна побачити невеличке інформативне вікно, власне на якому знаходиться інформація стосовно точних географічних координат «поінтера», а також кнопка «Show it!» (рис. 3.14).



Рисунок 3.14 — Вигляд інформативного вікна

Для зручності, у нижньому лівому куті екрану, користувачеві буде запропоновано список рекомендацій найбільш популярних та відвідуваних місць відносно його поточної геолокації (рис. 3.15). Важливим моментом є те, що користувач може сам знаходити історичні будівлі, які він хотів би переглянути, але які знаходяться не в області видимості: карту можна масштабувати (як зменшувати, так і збільшувати) та рухати.

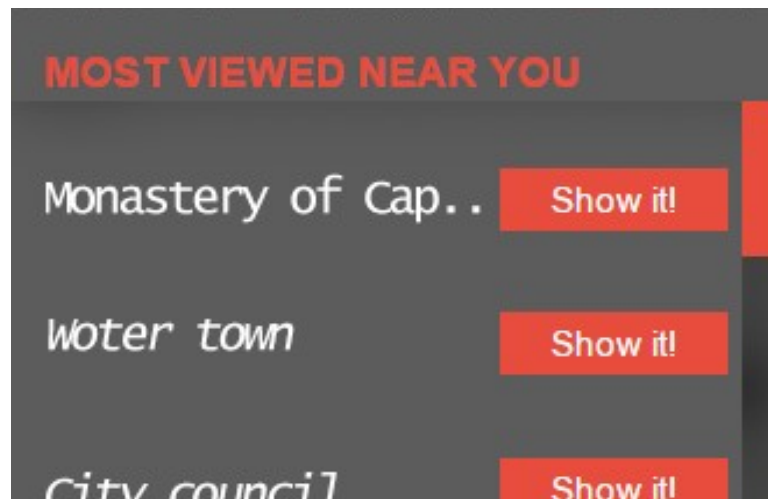


Рисунок 3.15 — Вигляд вікна рекомендацій

Після натиснення на кнопку «Show It!» відбудеться завантаження 3D моделі, і до моменту повного завантаження буде відображатись лодер (рис. 3.16).



Рисунок 3.16 — Вигляд вікна завантаження 3D-моделі

### 3.4 Тестування роботи програмного додатку

Чек-лист — це перелік тестів, які потрібно виконати у визначеній процедурі. Це допомагає зрозуміти, чи тестування виконано повністю та наскільки успішно. Це також допомагає формалізувати тестування окремо взятої функціональності, вносячи тести в список. Тестовий порядок у контрольному списку може бути суворим, а також випадковим.

Для тестування розробленого веб-ресурсу було складено чек-лист та виконано усі пункти з нього:

1) Відображення елементів сторінок:

- перевірити коректне відображення карти на головному екрані;
- перевірити коректне відображення кнопок, блоків, меню і т.п.;
- перевірити коректне відображення спливаючих вікон з помилками/сповіщеннями;
- перевірити чи величина елементів дає змогу користувачеві потрапити на них;

2) дії на сторінці:

- перевірити авторизацію користувача;
- перевірити коректність прокрутки;
- перевірити функцію масштабування карти;
- перевірити працездатність кнопок;
- перевірити доступність усіх елементів;
- перевірити швидкість відгуку елементів на дії з ними;

3) функції на пристроях:

- перевірити горизонтальне та вертикальне відображення;
- перевірити у режимі розділення екрану;

4) функціональність:

- перевірити коректність визначення геолокації;
  - перевірити коректність розташування поінтеру на карті відповідно до отриманих координат;
  - перевірити відкриття 3D-моделі після натиснення на поінтер;
  - перевірити чи оптимізовано час завантаження сторінки;
  - перевірити коректність відтворення 3D-моделі у веб-ресурсі;
  - перевірити відображення списку популярних місць відносно поточного місцезнаходження;
  - перевірити відсутність «крешів»;
  - зберігання даних в кеш браузера;
- 5) реакція на сторонні переривання:
- перевірити реакцію на вхідні та вихідні виклики;



- перевірити реакцію на зарядку пристрою;
- перевірити реакцію на блокування та розблокування пристрою.

Наступним етапом буде створення та проходження наборів тест-кейсів для детальнішого тестування веб-ресурсу (табл. 3.1-3.3).

Тест-кейс — це набір умов або змінних, за яких тестувальник визначатиме, чи відповідає система, що перевіряється, нормам. Процес розробки тестових випадків також може допомогти знайти проблеми у вимогах або дизайні програми.

Тестовий випадок — це документ, який містить набір даних тесту, передумов, очікуваних результатів та постумов, розроблених для конкретного сценарію тестування з метою перевірки відповідності конкретній вимозі. Тестовий випадок виступає початковою точкою для тестового виконання, і після застосування набору вхідних значень додаток має остаточний результат і залишає систему в певній кінцевій точці. [26]

Таблиця 3.1 — Позитивний сценарій тестування

Дія	Результат
На головному екрані веб-ресурсу в поля форми реєстрації ввести валідні дані для входу в Instagram-акаунт	Відбувається вхід в обліковий запис, з якого отримується інформація про останнє місце розташування та відображається у вигляді поінтеру на карті
Натиснути на поінтер на карті	Відображається 3D-модель, що відповідає останньому позначеному місцезнаходженню
Перемістити поінтер на будь-яку область на карті	Поінтер буде закріплено на обраному місці та призначено йому географічні координати
Обрати будь-яку доступну 3D-модель на карті	На головному екрані буде відображено обрану модель

## 4 ЕКОНОМІЧНА ЧАСТИНА

В магістерській кваліфікаційній роботі розробляється програмний засіб для VR відображення історичних пам'яток.

Для ефективного вкладення капіталу необхідна попередня підготовка техніко-економічного обґрунтування (ТЕО) інноваційного проекту. Значне місце цього етапу в інноваційному процесі обумовлюється тим, що чим більше вірогідної та грамотно оформленої інформації про підприємство (проект) одержить інвестор, тим менше на нього чекає ризику на етапі реалізації проекту.

Оскільки інновації за своїм змістом і складом наближаються до реальних інвестицій, то насамперед реальні інвестиційні проекти потребують розробки ТЕО. [27]

Тому в економічній частині магістерської роботи буде виконано такі етапи робіт:

- оцінювання комерційного потенціалу розробки;
- прогнозування витрат на виконання наукової роботи та впровадження її результатів;
- прогнозування комерційних ефектів від реалізації результатів розробки;
- розрахунок ефективності вкладених інвестицій та періоду їх окупності.

### 4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Для проведення технологічного аудиту було залучено 3-х незалежних експертів. Такими експертами є: керівник магістерської роботи — доц каф. ПЗ Черноволик Галина Олександрівна, ктн., доц каф. ПЗ — Ракитянська Ганна Борисівна, ктн., доц каф. ПЗ — Майданюк Володимир Павлович.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями, наведеними в таблиці 4.1.

Таблиця 4.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка.

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
№	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

## Продовження таблиці 4.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому Комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.2.

Таблиця 4.2 — Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1. Черноволик Г.О.	2. Ракитянська Г.Б.	3. Майданюк В.П.
	Бали, виставлені експертами:		
1	4	3	4
2	2	2	1
3	4	4	4
4	4	3	3
5	4	4	4
6	3	2	2
7	3	3	3
8	2	2	3
9	1	1	1
10	3	4	3
11	4	4	4
12	4	4	4
Сума балів	СБ <sub>1</sub> = 38	СБ <sub>2</sub> = 36	СБ <sub>3</sub> = 36
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{38+36+36}{3} = 37$		

Отже, з отриманих даних таблиці 4.2 видно, що нова розробка має рівень комерційного потенціалу вище середнього.

Реалізація розробки відбуватиметься через розповсюдження у вигляді ліцензійної угоди на певний термін дії та надання технічної підтримки, що зацікавить клієнтів.

Якість виконання даної розробки підтверджена мануальним тестуванням, не поступається подібним програмним продуктам, вирізняється новизною рішень та відповідає стандартам ISO.

Шляхом зіставлення розробленого програмного засобу для VR відображення історичних пам'яток із використанням мобільних пристроїв з показниками найближчого аналога, основні функціональні характеристики зведено до таблиці 4.3.

Таблиця 4.3 – Порівняння функціональних характеристик з аналогом

Показники	Аналог	Розробка	
1. Швидкодія	83%	95%	1,44
2. Зручність	75%	92%	1,23
3. Функціональність	64%	96%	1,5
4. Підтримка	87%	100%	1,15

Переваги розробки полягають у тому, що вона зручна та зрозуміла у використанні, це веб-додаток, який миттєво запускається як на десктопах, так і на смартфонах, конвертація 3д моделей в json гарантує швидке завантаження «важких» моделей будівель, завдяки логізації в Instagram користувач опиняється у відзначеному місці з останнього опублікованого поста.

Розробка орієнтована на використання у сфері туризму, у навчальних цілях, на людей, які цікавляться історією та пам'ятками архітектури, на активних користувачів соцмереж (Instagram), тощо.

Технічна готовність розробленого програмного засобу уже реалізована у вигляді Android додатку, до якого написана інструкція користувача, та може бути впроваджена для продажу. Якщо розглядати існуючий стан робіт із комерціалізації розробки, то можна зазначити про проведення переговорів щодо впровадження даної системи для працівників організації з метою їхнього культурного розвитку.

#### **4.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько–технологічної роботи**

Прогнозування витрат на виконання науково-дослідної та конструкторсько-технологічної роботи складається з таких етапів:

- розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;
- розрахунок загальних витрат на виконання даної роботи;
- прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Основна заробітна плата для розробників визначається за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де  $M$  — місячний посадовий оклад конкретного розробника (інженера,

дослідника, науковця тощо), грн.;

$T_p$  — кількість робочих днів у місяці,  $T_p = 22$  дні;

$t$  — число робочих днів роботи розробника.

Розрахунки заробітних плат для наукового керівника та розробника наведені в таблиці 4.4.

Таблиця 4.4 — Розрахунки основної заробітної плати

Найменування посади виконавця	Місячний оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на оплату праці, грн.
Науковий керівник	7000	318	3	954
Інженер-програміст	12500	568	66	37488
Усього:				38442

Розрахуємо додаткову заробітну плату:

$$Z_{\text{дод}} = (0,1 \dots 0,12) \cdot Z_0, \quad (4.2)$$

$$Z_{\text{дод}} = 0,1 \cdot 38442 = 3844 \text{ (грн.)}$$

Нарахування на заробітну плату  $H_{\text{зп}}$  розробників, які брали участь у виконанні даного етапу роботи, розраховується за формулою:

$$H_{\text{зп}} = (Z_0 + Z_{\text{дод}}) \cdot \frac{\beta}{100}, \quad (4.3)$$

де  $Z_0$  — основна заробітна плата розробників, грн.;

$Z_{\text{дод}}$  — додаткова заробітна плата всіх розробників, грн.;

$\beta$  — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %.

$$H_{\text{зп}} = (38442 + 3844) \cdot \frac{22}{100} = 9303 \text{ (грн.)}$$

Амортизаційні відрахування за прямолінійним методом розраховується за такою формулою:

$$A = \frac{Ц}{T} \cdot \frac{T_v}{12}, \quad (4.4)$$

де  $Ц$  — балансова вартість обладнання, грн.;

$T$  — корисний час використання;

$T_v$  — термін використання.

Отже, розрахуємо амортизаційні відрахування та зведемо результати до таблиці 4.5:

$$A_k = \frac{200000}{20} \cdot \frac{3}{12} = 2500 \text{ (грн).}$$

$$A_k = \frac{21000}{2} \cdot \frac{3}{12} = 2625 \text{ (грн).}$$

$$A_c = \frac{7000}{2} \cdot \frac{1}{12} = 292 \text{ (грн).}$$

Таблиця 4.5 — Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення, приміщень тощо	Балансова вартість, грн.	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
Приміщення	200000	3	2500
Комп'ютер	21000	3	2625
Смартфон	7000	1	292
Усього:			5417

Витрати на матеріали: папір — 100 грн., канцелярія — 45 грн, VR-окуляри — 270 грн., інтернет за 3 місяці — 507 грн. Разом ця стаття витрат складає 922 грн.

Під час створення програмного продукту використовувалось безкоштовне програмне забезпечення.

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_n, \quad (4.5)$$

де  $V$  — вартість 1кВт-години електроенергії ( $V = 0,9$  грн/кВт);

$\Pi$  — установлена потужність комп'ютера ( $\Pi = 0,3$  кВт);

$\Phi$  — фактична кількість годин роботи комп'ютера ( $\Phi = 528$  год);

$K_n$  — коефіцієнт використання потужності ( $K_n < 1$ ,  $K_n = 0,8$ ).

Потужність комп'ютера складає 230 Вт/год = 0,23 кВт/год + потужність на освітлення, тоді  $\Pi = 0,3$  кВт/год.



$$V_e = 0,9 \cdot 0,3 \cdot 528 \cdot 0,8 = 114 \text{ (грн)}.$$

Розрахуємо інші витрати  $V_{ін}$ .

Інші витрати  $I_v$  можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot Z_0, \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 3 \cdot 38442 = 115326 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даного етапу роботи —  $V$ .

$$V = 38442 + 3844 + 9303 + 5417 + 922 + 114 + 115326 = 173398 \text{ (грн)}.$$

Розрахуємо загальну вартість наукової розробки  $V_{заг}$  за формулою:

$$V_{заг} = \frac{V}{\alpha}, \quad (4.7)$$

де  $\alpha$  — частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відносних одиницях = 1.

$$V_{заг} = \frac{173398}{1} = 173398 \text{ (грн)}.$$

Прогнозування загальних витрат на виконання та впровадження результатів виконаної наукової розробки здійснюється за формулою:

$$ЗВ = \frac{V_{заг}}{\beta}, \quad (4.8)$$

де  $\beta$  — коефіцієнт, який характеризує етап виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{173398}{0,9} = 192664 \text{ (грн)}.$$

### **4.3 Прогнозування комерційних ефектів від реалізації результатів розробки**

Магістерська робота містить економічну частину обґрунтування економічної доцільності розробки програмного засобу для VR відображення історичних пам'яток, для того, щоб виконати дану розробку потрібно 66

робочих днів. Дана розробка вважається економічно вигідною, якщо її окупність становитиме 1 рік.

Термін окупності витрат на розробку становитиме 0,62 роки, тому розробка нового методу буде економічно ефективною. Новий метод є економічно доцільним для споживача. Отримані вище результати доводять, що цей метод є більш корисніший та необхідніший для нової розробки.

Зростання чистого продукту для даного методу можна оцінити у теперішній вартості грошей. Зростання чистого прибутку забезпечить підприємству надходження додаткових коштів, що покращить фінансові результати діяльності.

Моніторинг ринку показує, що кількість підприємств, які потенційно зацікавлені у використанні нашої розробки в Україні складає приблизно 250 шт. (в розрахунку на рік). Реальний попит на розробку може становити 10-20 шт. за рік. Середня ціна подібної розробки, яка виконує аналогічні функції, дорівнює приблизно 35 тис. грн. Але оскільки запропонована нами мобільна система має значно кращі технічні параметри та функціональні можливості, то це дозволяє реалізовувати нашу розробку приблизно на 20-30% дорожче.

Припустимо, що наша розробка буде користуватися підвищеним попитом на ринку протягом 3-років після впровадження.

Результати нашої розробки можуть бути впроваджені вже з 1 січня 2020 року, а її результати будуть виявлятися протягом 2020-го, 2021-го та 2022-го років. Прогноз попиту на розробку складає по роках:

- 1-й рік після впровадження (2020 р.) — приблизно 19 шт.;
- 2-й рік після впровадження (2021 р.) — приблизно 36 шт.;
- 3-й рік після впровадження (2022 р.) — приблизно 49 шт.

На 4-й рік (2023 р.) не планується отримання прибутків, оскільки високою є ймовірність, що будуть розроблені нові, більш ефективні системи подібного типу.

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку

підприємства  $\Delta \Pi_i$  для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta \Pi_i = \sum_1^n (\Delta \Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_i, \quad (4.9)$$

де  $\Delta \Pi_{\text{я}}$  — покращення основного якісного показника від впровадження результатів розробки у даному році;

$N$  — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$\Delta N$  — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$  — основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$n$  — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту  $\Delta \Pi_i$  протягом першого року складатиме:

$$\Delta \Pi_1 = 1 \cdot 25000 + 19 \cdot 45000 = 880000 \text{ (грн)}.$$

Протягом другого року:

$$\Delta \Pi_2 = 1 \cdot 25000 + (19 + 17) \cdot 45000 = 1645000 \text{ (грн)}.$$

Протягом третього року:

$$\Delta \Pi_3 = 1 \cdot 25000 + (19 + 17 + 13) \cdot 45000 = 2230000 \text{ (грн)}.$$

#### **4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності**

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1) Розраховано теперішню вартість інвестицій  $PV$ , що вкладаються в наукову розробку (прогнозована величина загальних витрат  $ЗВ$  на виконання та впровадження результатів НДДКР).  $ЗВ = PV$ .

2) Розраховано очікуване збільшення прибутку  $\Delta\Pi$ , яке отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження.

3) Для спрощення подальших розрахунків побудовано вісь часу, на яку наносяться всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів. Платежі показуються у ті терміни, коли вони здійснюються.

Рисунок, що характеризує рух платежів у грн. (інвестицій та додаткових прибутків) матиме вигляд, наведений на рисунку 4.1.

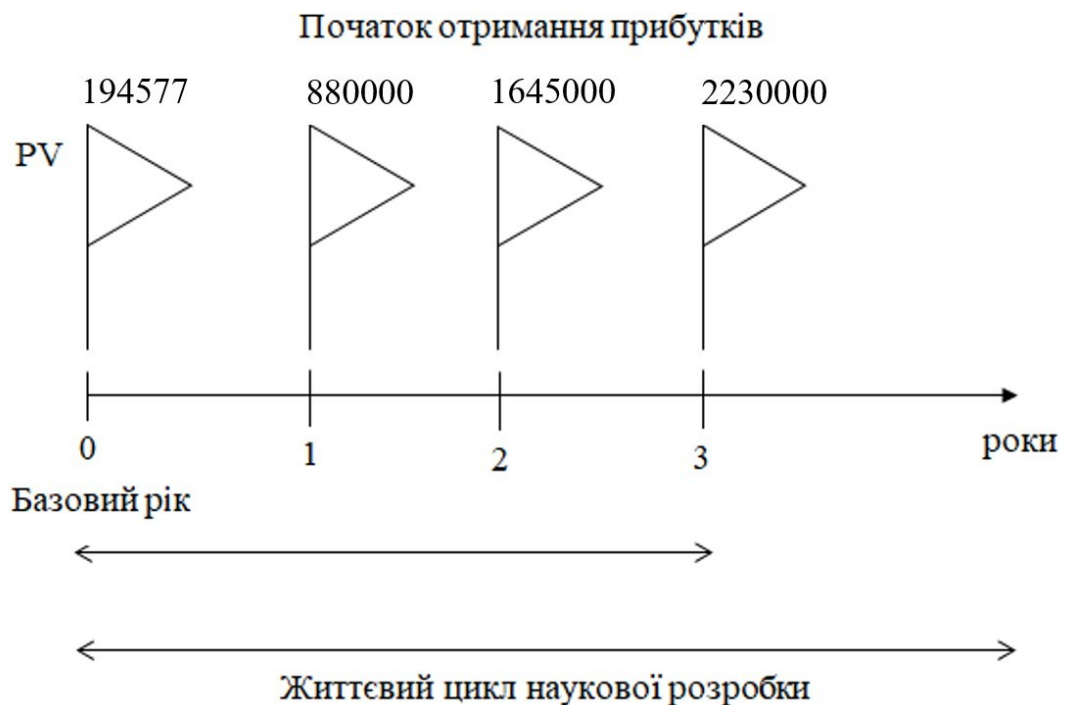


Рисунок 4.1 — Вісь часу з фіксацією платежів у грн., що мають місце під час розробки та впровадження результатів НДДКР

4) Абсолютна ефективність  $E_{абс}$  вкладених інвестицій розраховується за формулою:

$$E_{абс} = (ПП - PV), \quad (4.10)$$

де ПП — приведена вартість чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV — теперішня вартість інвестицій  $PV=ЗВ$ , грн.

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.11)$$

де  $\Delta\Pi_i$  — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн.;

t — період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

$\tau$  — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t — період часу (в роках) від моменту отримання чистого прибутку до точки «0».

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{880000}{(1 + 0,1)^2} + \frac{1645000}{(1 + 0,1)^3} + \frac{2230000}{(1 + 0,1)^4} = 3486409,5 \text{ (грн).}$$

Тоді розрахуємо  $E_{абс}$ :

$$E_{абс} = 3486409,5 - 192664 = 3293745,5 \text{ (грн).}$$

Оскільки  $E_{абс} > 0$ , то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

5) Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_v$  за формулою:

$$E_v = \sqrt[t_{ж}]{\left(1 + \frac{E_{абс}}{PV}\right)} - 1, \quad (4.12)$$

де  $E_{абс}$  — абсолютна ефективність вкладених інвестицій, грн.;

$PV$  — теперішня вартість інвестицій  $PV = 3B$ , грн.;

$T_{ж}$  — життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_b = \sqrt[3]{1 + \frac{3293745,5}{192664}} - 1 = 1,62 \text{ або } 162\%$$

Далі, розраховану величину  $E_b$  порівнюємо з мінімальною (бар'єрною) ставкою дисконтування  $\tau_{\min}$ , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування  $\tau_{\min}$  визначається за формулою:

$$\tau = d + f, \quad (4.13)$$

де  $d$  — середньозважена ставка за депозитними операціями в комерційних банках ( $d = 0,2$ );

$f$  — показник, що характеризує ризикованість вкладень, величина  $f = 0,1$ .

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки  $E_b = 162\% > \tau_{\min} = 30\%$ , то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

б) Термін окупності вкладених у реалізацію наукового проекту інвестицій  $T_{ок}$  розраховується за формулою:

$$T_{ок} = \frac{1}{E_b}, \quad (4.14)$$

$$T_{ок} = \frac{1}{1,62} = 0,62 \text{ (років)}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

#### 4.5 Висновки до розділу 4

В даному розділі було проведено оцінку комерційного потенціалу розробки програмного засобу для VR відображення історичних пам'яток, спрогнозовано витрати на виконання наукової роботи та впровадження її результатів, спрогнозовано комерційні ефекти від реалізації результатів

розробки та розраховано ефективність вкладених інвестицій та період їх окупності.

Проаналізувавши отримані результати, можна зробити висновок, що розробка даної системи є актуальною та доцільною з точки зору економічних розрахунків. Таким чином, відносна (щорічна) ефективність вкладених в наукову розробку інвестицій  $E_v$  становить 162%. Тому, обравши термін окупності даної наукової розробки, що складає 0,62 роки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

## ВИСНОВКИ

Віртуальні історичні пам'ятки являють собою інтегративну культурну форму, що характеризується структурною та функціональною різноманітністю. З одного боку, це нова технологія безконтактної інформаційної взаємодії користувача з музейним середовищем, комп'ютерна система, яка забезпечує певні візуальні ефекти. З іншого — віртуальне відображення 3D пам'яток як інформаційний ресурс має сутність медіа: це і засіб масової комунікації, і нова форма видавництва. Крім того, це вагомий елемент сучасного культурного середовища, що сприяє розвитку творчих здібностей, культурному обміну та спілкуванню користувачів.

Створений програмний засіб надає користувачам безоплатний доступ до визначних та цікавих місць. Він є каналом поширення культурних цінностей і долучення населення до культури.

У тому темпі, в якому розвивається наше місто, розробка програмних продуктів, що дозволять краще з ним ознайомитися, будуть доречними та перспективними. А отже, розробка, орієнтована на використання у сфері туризму, у навчальних цілях (до прикладу, під час уроків історії), на людей, які цікавляться історією та пам'ятками архітектури для свого саморозвитку, та просто на активних користувачів соцмереж (яких зараз переважна більшість), матиме великий попит.

Було покращено точність визначення останньої відзначеної користувачем геолокації шляхом отримання відповідної інформації з його Instagram-акаунту, для подальшого відображення користувачеві 3D-моделей історичних пам'яток відносно отриманої області виведеної на карту, підвищено швидкодію за рахунок конвертації розроблених «важких» 3D моделей будівель в json-формат, а також збільшено зручність в користуванні шляхом створення клієнтоорієнтованого web-ресурсу.

В процесі розробки:



- визначено сутність та значення сучасних методів представлення 3D об'єктів, їх переваги та недоліки;
- проаналізовано наявні методи та технології моніторингу геоданих користувачів;
- розроблено метод моніторингу та отримання даних про геолокацію із соцмережі для відображення 3D моделей у веб-додатку;
- спроектовано концептуальну, логічну та фізичну моделі веб-застосунку й описано принцип роботи розроблених засобів системи;
- спроектовано базу даних, що міститиме 3D-моделі історичних пам'яток;
- розроблено інструкцію користувача із підключення та використання ресурсу.

Отже, було розроблено метод та програмний засіб для VR 3D відображення історичних пам'яток.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

- 1) Сканируем в 3D с помощью фотоаппарата или 123D Catch. URL: <https://habrahabr.ru/post/134781/> (Дата обращения: 10.10.2019).
- 2) Дизайн и конструирование. URL: <http://ve-group.ru/3dvr-resheniya/dizayn-i-konstruirovanie/> (Дата обращения: 15.09.2019).
- 3) Разработка программ для систем виртуальной реальности. URL: <http://simmakers.ru/virtualnaya-realnost-razrabotka-programm/> (Дата обращения: 08.09.2019)
- 4) Виртуальная реальность: История, теория, практика. URL: <http://itc.ua/articles/virtualnaya-realnost-istoriya-teoriya-praktika/> (Дата обращения: 06.09.2019)
- 5) Каролін Бегг, Томас Конноллі. Бази даних. Проектування, реалізація та підтримка. Теорія і практика. – М.: Вільямс, 2003 – 1436 с.
- 6) Степанов Олександр, Пол Мак-Джонс Начала программирования = Elements of Programming (Hardcover). — М.: «Вильямс», 2011. — С. 272.
- 7) Петюшкін Олексій–HTML. Експрес-курс–Спб.: БВХ-Петебург, 2003–256 с.
- 8) Дополненная, виртуальная и прочие реальности. URL: <https://www.it.ua/knowledge-base/technology-innovation/dopolnennaja-virtualnaja-i-prochie-realnosti> (Дата обращения: 06.09.2019)
- 9) Смірнова І.Є. – Початок web-дизайну – Спб.: БВХ-Петебург, 2003 – 252 с.
- 10) Хольцшлаг М. Використання HTML і XHTML. Спеціальне видання. М.: «Вильямс», 2003 – 736 с.
- 11) Електронні інформаційні ресурси: створення, використання, доступ: Збірник матеріалів Міжнародної науково-практичної Інтернет конференції. – Вінниця : ВНТУ, 2019. – 292 с.

- 12) Соціальна мережа. URL: [https://uk.wikipedia.org/wiki/Соціальна\\_мережа](https://uk.wikipedia.org/wiki/Соціальна_мережа) (Дата обращения: 01.10.2019).
- 13) Google Arts & Culture URL: <https://artsandculture.google.com/> (Дата обращения: 01.10.2019).
- 14) Гриншкун А.В. Технология дополнительной реальности и подходы к ее использованию при создании учебных заданий для школьников.//Вестник Московского городского педагогического университета. Серия: Информатика и информатизация образования. 2017. № 3 (41). С. 99-105.
- 15) 3dspace: VR for Web Developers. URL: [3dspace.com/2016/04/vrfor-web-developers/](http://3dspace.com/2016/04/vrfor-web-developers/) (Дата доступа: 20.11.2019).
- 16) Hi-News.ru: Крупнейшие vr-компании сформировали глобальную ассоциацию виртуальной реальности. URL: <https://hinews.ru/technology/krupnejshie-vr-kompanii-sformirovali-globalnuyuassociaciyu-virtualnoj-realnostil> (Дата доступа: 01.06.2017.)
- 17) Holographica: Google опубликовала приложение для создания VR-фотографий. URL: <http://holographica.space/news/googlecardboard-camera-android-2511>. (Дата доступа: 22.10.2019.)
- 18) WhatIs.com: room-scale VR. URL: <http://whatis.techtarget.com/definition/room-scale-VR-room-scale-virtualreality>. (Дата доступа: 05.11.2019).
- 19) Designmodo: Virtual Reality and Web Design: Everything You Need to Know. URL: [designmodo.com/virtual-reality-design/](http://designmodo.com/virtual-reality-design/). (Дата доступа: 21.09.2019).
- 20) Co.Design: Why A Virtual Reality Web May Never Happen. URL: <https://www.fastcodesign.com/3058591/why-a-virtual-reality-web-may-neverhappen>. (Дата доступа: 02.10.2019).
- 21) Instructables: DIY You Virtual Reality Headset - Google Cardboard URL: <http://www.instructables.com/id/DIY-You-Virtual-Reality-HeadsetGoogle-Cardboard/>. (Дата доступа: 16.10.2019).
- 22) XinReality: Timewarp. URL: <https://xinreality.com/wiki/Timewarp>. (Дата доступа: 22.11.2019).

- 23) Офіційний сайт проекту WebVR. URL: <https://webvr.info/developers/>.  
(Дата доступу: 12.08.2019).
- 24) Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.
- 25) Електронні інформаційні ресурси: створення, використання, доступ: Збірник матеріалів Міжнародної науково-практичної Інтернет конференції. – Вінниця: ВНТУ, 2019. – 292 с.
- 26) Офіційний опис проекту WebVR. URL: <https://github.com/w3c/webvr/blob/master> (Дата доступу: 12.08.2019).
- 27) Документація WebVR. URL: <https://w3c.github.io/webvr/spec/1.1/>.  
(Дата доступу: 12.08.2019).
- 28) Офіційний блог Google для розробників. URL: <https://developers.google.com/web/fundamentals/vr/getting-started-withwebvr/>.  
(Дата доступу: 17.09.2019).
- 29) Документація до бібліотеки WebGL. URL: [https://developer.mozilla.org/uk/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/uk/docs/Web/API/WebGL_API). (Дата доступу: 4.10.2019).
- 30) Документація до Three.js. URL: <https://threejs.org/docs/index.html#manual/introduction/Creating-a-scene>. (Дата доступу: 7.08.2019).
- 31) Tony Parisi. Learning Virtual Reality / Tony Parisi // O'Reilly Media, Inc. – 2015. – First Edition. – С. 75-97.

## **ДОДАТКИ**

**Додаток А. Технічне завдання**  
Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ  
д.т.н., проф. О. Н. Романюк  
" \_\_\_\_ " \_\_\_\_\_ 2019 р.

**Технічне завдання**  
**на магістерську кваліфікаційну роботу**  
**«Розробка методу та програмного засобу для VR 3D відображення**  
**історичних пам'яток»**  
**за спеціальністю 121 – Інженерія програмного забезпечення**

Керівник магістерської кваліфікаційної роботи:

\_\_\_\_\_ к.т.н., доц. Г.О. Черноволик  
" \_\_\_\_ " \_\_\_\_\_ 2019 р.

Виконав:

\_\_\_\_\_ студент гр. 2ПІ-18м Д.В. Гончарук  
" \_\_\_\_ " \_\_\_\_\_ 2019 р.

## **1. Найменування та галузь застосування**

Магістерська кваліфікаційна робота: «Розробка методу та програмного засобу для VR 3D відображення історичних пам'яток».

Галузь застосування — туристичний помічник.

## **2. Підстава для розробки.**

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ №\_\_ ректора по ВНТУ про закріплення тем МКР.

## **3. Мета та призначення розробки.**

Метою роботи є покращення точності визначення останньої відзначеної користувачем геолокації шляхом отримання відповідної інформації з його Instagram-акаунту, для подальшого відображення користувачеві 3D-моделей історичних пам'яток відносно отриманої області виведеної на карту, підвищення швидкодії за рахунок конвертації розроблених «важких» 3D моделей будівель в json-формат, а також збільшення зручності в користуванні шляхом створення клієнтоорієнтованого web-ресурсу.

Призначення роботи — розробка методу та програмного засобу для VR 3D відображення історичних пам'яток.

## **3 Вихідні дані для проведення НДР**

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

- 1) Документація до Three.js. URL: <https://threejs.org/docs/index.html#manual/introduction/Creating-a-scene>. (Дата доступу: 7.08.2019).
- 2) Tony Parisi. Learning Virtual Reality / Tony Parisi // O'Reilly Media, Inc. – 2015. – First Edition. – С. 75-97.

3) Електронні інформаційні ресурси: створення, використання, доступ: Збірник матеріалів Міжнародної науково-практичної Інтернет конференції. – Вінниця : ВНТУ, 2019. – 292 с.

4) Google Arts & Culture URL: <https://artsandculture.google.com/> (Дата звернення: 01.10.2019).

5) Дополненная, виртуальная и прочие реальности. URL: <https://www.it.ua/knowledge-base/technology-innovation/dopolnennaja-virtualnaja-i-prochie-realnosti> (Дата звернення: 06.09.2019)

#### **4. Технічні вимоги**

Операційна система — Linux; середовище розробки — Visio Studio Code; мови програмування — JavaScript, NodeJS.

Таким чином, розроблена система зовсім невибаглива і може бути використана навіть на найпростіших пристроях.

#### **5. Конструктивні вимоги.**

Конструкція пристрою повинна відповідати естетичним та ергономічним вимогам, повинна бути зручною в обслуговуванні та керуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

#### **6. Перелік технічної документації, що пред'являється по закінченню робіт:**

- пояснювальна записка до МКР;
- технічне завдання;
- лістинги програми.

#### **8. Вимоги до рівня уніфікації та стандартизації**

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.



### 9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз сучасного стану питання та постановка задач дослідження	
2	Розробка методу відображення 3D моделей у веб-додатку на основі отриманої із соцмережі геолокації	
3	Опис реалізації рішення та тестування розробленого програмного ресурсу	
4	Економічна частина	

### 10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

### Додаток Б. Акт впровадження

УЗГОДЖЕНО _____ Директор ТОВ “Екзістек” _____ Феферман О.Д. _____ « ____ » _____ 20__ року	ЗАТВЕРДЖУЮ _____ Завідувач кафедри ПЗ _____ д.т.н., професор Романюк О. Н. _____ « ____ » _____ 20__ року
--	--

АКТ ВПРОВАДЖЕННЯ № \_\_\_\_\_  
 результатів науково-дослідних робіт  
 Замовник ТОВ «Екзістек»  
 (найменування організації)

Цим актом підтверджується, що результати роботи – «Розробка методу та програмного засобу для VR 3D відображення історичних пам'яток»  
 (найменування теми)

що виконав студент гр. 2ПІ – 18м ВНТУ Гончарук Д.В.  
 (виконавці)

За договором про творчу співдружність без взаємних грошових розрахунків, на громадських засадах відповідно до теми затвердженої наказом № \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

виконана з \_\_\_\_\_ по \_\_\_\_\_  
 (строки виконання)

впроваджено у ТОВ «Екзістек»  
 (найменування організації, де здійснювалося впровадження)

1. Вид впроваджених результатів: експлуатація програмного забезпечення  
(експлуатація виробу, роботи, технології)
2. Характеристика масштабу впровадження: одиничне  
(унікальне, одиничне, партія, масове, серійне)
3. Форма впровадження: дослідний зразок
4. Новизна результатів науково-дослідної роботи якісно нові  
(піонерські, принципово нові, якісно нові, модифікації, модернізація старих розробок)
5. Впроваджені: для незалежної реалізації продукту на різних ринках.
6. Соціальний та науково-технічний ефект: спеціальне призначення  
(охорона навколишнього середовища, поліпшення й оздоровлення умов праці, удосконалення структури керування, науково-технічних напрямків, спеціальне призначення)

Від виконавця:  
 Студент групи 2ПІ-18м  
 \_\_\_\_\_ Гончарук Д.В.

Від ТОВ «Екзістек»  
 Директор  
 \_\_\_\_\_ Феферман О.Д.

Керівник: д.т.н., доцент кафедри \_\_\_\_\_ Черноволик  
 Г.О.



## Додаток В. Лістинг

### Лістинг файлу App.js

```
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');

var PORT = 8080;
var HOST_NAME = 'localhost';
var DATABASE_NAME = 'cityvr';

//Database initialize
var mongoose = require('mongoose');
mongoose.connect('mongodb://' + HOST_NAME + '/' + DATABASE_NAME);

var index = require('./routes/index');
var user = require('./routes/user');
var itemRouter = require('./routes/build');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

// database engine setup
// uncomment after placing your favicon in /public
//app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', index);
app.use('/user', user);
app.use('/api', itemRouter);

// catch 404 and forward to error handler
app.use(function (req, res, next) {
  var err = new Error('Not Found');
  err.status = 404;
  next(err);
});

// error handler
app.use(function (err, req, res, next) {
  // set locals, only providing error in development
```

```

res.locals.message = err.message;
res.locals.error = req.app.get('env') === 'development' ? err : {};

// render the error page
res.status(err.status || 500);
res.render('error');
});

module.exports = app;

```

## Лістинг файлу www (запуск сервера)

```

#!/usr/bin/env node

/**
 * Module dependencies.
 */

var app = require('./app');
var debug = require('debug')('test-myserver:server');
var http = require('http');

/**
 * Get port from environment and store in Express.
 */

var port = normalizePort(process.env.PORT || '80');
app.set('port', port);

/**
 * Create HTTP server.
 */

var server = http.createServer(app);

/**
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);

/**
 * Normalize a port into a number, string, or false.
 */

function normalizePort(val) {
  var port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;

```

```

}

if (port >= 0) {
  // port number
  return port;
}

return false;
}

/**
 * Event listener for HTTP server "error" event.
 */

function onError(error) {
  if (error.syscall !== 'listen') {
    throw error;
  }

  var bind = typeof port === 'string'
    ? 'Pipe ' + port
    : 'Port ' + port;

  // handle specific listen errors with friendly messages
  switch (error.code) {
    case 'EACCES':
      console.error(bind + ' requires elevated privileges');
      process.exit(1);
      break;
    case 'EADDRINUSE':
      console.error(bind + ' is already in use');
      process.exit(1);
      break;
    default:
      throw error;
  }
}

/**
 * Event listener for HTTP server "listening" event.
 */

function onListening() {
  var addr = server.address();
  var bind = typeof addr === 'string'
    ? 'pipe ' + addr
    : 'port ' + addr.port;
  debug('Listening on ' + bind);
}

```

## Лістинг файлу DeviceOrientationControl.js

```

/* globals THREE */

```

```

/**
 * DeviceOrientationControls - applies device orientation on object rotation
 *
 * @param {Object} object - instance of THREE.Object3D
 * @constructor
 *
 * @author richt / http://richt.me
 * @author WestLangley / http://github.com/WestLangley
 * @author jonobr1 / http://jonobr1.com
 * @author arodic / http://aleksandarrodic.com
 * @author doug / http://github.com/doug
 *
 * W3C Device Orientation control
 * (http://w3c.github.io/deviceorientation/spec-source-orientation.html)
 */

(function() {

    var deviceOrientation = {};
    var screenOrientation = window.orientation || 0;

    function onDeviceOrientationChangeEvent(evt) {
        deviceOrientation = evt;
    }
    window.addEventListener('deviceorientation', onDeviceOrientationChangeEvent, false);

    function getOrientation() {
        switch (window.screen.orientation || window.screen.mozOrientation) {
            case 'landscape-primary':
                return 90;
            case 'landscape-secondary':
                return -90;
            case 'portrait-secondary':
                return 180;
            case 'portrait-primary':
                return 0;
        }
        // this returns 90 if width is greater then height
        // and window orientation is undefined OR 0
        // if (!window.orientation && window.innerWidth > window.innerHeight)
        // return 90;
        return window.orientation || 0;
    }

    function onScreenOrientationChangeEvent() {
        screenOrientation = getOrientation();
    }
    window.addEventListener('orientationchange', onScreenOrientationChangeEvent, false);

    THREE.DeviceOrientationControls = function(object) {

        this.object = object;

        this.object.rotation.reorder('YXZ');
    }

```

```

this.freeze = true;

this.movementSpeed = 1.0;
this.rollSpeed = 0.005;
this.autoAlign = true;
this.autoForward = false;

this.alpha = 0;
this.beta = 0;
this.gamma = 0;
this.orient = 0;

this.alignQuaternion = new THREE.Quaternion();
this.orientationQuaternion = new THREE.Quaternion();

var quaternion = new THREE.Quaternion();
var quaternionLerp = new THREE.Quaternion();

var tempVector3 = new THREE.Vector3();
var tempMatrix4 = new THREE.Matrix4();
var tempEuler = new THREE.Euler(0, 0, 0, 'XYZ');
var tempQuaternion = new THREE.Quaternion();

var zee = new THREE.Vector3(0, 0, 1);
var up = new THREE.Vector3(0, 1, 0);
var v0 = new THREE.Vector3(0, 0, 0);
var euler = new THREE.Euler();
var q0 = new THREE.Quaternion(); // - PI/2 around the x-axis
var q1 = new THREE.Quaternion(- Math.sqrt(0.5), 0, 0, Math.sqrt(0.5));

this.update = (function(delta) {

    return function(delta) {

        if (this.freeze) return;

        // should not need this
        //var orientation = getOrientation();
        //if (orientation !== this.screenOrientation) {
        //this.screenOrientation = orientation;
        //this.autoAlign = true;
        //}

        this.alpha = deviceOrientation.gamma ?
            THREE.Math.degToRad(deviceOrientation.alpha) : 0; // Z
        this.beta = deviceOrientation.beta ?
            THREE.Math.degToRad(deviceOrientation.beta) : 0; // X'
        this.gamma = deviceOrientation.gamma ?
            THREE.Math.degToRad(deviceOrientation.gamma) : 0; // Y''
        this.orient = screenOrientation ?
            THREE.Math.degToRad(screenOrientation) : 0; // O

        // The angles alpha, beta and gamma

```



```

// form a set of intrinsic Tait-Bryan angles of type Z-X'-Y''

// 'ZXY' for the device, but 'YXZ' for us
euler.set(this.beta, this.alpha, - this.gamma, 'YXZ');

quaternion.setFromEuler(euler);
quaternionLerp.slerp(quaternion, 0.5); // interpolate

// orient the device
if (this.autoAlign) this.orientationQuaternion.copy(quaternion); // interpolation breaks the
auto alignment
else this.orientationQuaternion.copy(quaternionLerp);

// camera looks out the back of the device, not the top
this.orientationQuaternion.multiply(q1);

// adjust for screen orientation
this.orientationQuaternion.multiply(q0.setFromAxisAngle(zee, - this.orient));

this.object.quaternion.copy(this.alignQuaternion);
this.object.quaternion.multiply(this.orientationQuaternion);

if (this.autoForward) {

    tempVector3
        .set(0, 0, -1)
        .applyQuaternion(this.object.quaternion, 'ZXY')
        .setLength(this.movementSpeed / 50); // TODO: why 50 :S

    this.object.position.add(tempVector3);

}

if (this.autoAlign && this.alpha !== 0) {

    this.autoAlign = false;

    this.align();

}

};

})();

// //debug
// window.addEventListener('click', (function(){
//   this.align();
// }).bind(this));

this.align = function() {

    tempVector3
        .set(0, 0, -1)

```

```

        .applyQuaternion( tempQuaternion.copy(this.orientationQuaternion).inverse(), 'ZXY' );

        tempEuler.setFromQuaternion(
            tempQuaternion.setFromRotationMatrix(
                tempMatrix4.lookAt(tempVector3, v0, up)
            )
        );

        tempEuler.set(0, tempEuler.y, 0);
        this.alignQuaternion.setFromEuler(tempEuler);

    };

    this.connect = function() {
        this.freeze = false;
    };

    this.disconnect = function() {
        this.freze = true;
    };

};

})();

```

### Лістинг коду mainScene.js

```

function Three (_obj,_container) {
    var self = this;
    var scene, renderer, camera, model, controls, effect;
    var camPosition = {
        x: 10,
        y: 8,
        z: 0
    };
    var rotateX = 0;
    var rotateY = 0;
    var loader = new THREE.JSONLoader();
    if(_obj && typeof _obj === 'object') {
        model = loader.parse(_obj);
    }

    function render() {
        requestAnimationFrame(render);
        effect.render(scene, camera);
    }

    self.update = function(dt) {
        self.resize();
        camera.updateProjectionMatrix();
        controls.update(dt);
    };
}

```

```

self.resize = function () {
    var width = _container.offsetWidth;
    var height = _container.offsetHeight;

    camera.aspect = width / height;
    camera.updateProjectionMatrix();

    renderer.setSize(width, height);
    effect.setSize(width, height);
};

self.init = function () {
    try { // try to create a WebGLRenderer
        if (window.WebGLRenderingContext) {
            renderer = new THREE.WebGLRenderer( {
                canvas: _container,
                antialias: true
            });
        }
    }
    catch (e) {
    }
    if (!renderer) { // If the WebGLRenderer couldn't be created, try a CanvasRenderer.
        renderer = new THREE.CanvasRenderer( { canvas: _container } );
        renderer.setSize( _container.offsetWidth , _container.offsetHeight);
    }

    scene = new THREE.Scene();
    camera = new THREE.PerspectiveCamera(90, _container.offsetWidth / _container.offsetHeight,
0.001, 700);
    camera.position.set(camPosition.x, camPosition.y, camPosition.z);
    // camera.rotation.x = -Math.PI / 2;
    camera.scale.addScalar(1);

    effect = new THREE.StereoEffect(renderer);
    effect.setSize(window.innerWidth, window.innerHeight);

    controls = new THREE.OrbitControls(camera, renderer.domElement);
    controls.target.set(
        camera.position.x + 0.15,
        camera.position.y,
        camera.position.z
    );
    controls.noPan = true;
    controls.noZoom = true;
    createWorld();
    modelLoadedCallback( model.geometry, model.materials);

    window.addEventListener('deviceorientation', self.setOrientationControls, true);
    render();
};

self.render = function () {
    render();
}

```

```

};

function fullscreen() {
  if (_container.requestFullscreen) {
    _container.requestFullscreen();
  } else if (_container.msRequestFullscreen) {
    _container.msRequestFullscreen();
  } else if (_container.mozRequestFullScreen) {
    _container.mozRequestFullScreen();
  } else if (_container.webkitRequestFullscreen) {
    _container.webkitRequestFullscreen();
  }
}

function createWorld() {
  var light; // A light shining from the direction of the camera.
  light = new THREE.DirectionalLight();
  light.position.set(0,500,100);

  var lightP = new THREE.PointLight(0x999999, 2, 100);
  light.position.set(50, 50, 50);

  var lightScene = new THREE.PointLight(0x999999, 2, 100);
  lightScene.position.set(0, 5, 0);

  var ground = new THREE.Mesh(
    new THREE.PlaneGeometry(100, 100, 2, 2),
    new THREE.MeshLambertMaterial({color: '#ffcaa9'}));
  ground.rotation.set(-90 * (3.14/180), 0, 0, 'XYZ');

  // scene.add(light);
  scene.add(lightP);
  scene.add(lightScene);
  scene.add(ground);
}

function modelLoadedCallback(geometry, materials) {

  /* create the object from the geometry and materials that were loaded. There
  can be multiple materials, which can be applied to the object using MeshFaceMaterials.
  Note tha the material can include references to texture images might finish
  loading later. */

  var object = new THREE.Mesh(geometry, new THREE.MeshFaceMaterial(materials));

  /* Determine the ranges of x, y, and z in the vertices of the geometry. */

  var xmin = Infinity;
  var xmax = -Infinity;
  var ymin = Infinity;
  var ymax = -Infinity;
  var zmin = Infinity;
  var zmax = -Infinity;
  for (var i = 0; i < geometry.vertices.length; i++) {
    var v = geometry.vertices[i];

```

```

    if (v.x < xmin)
        xmin = v.x;
    else if (v.x > xmax)
        xmax = v.x;
    if (v.y < ymin)
        ymin = v.y;
    else if (v.y > ymax)
        ymax = v.y;
    if (v.z < zmin)
        zmin = v.z;
    else if (v.z > zmax)
        zmax = v.z;
}

/* translate the center of the object to the origin */
var centerX = (xmin+xmax)/2;
var centerY = (ymin+ymax)/2;
var centerZ = (zmin+zmax)/2;
var max = Math.max(centerX - xmin, xmax - centerX);
max = Math.max(max, Math.max(centerY - ymin, ymax - centerY) );
max = Math.max(max, Math.max(centerZ - zmin, zmax - centerZ) );
var scale = 10/max;
object.position.set( 0, 0, 0 );
object.rotation.x = 0;
object.rotation.y = -180;
console.log("Loading finished, scaling object by " + scale);
console.log("Center at ( " + centerX + ", " + centerY + ", " + centerZ + " )");

/* Create the wrapper, model, to scale and rotate the object. */

model = new THREE.Object3D();
model.add(object);
model.scale.set(scale,scale,scale);
rotateX = rotateY = 0;
scene.add(model);
render();
}

self.setOrientationControls = function(e) {
    if (!e.alpha) {
        return;
    }
}

controls = new THREE.DeviceOrientationControls(camera, true);
controls.connect();
controls.update();
renderer.domElement.addEventListener('click', fullscreen, false);
// window.removeEventListener('deviceorientation', self.setOrientationControls, true);
}
}

document.addEventListener("DOMContentLoaded",function () {

});

```

**Додаток Г. Ілюстративний матеріал****ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ЗАХИСТУ МАГІСТЕРСЬКОЇ  
КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Завідувач кафедри ПЗ, д. т. н., професор \_\_\_\_\_ О. Н. Романюк

Науковий керівник, к. т. н., доцент кафедри ПЗ \_\_\_\_\_ Г. О. Черноволик

Рецензент, к. т. н., доцент кафедри КН \_\_\_\_\_ І. Р. Арсенюк

Нормоконтроль, к. т. н., доцент кафедри ПЗ \_\_\_\_\_ Г. О. Черноволик

Виконавець, студент групи 2ПІ-18м \_\_\_\_\_ Д. В. Гончарук

## Слайд 1 — Титульний аркуш



## Розробка методу та програмного засобу для VR 3D відображення історичних пам'яток

Виконав: ст. II курсу, гр. 2ПІ-18м Гончарук Д.В.  
Керівник: к.т.н., доцент Черноволик Г.О.

спеціальність 121 — «Інженерія програмного забезпечення»

## Слайд 2 — Об'єкт, предмет та мета дослідження

**Об'єкт дослідження** — процес побудови та відображення 3D-моделей історичних пам'яток із застосуванням технології віртуальної реальності на основі отриманих координат для створення веб-ресурсу.

**Предмет дослідження** — методи моніторингу та отримання даних про останню відзначену геолокацію, отриману через мережу Instagram.

**Метою роботи є** покращення точності визначення останньої відзначеної користувачем геолокації шляхом отримання відповідної інформації з його Instagram-акаунту, для подальшого відображення користувачеві 3D-моделей історичних пам'яток відносно отриманої області виведеної на карту, підвищення швидкодії за рахунок конвертації розроблених «важких» 3D моделей будівель в json-формат, а також збільшення зручності в користуванні шляхом створення клієнтоорієнтованого web-ресурсу.

## Слайд 3 — Задачі дослідження



## Задачі дослідження

визначити сутність та значення сучасних методів представлення 3D об'єктів, їх переваги та недоліки

проаналізувати наявні методи та технології моніторингу геоданих користувачів

розробити метод моніторингу та отримання даних про геолокацію із соцмережі для відображення 3D моделей у веб-додатку

розробити методику конвертації 3D моделей

спроєкувати концептуальну, логічну та фізичну моделі веб-застосунку й описати принцип роботи розроблених засобів системи

спроєкувати базу даних, що міститиме 3D-моделі історичних пам'яток

розробити інструкцію користувача із підключення та використання ресурсу

## Слайд 4 — Наукова новизна



## Наукова новизна

Подальшого розвитку отримав метод моніторингу та отримання даних про останню відзначену геолокацію, який відрізняється від існуючих використанням географічних координат, отриманих через мережу Instagram, що дозволяє розподіляти на карті усі доступні у визначеному радіусі 3D-об'єкти з оптимальною швидкістю.

Розроблено методику конвертації створених «важких» 3D моделей будівель, яка відрізняється від існуючих застосуванням json-формату, що дозволяє підвищити продуктивність веб-ресурсу.

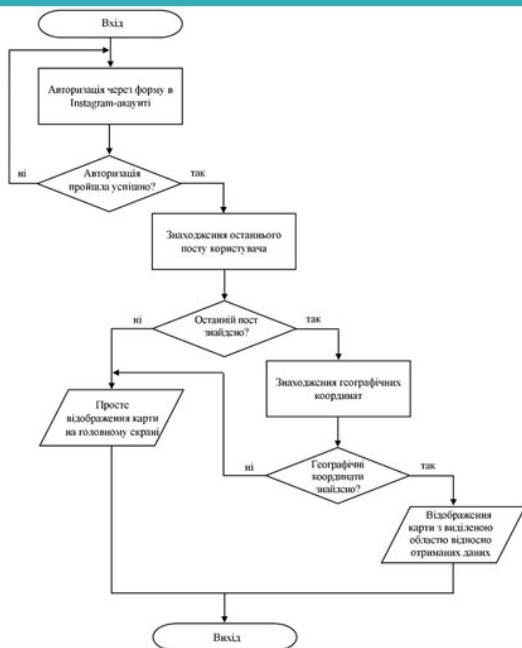
### Слайд 5 — Аналоги

#### Порівняльна таблиця розробленого ресурсу з аналогом

№	Критерії	Мій web-сервіс	Google Arts & Culture
1	Інформаційність	-	+
2	Якість 3D відображення	+	+
3	Швидкодія	+	-
4	Продуктивність	+	-
5	Підтримка	+	+
6	Зручність	+	+
7	Простота до використання	+	-
8	Геолокація	+	+



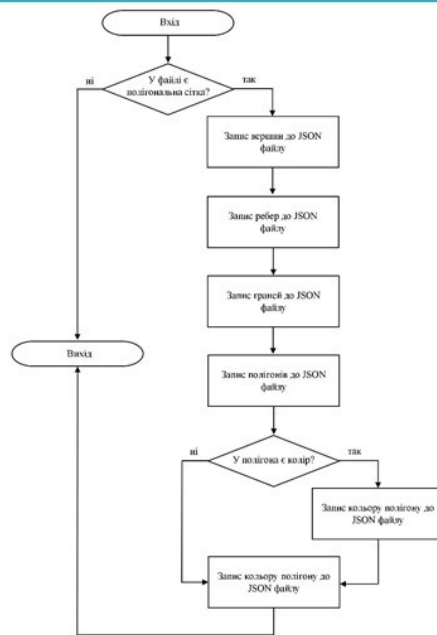
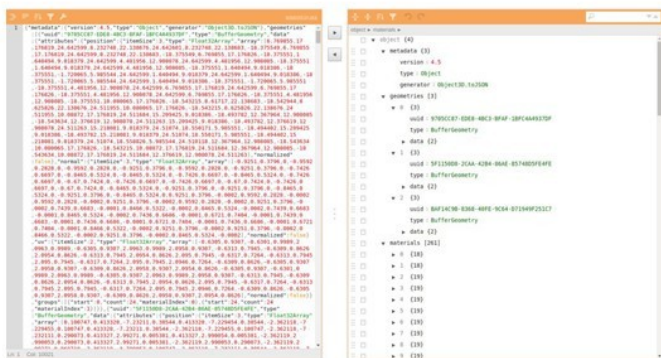
### Слайд 6 — Алгоритм отримання даних про останню відзначену геолокацію



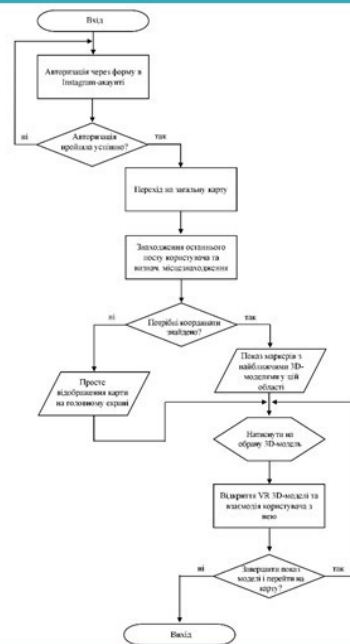
Алгоритм отримання даних про останню відзначену геолокацію

### Слайд 7 — Алгоритм конвертації створених 3D моделей

### Алгоритм конвертації створених «важких» 3D моделей будівель

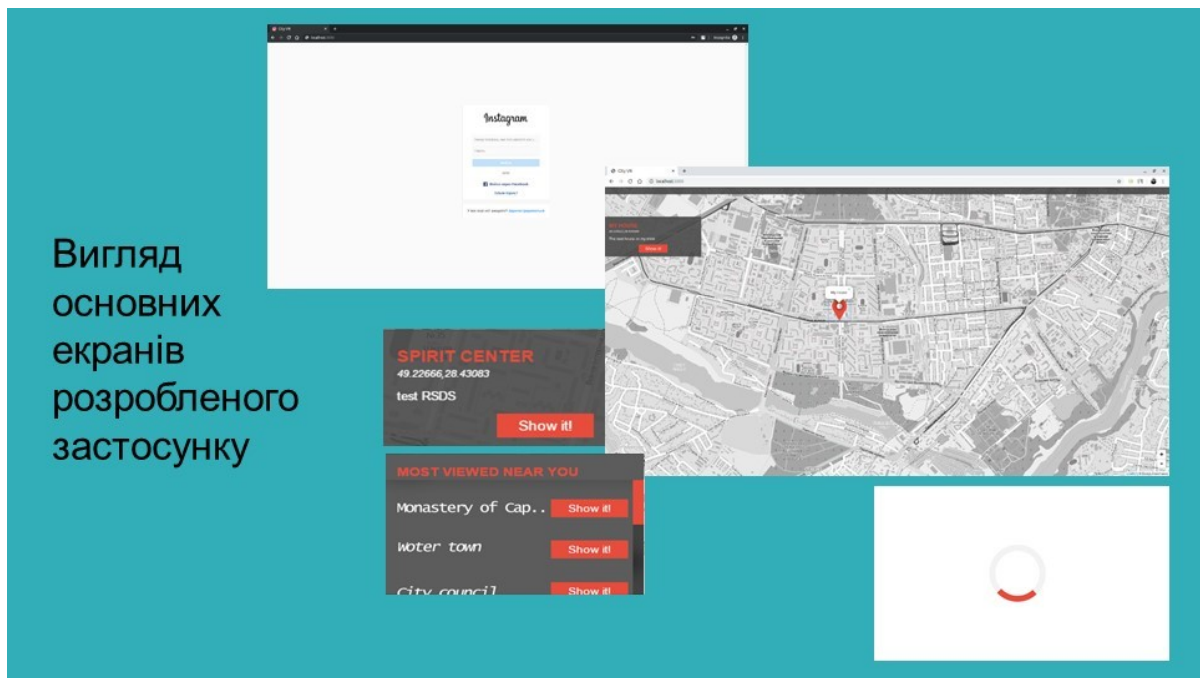


## Слайд 8 — Алгоритм взаємодії користувача з ресурсом



Алгоритм взаємодії  
користувача з ресурсом

## Слайд 9 — Аналоги



## Слайд 10 — Результат роботи розробленого застосунку

## Результат роботи розробленого застосунку



### Слайд 11 — Економічне обґрунтування

## Економічне обґрунтування

Рівень комерційного потенціалу розробки вище середнього

Розроблювання даної системи є актуальною та доцільною з точки зору економічних розрахунків

Відносна (щорічна) ефективність вкладених в наукову розробку інвестицій  $E_v$  становить 162%

Термін окупності даної наукової розробки складає 0,43 роки

*Отже, фінансування даної наукової розробки буде економічно доцільним.*

## Слайд 12 — Акт впровадження

**ДОДАТОК Б Акт впровадження**

<p><b>УЗГОДЖЕНО</b></p> <p>Директор ТОВ «Екзістек» _____</p> <p>Феферман О.Д. _____</p> <p>« 2 » 12 20__ року</p>	<p><b>ЗАТВЕРДЖУЮ</b></p> <p>Завідувач кафедри ІЗ _____</p> <p>д.т.н., професор Романюк О. Н. _____</p> <p>« » 20__ року</p>
---	---

**АКТ ВПРОВАДЖЕННЯ № \_\_\_\_\_**

результатів науково-дослідних робіт

Замовник ТОВ «Екзістек» \_\_\_\_\_  
(найменування організації)

Цим актом підтверджується, що результати роботи – «Розробка методу та програмного засобу для VR 3D відображення історичних пам'яток» (найменування теми) що виконав студент гр. ЗПІ – 18м ВНТУ Гончарук Д.В. (виконавець)

За договором про творчу співдружність без взаємних грошових розрахунків, на громадських засадах відповідно до теми затвердженої наказом № \_\_\_\_\_ від «\_\_» \_\_\_\_\_ 20\_\_ р. виконана з \_\_\_\_\_ по \_\_\_\_\_ (строки виконання)

впроваджено у ТОВ «Екзістек» \_\_\_\_\_ (найменування організації, де здійснювалося впровадження)

1. Вид впроваджених результатів: експлуатація програмного забезпечення (експлуатація виробу, роботи, технології)
2. Характеристика масштабу впровадження: одиначне (унікальне, одиначне, партія, масове, серійне)
3. Форма впровадження: дослідний зразок
4. Новизна результатів науково-дослідної роботи: якісно нові (пошукові, прикладно нові, експлоат. нові, модифікації, модернізації старих розробок)
5. Впроваджені: для незалежної реалізації продукту на різних ринках.
6. Соціальний та науково-технічний ефект: спеціальне призначення (охорона навколишнього середовища, поліпшення й одороження умов праці, удосконалення структури керування, науково-технічних напрямків, спеціальне призначення)

<p>Від виконавця: Студент групи ЗПІ-18м _____ Гончарук Д.В.</p> <p>Керівник: к.т.н., доцент кафедри _____</p>	<p>Від ТОВ «Екзістек» Директор _____ Феферман О.Д.</p> <p>Черноволук Г.О.</p>
---	---

## Впровадження

Впровадження результатів досліджень підтверджуються відповідними актами та використовуються на такому підприємстві: компанія «ТОВ Екзістек» для забезпечення працівників організації ресурсом, що гарантуватиме цікаве дозвілля, культурний розвиток та ознайомлення з архітектурними пам'ятками рідного міста, для можливості удосконалення існуючих та створення нових методів та втілення ідей компанії для незалежної реалізації продукту на різних ринках.

***В процесі розробки:***

- визначено сутність та значення сучасних методів представлення 3D об'єктів, їх переваги та недоліки;
- проаналізовано наявні методи та технології моніторингу геоданих користувачів;
- розроблено метод моніторингу та отримання даних про геолокацію із соцмережі для відображення 3D моделей у веб-додатку;
- спроектовано концептуальну, логічну та фізичну моделі веб-ресурсу й описано принцип роботи розроблених засобів системи;
- спроектовано базу даних, що міститиме 3D-моделі історичних пам'яток
- розроблено інструкцію користувача із підключення та використання ресурсу.

Отже, було розроблено метод та програмний засіб для VR 3D відображення історичних пам'яток.

**Висновки**