Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
бакалавр
(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ МОНІТОРИНГУ ТА ПОШУКУ МОБІЛЬНИХ ПРИСТРОЇВ
SOFTWARE IMPLEMENTATION OF THE APPLICATION FOR MONITORING AND SEARCH OF MOBILE DEVICES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-121сп
Спеціальності 121 Інженерія програмного забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

ГАЛЧАНСЬКИЙ В.Д.
(ПРІЗВИЩЕ та ініціали)
Керівник ЛЕОЩЕНКО С.Д.
(ПРІЗВИЩЕ та ініціали)
Рецензент СКРУПСЬКИЙ С.Ю.
(ПРІЗВИЩЕ та ініціали)

2024

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
_____С.О. СУББОТІН
«____» _____2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГАЛЧАНСЬКОГО Віталія Дмитровича
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація застосунку моніторингу та пошуку мобільних пристроїв. Software Implementation of the Application for Monitoring and Search of Mobile Devices

керівник проєкту (роботи) д-р філософ., Леощенко Сергій Дмитрович ,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від " 24 " квітня 2024 року № 168
2. Строк подання студентом проєкту (роботи) 07 червня 2024 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЛЕОЩЕНКО С.Д., ст.викл.		
Нормоконтроль	ЛИПОВЕЦЬ М.В., асистент		

7. Дата видачі завдання « 24 » квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки.		
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження	5 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	6 тиждень	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка) _____ Віталій ГАЛЧАНСЬКИЙ
(підпис) (Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи) _____ Сергій ЛЕОЩЕНКО
(підпис) (Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 99 с., 3 табл., 25 рис., 4 дод., 24 джерела.

РОДИНА, ПОШУК, ЗАХИСТ, БЕЗПЕКА, МОБІЛЬНИЙ ПРИСТРІЙ, КРОСПЛАТФОРМНИЙ ЗАСТОСУНОК, РОЗРОБКА, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об’єкт дослідження – процес програмної реалізації застосунку моніторингу та пошуку мобільних пристроїв.

Предмет дослідження – програмні засоби для програмної реалізації застосунку моніторингу та пошуку мобільних пристроїв.

Метою роботи є програмна реалізація застосунку для застосунку моніторингу та пошуку мобільних пристроїв.

Матеріали, методи та технічні засоби: мова програмування Python, вебфреймворк Django, фреймворк Flask, система керування базами даних MySQL, середовище розробки Atom та засоби моделювання UML.

Результати. Розроблено застосунок – моніторингу та пошуку мобільних пристроїв.

Практична цінність роботи полягає у програмній реалізації застосунку моніторингу та пошуку мобільних пристроїв у вигляді кросплатформного застосунку.

Висновки. Реалізовано застосунок моніторингу та пошуку мобільних пристроїв у вигляді кросплатформного застосунку.

Галузь використання – батьки, діти, інші члени родини.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 99 pages, 3 tables, 25 figures, 4 appendixes, 24 sources.

FAMILY, SEARCH, PROTECTION, SECURITY, MOBILE DEVICE, CROSS-PLATFORM APPLICATION, DEVELOPMENT, PROGRAMMING LANGUAGE, SOFTWARE.

The object of research is the process of software implementation of the mobile device monitoring and search application.

The subject of the research is software tools for the software implementation of the mobile device monitoring and search application.

The aim of this work is to provide a software implementation of an application for monitoring and searching mobile devices.

Materials, methods, and technical tools: Python programming language, Django web framework, Flask framework, MySQL database management system, Atom development environment, and UML modeling tools.

Results. An application for monitoring and searching mobile devices has been developed.

The practical value of the work lies in the software implementation of the mobile device monitoring and search application in the form of a cross-platform application.

Conclusions. Implemented an application for monitoring and searching mobile devices in the form of a cross-platform application.

Scope of use – parents, children, and other family members.

ЗМІСТ

С.

Перелік скорочень та умовних познак 8

Вступ 9

1 Аналіз предметної області 14

1.1 Застосунки родинні локатори або ж програмне забезпечення моніторингу та пошуку мобільних пристроїв 15

1.2 Огляд існуючих програмних рішень 19

1.2.1 Застосунок Life360 23

1.2.2 Програмний комплекс Google Family Link 26

1.2.3 Застосунок Family GPS Tracker Desktop 27

1.2.4 Дослідження та аналіз результатів порівняння 30

1.3 Постановка завдання 34

1.3.1 Функціональність системи 34

1.3.2 Вимоги до інтерфейсу 35

1.3.3 Вимоги до безпеки 35

1.3.4 Не функціональні вимоги до продукту 35

1.4 Висновки за розділом 1 36

2 Матеріали і методи 37

2.1 Вибір мови програмування 37

2.2 Вибір середовища розробки 45

2.3 Структурна схема розробки 53

2.4 Висновки за розділом 2 54

3 Основні рішення щодо реалізації компонентів системи 55

3.1 Проєктування моделей системи 55

3.2 Проєктування інтерфейсу 60

4 Експлуатація, тестування та експериментальне дослідження програми 66

4.1 Призначення й умови застосування програми 66

4.2 Вимоги до інтерфейсу 66

4.3 Умови виконання програми 67
4.4 Методика та результати тестування 67
Висновки 70
Перелік джерел посилань 72
Додаток А Технічне завдання 75
Додаток Б Опис програми 79
Додаток В Текст програми 83
Додаток Д Слайди презентації 93

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

IDE – Integrated Development Environment;

IoT – Internet of Things;

GPS – Global Positioning System;

ORM – Object Relational Mapping;

UML – Unified Modeling Language;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПЗМтаПМП – програмне забезпечення моніторингу та пошуку мобільних пристроїв;

СКБД – система керування базами даних.

ВСТУП

Стрімкий розвиток технологій, особливо мобільних пристроїв, докорінно змінив наше повсякденне життя. Мобільні телефони, які колись були просто засобами зв'язку, перетворилися в незамінні багатофункціональні пристрої, глибоко інтегровані в особисту і професійну сфери. Однією з важливих розробок у цій галузі є програми родинного локатору, призначені для відстеження та визначення місцезнаходження мобільних пристроїв у сімейній мережі [1]–[3]. Актуальність розробки цих додатків обумовлена безліччю факторів, включаючи міркування безпеки, зміни в суспільстві, технічний прогрес і психологічну користь [1]–[3].

Однією з основних причин розробки застосунків моніторингу та пошуку мобільних пристроїв є безпека дітей. Батьки все більше турбуються про безпеку своїх дітей у зв'язку з випадками викрадення дітей, їх втрати або потрапляння в небезпечні ситуації. Програми моніторингу та пошуку мобільних пристроїв забезпечують відстеження в режимі реального часу, дозволяючи батькам відстежувати місцезнаходження своїх дітей і отримувати негайні сповіщення, якщо вони відправляються в небезпечні райони [1]–[3].

Крім того, ці програми незамінні для спостереження за літніми членами сім'ї, особливо тими, хто страждає такими захворюваннями, як деменція або хвороба Альцгеймера. Такі люди схильні йти з дому і губитися, що створює значні ризики для безпеки. Програмне забезпечення моніторингу та пошуку мобільних пристроїв (ПЗМтаПМП) може допомогти опікунам забезпечити свою безпеку, надаючи точні дані про місцезнаходження в режимі реального часу та сповіщення про надзвичайні ситуації [1]–[3].

У надзвичайних ситуаціях, будь то стихійні лиха чи нещасні випадки, програми ПЗМтаПМП дозволяють швидко та ефективно відстежувати місцезнаходження. Ця функціональність може прискорити рятувальні операції та забезпечити своєчасну допомогу, потенційно рятуючи життя [1]–[3].

Сучасне суспільство характеризується високою мобільністю та урбанізацією. Сім'ї часто живуть у великих містах, де стежити один за одним буває непросто. При напруженому способі життя і розрізненому графіку роботи члени сім'ї можуть перебувати в різних місцях протягом дня. ПЗМтаПМП усувають цю прогалину, гарантуючи, що всі залишатимуться на зв'язку, незважаючи на фізичну відстань [1]–[3].

Структура сім'ї зазнала змін, з'явилося більше родин з одним батьком, сімей з подвійним доходом і змішаних сімей. Ці зміни вимагають інструментів, які можуть забезпечити впевненість і полегшити координацію між членами сім'ї. ПЗМтаПМП пропонують практичне рішення для управління складнощами сучасного сімейного життя, гарантуючи, що всі будуть в курсі подій [1]–[3].

У міру того як зростає обізнаність про технології забезпечення безпеки, виникає потреба в легкодоступних і надійних рішеннях. Поширення таких застосунків вказує на суспільний попит на інструменти, що підвищують особисту та сімейну безпеку. Задоволення цього попиту має вирішальне значення для розробників технологій, щоб залишатися актуальними і реагувати на потреби споживачів [1]–[3].

Досягнення технології GPS значно підвищили точність та надійність відстеження місцезнаходження. Сучасні смартфони оснащені складними GPS-чіпами, які надають точні дані про місцезнаходження, що робить застосунки ПЗМтаПМП більш ефективними і надійними [1]–[3].

Розвиток Інтернету речей дозволяє програмам ПЗМтаПМП інтегруватися з різними розумними пристроями, такими як розумні годинники, системи домашньої безпеки та транспортні засоби. Така інтеграція підвищує функціональність і зручність цих додатків, забезпечуючи комплексні рішення в області безпеки [1]–[3].

З розвитком надійних протоколів шифрування та безпеки питання конфіденційності, пов'язані з відстеженням місцезнаходження, вирішуються більш ефективно. Користувачі можуть бути впевнені в безпеці своїх даних, що

сприяє більш широкому впровадженню додатків для визначення місця розташування сім'ї [1]–[3].

Знання точного місцезнаходження близьких людей забезпечує величезний душевний спокій, зменшуючи тривогу та стрес. Батьки, зокрема, можуть відчувати себе спокійніше, знаючи, що їхні діти в безпеці, що може позитивно вплинути на їхнє загальне самопочуття та продуктивність [1]–[3].

Регулярні перевірки, проведені за допомогою ПЗМтаПМП, можуть зміцнити сімейні зв'язки. Можливість легко ділитися інформацією про місцезнаходження та оновленнями сприяє відчуттю близькості та взаємозв'язку, навіть коли члени сім'ї фізично знаходяться окремо [1]–[3].

Ці програми полегшують координацію сімейних заходів та розкладів. Наприклад, батьки можуть забезпечити своєчасну доставку дітей, а члени сім'ї можуть більш ефективно планувати зустрічі. Така простота координації підвищує згуртованість сім'ї і знижує навантаження на матеріально-технічне забезпечення [1]–[3].

Розробка ПЗМтаПМП також узгоджується з більш широкими правилами та ініціативами щодо безпеки. У багатьох регіонах діють закони, які вимагають посилення заходів безпеки для дітей та вразливих груп населення. Ці програми можуть допомогти сім'ям дотримуватися таких правил, сприяючи більш широким зусиллям щодо забезпечення громадської безпеки [1]–[3].

Незважаючи на те, що конфіденційність є серйозною проблемою, добре розроблені ПЗМтаПМП забезпечують суворий контроль конфіденційності, дозволяючи користувачам визначати, хто і коли може бачити їх місцезнаходження. Цей контроль спрямований на усунення можливих зловживань і відповідає нормативним стандартам захисту даних [1]–[3].

Ринок застосунків безпеки та локаторів швидко розширюється. Оскільки все більше людей усвідомлюють переваги цих інструментів, попит на них продовжує зростати. Розробка надійного програмного забезпечення для

сімейних локаторів може задовольнити цей ринковий попит, відкриваючи вигідні можливості для розробників та бізнесу [1]–[3].

Технологічним компаніям розробка сучасних додатків для визначення місця розташування сімейства локаторів може забезпечити конкурентну перевагу. Унікальні функції, чудова точність і надійний захист дозволяють виділяти продукт на насиченому ринку, залучаючи більше користувачів і підвищуючи доходи [1]–[3].

ПЗМтаПМП пропонують економічно ефективне рішення безпеки порівняно з такими альтернативами, як Послуги особистої охорони або спеціалізовані GPS-пристрої. Така доступність робить їх доступними для широкої аудиторії, підвищуючи загальну громадську безпеку [1]–[3].

Етичні міркування при розробці таких застосунків мають першорядне значення. Забезпечення відповідального використання застосунків з чіткої згоди і прозорості має вирішальне значення. Розробники повинні надавати пріоритет етичним стандартам для захисту конфіденційності користувачів та запобігання зловживанню [1]–[3].

Технологічні компанії несуть соціальну відповідальність за забезпечення громадської безпеки. Розробляючи програми для розташування сімей, вони можуть відігравати вирішальну роль у захисті окремих осіб та громад, узгоджуючи свої бізнес-цілі з більш широкими суспільними вигодами [1]–[3].

Розробка програм для розташування сім'ї є нагальним завданням, зумовленим багатьма факторами, включаючи підвищення особистої безпеки, зміни в суспільстві, технічний прогрес, психологічні переваги, юридичні наслідки, ринковий попит та етичні міркування [1]–[3]. Ці програми є безцінними інструментами для сучасних сімей, забезпечуючи душевний спокій, покращуючи координацію рухів і забезпечуючи безпеку близьких. У міру подальшого розвитку технологій потреба в надійних, безпечних і зручних у використанні додатках для визначення місця розташування сім'ї буде тільки

зростати, що робить їх розробку найважливішим пріоритетом для технологічних компаній і суспільства в цілому [1]–[3].

Отже актуальною задачею залишається програмна реалізація застосунку моніторингу та пошуку мобільних пристроїв.

Для виконання такої задачі необхідно виконати наступні завдання:

– дослідити та проаналізувати використання сімейних локаторів на мобільних пристроях;

– дослідити існуючі рішення для організації родинних локаторів (виявити їхні переваги та недоліки);

– сформувати набір функціональних вимог до застосунку моніторингу та пошуку мобільних пристроїв;

– спроєктувати та реалізувати архітектуру застосунку, який би реалізовував набір функціональних вимог;

– протестувати розробку;

– за результатами тестування сформувати висновки за виконаною роботою.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Цей розділ буде присвячено критичному аналізу взаємозв'язку між різними існуючими розробками за темою проєкту. Реалізації, виконані різними авторами, мають дійсно важливе значення для створення основи для досліджень і надають інформацію про те, що було зроблено іншими людьми для вирішення даної проблеми.

У 21 столітті, з розвитком технологій, сучасне суспільство, включаючи навіть маленьких дітей використовує найрізноманітніші гаджети, в тому числі смартфони. І в зв'язку з цим, зростає і ризик крадіжки мобільних телефонів, або вже тим більше крадіжки дітей. Багатьом батькам стає важко працювати і в той же час піклуватися про своїх малюків в денний час. Таким чином, крім наявності няні на повний робочий день, потрібно забезпечити дитині повноцінний догляд і контроль [1]–[3].

Ступінь крадіжки смартфонів дітей не тільки в громадських місцях, але і в затишних місцях просто показує, наскільки важливо мати системи спостереження за дитиною, щоб відстежувати кожною дитини. Це рішення призначене головним чином для родин, але може бути легко інтегровано і в інші середовища – лікарняні установи і дитячі садочки [1]–[3].

За ці роки технології значно просунулися. Для відстеження людей були створені різні рішення. Використання технології для відстеження дітей дасть мною переваг. За останні кілька років індустрія дитячого GPS-відстеження значно просунулася і є досить новим застосуванням інформаційних технологій. Деякі з цих систем GPS-відстеження можуть визначати частоту серцевих скорочень, артеріальний тиск і температуру тіла дитини [1]–[3]. Ця інформація може допомогти батькам дізнатися, чи відчуває їх дитина себе комфортно або у неї є почуття страху і дискомфорту, що вказує на те, що у нього можуть бути проблеми. Нові пристрої використовують GPS, Wi-Fi та інші технології відстеження місцеположення і можуть бути пов'язані з додатками на телефоні батьків. За допомогою технологій можна значно

спростити відстеження дітей і, отже, полегшити тягар батьків і опікунів, які про них піклуються [1]–[3].

1.1 Застосунки родинні локатори або ж програмне забезпечення моніторингу та пошуку мобільних пристроїв

Популярність ПЗМтаПМП, або ж застосунків, на кшталт, родинних локаторів зростає, вони стають незамінними інструментами для сучасних сімей. Ці програми використовують поєднання передових технологій для відстеження місця розташування в режимі реального часу та інші функції безпеки, забезпечуючи спокій і підвищену захищеність [4]–[6].

Для ефективної роботи ПЗМтаПМП використовуються кілька ключових технологій і методологій.

Глобальна система позиціонування (global positioning system – GPS) є наріжним каменем ПЗМтаПМП. Супутники GPS, що обертаються навколо Землі, передають сигнали, які пристрої з підтримкою GPS використовують для визначення свого точного розташування [4]–[6]. Ось як це працює:

– супутники GPS безперервно відправляють сигнали, що містять інформацію про їх місцезнаходження і точний час передачі сигналу;

– прийом сигналу – GPS-приймач в смартфоні або іншому пристрої приймає сигнали від декількох супутників [4]–[6];

– тріангуляція – приймач використовує затримку часу між надсиланням сигналу та його отриманням для обчислення відстані до кожного супутника. Тріангулюючи ці відстані принаймні з трьох супутників, приймач може точно визначити своє місце розташування на Землі [4]–[6].

У місцях, де сигнали GPS слабкі або недоступні, наприклад, у приміщеннях або в густонаселених районах, ПЗМтаПМП можуть використовувати тріангуляцію стільникової мережі. Цей метод дозволяє визначити місце розташування пристрою на основі його близькості до веж стільникового зв'язку [4]–[6]. Пристрій підтримує зв'язок з декількома вежами

стільникового зв'язку, і додаток використовує силу цих сигналів для оцінки свого місця розташування [4]–[6].

Визначення місця розташування по Wi-Fi – ще одна додаткова технологія. Вона особливо корисна в міських умовах, де мережі Wi-Fi широко поширені. Пристрій сканує сусідні мережі Wi-Fi та використовує базу даних відомих точок доступу Wi-Fi для визначення свого місцезнаходження [4]–[6]. Цей метод підвищує точність при недостатніх сигналах GPS і стільникового зв'язку.

Деякі вдосконалені програми для визначення місцезнаходження також використовують Bluetooth-маяки для розміщення в приміщенні. Ці маяки – це невеликі пристрої, які передають постійний сигнал Bluetooth. Коли смартфон потрапляє в зону дії, він виявляє маяк і використовує потужність сигналу для визначення близькості, забезпечуючи точне відстеження місця розташування всередині приміщень [4]–[6].

ПЗМтаПМП об'єднують дані GPS, стільникових мереж, Wi-Fi та Bluetooth, щоб забезпечити точне та надійне відстеження місцезнаходження. Передові алгоритми обробляють ці дані для підвищення точності, виправлення помилок і надання оновлень в режимі реального часу [4]–[6].

ПЗМтаПМП оснащені інтуїтивно зрозумілим інтерфейсом користувача, який відображає місцезнаходження членів сім'ї на карті в режимі реального часу. Користувачі можуть налаштувати сповіщення про геозонування, які сповіщатимуть їх, коли член сім'ї входить або залишає заздалегідь визначену зону. Інші функції можуть включати історію місцезнаходжень, екстрені сповіщення та засоби прямого зв'язку [4]–[6].

Популярність програм family locator можна пояснити кількома важливими факторами.

Головна турбота батьків – безпека їхніх дітей. ПЗМтаПМП дозволяють батькам відстежувати місцезнаходження своїх дітей, забезпечуючи їх безпеку, незалежно від того, перебувають вони в школі, вдома у друзів або в подорожі.

Відстеження та оповіщення в режимі реального часу забезпечують впевненість і душевний спокій [4]–[6].

Для сімей, які мають людей похилого віку, особливо тих, хто має когнітивні порушення, такі як деменція, програми локаторів є безцінними. Вони допомагають відстежувати рух людей похилого віку, гарантуючи, що вони не підуть і не загубляться, що є поширеною та небезпечною проблемою [4]–[6].

Ці програми також можуть підвищити особисту безпеку людей у потенційно небезпечних ситуаціях. Користувачі можуть швидко повідомляти про своє місцезнаходження надійним контактам у разі надзвичайних ситуацій, гарантуючи швидке прибуття допомоги.

Останні досягнення в технології GPS значно підвищили точність визначення місця розташування. Завдяки високоточним GPS-чіпам у сучасних смартфонах програми для визначення місцезнаходження сім'ї можуть забезпечити високу точність визначення місцезнаходження, підвищуючи їх надійність та ефективність [4]–[6].

Широке поширення смартфонів, оснащених передовими датчиками і можливостями підключення, зробило ПЗМтаПМП легко доступними для широких мас. Більшість людей вже мають необхідне обладнання, що полегшує розгортання та використання застосунків [4]–[6].

Завдяки поширенню високошвидкісного Інтернету і широкому охопленню стільниковою мережею, передача даних в режимі реального часу стає безперебійною. Таке підключення забезпечує своєчасну і точну інформацію про місцезнаходження, що підвищує ефективність застосунків [4]–[6].

Сучасний спосіб життя характеризується високою мобільністю, коли члени сім'ї часто перебувають у різних місцях протягом дня. ПЗМтаПМП допомагають подолати цей розрив, гарантуючи, що, незважаючи на фізичну відстань, сім'ї залишаються на зв'язку та інформовані про місцезнаходження один одного [4]–[6].

У міру того, як все більше людей живе в містах, складність навігації в жвавому і багатолюдному середовищі зростає. ПЗМтаПМП забезпечують високий рівень безпеки та координації, полегшуючи управління міським життям [4]–[6].

Зростання кількості неповних домогосподарств, сімей з подвійним доходом та змішаних сімей збільшило потребу в інструментах, що полегшують координацію та спілкування. ПЗМтаПМП задовольняють цю потребу, надаючи платформу для отримання оновлень та сповіщень у режимі реального часу [4]–[6].

Знання місцезнаходження близьких в режимі реального часу значно знижує тривогу і стрес. Батьки, зокрема, почуваються спокійніше, знаючи, що їхні діти в безпеці. Цей спокій підвищує загальне самопочуття та продуктивність [4]–[6].

Регулярні перевірки та оновлення, що проводяться за допомогою цих додатків, можуть зміцнити сімейні зв'язки. Можливість залишатися на зв'язку, ділитися місцем розташування та легко спілкуватися сприяє зміцненню почуття близькості та довіри в сім'ї.

ПЗМтаПМП спрощують логістику сімейного життя. Завдяки даним про місцезнаходження в режимі реального часу стає простіше координувати зустрічі, запобігаючи непорозумінням і зміцнюючи сімейну гармонію.

Програми розташування сім'ї часто є економічно вигідними, особливо в порівнянні з такими альтернативами, як спеціалізовані GPS-пристрої або служби особистої безпеки. Багато застосунків пропонують безкоштовні версії з основними функціями, що робить їх доступними для широкої аудиторії.

Забезпечуючи надійні засоби відстеження та координації, ці програми можуть заощадити сім'ям гроші. Наприклад, вони можуть зменшити потребу в додаткових послугах по догляду за дітьми, дозволяючи краще керувати рухами та діяльністю дітей [4]–[6].

Розробники впровадили суворий контроль за конфіденційністю, що дозволяє користувачам визначати, хто може отримати доступ до даних про їх

місцезнаходження. Такий етичний підхід вирішує проблеми конфіденційності та зміцнює довіру між користувачами.

Розробка та використання ПЗМтаПМП сприяють більш широкому забезпеченню громадської безпеки. Підвищуючи індивідуальну та сімейну безпеку, ці програми відіграють важливу роль у створенні більш безпечних спільнот [4]–[6].

Зростаючий попит на рішення безпеки підвищує популярність ПЗМтаПМП. У міру того, як зростає обізнаність про ці інструменти, все більше сімей впроваджують їх, що сприяє їх подальшому розвитку та вдосконаленню.

Технологічні компанії постійно впроваджують інновації, додаючи нові функції та покращуючи досвід користувачів. Інтеграція з розумними пристроями, покращений аналіз даних та зручні інтерфейси роблять ці програми актуальними та привабливими [4]–[6].

ПЗМтаПМП стали популярними завдяки своїй здатності забезпечувати підвищену безпеку, зручність і душевний спокій. Вони використовують передові технології, такі як GPS, стільникова тріангуляція та позиціонування Wi-Fi, щоб забезпечити точне та надійне відстеження місцезнаходження. Соціальні зміни, технічний прогрес та економічні фактори сприяють їх впровадженню. Оскільки ці програми продовжують розвиватися, вони, ймовірно, стануть ще більш невід'ємною частиною сучасного сімейного життя, гарантуючи, що сім'ї залишатимуться на зв'язку та в безпеці у все більш складному світі [4]–[6].

1.2 Огляд існуючих програмних рішень

ПЗМтаПМП оснащені різними функціями, призначеними для підвищення безпеки, підключення та зручності членів сім'ї. Ці функції використовують передові технології для відстеження, оповіщення та інших функцій у режимі реального часу [7]–[9].

Відстеження місцезнаходження в режимі реального часу. Відстеження в реальному часі забезпечує своєчасну інформацію про місцезнаходження членів сім'ї. Відображення карти – показує точне місце розташування кожного члена сім'ї на карті. Спеціальні зони – дозволяють користувачам встановлювати віртуальні межі навколо певних місць (наприклад, будинку, школи). Оповіщення – відправляють повідомлення, коли член сім'ї входить в зазначене місце або залишає його [7]–[9].

Історія місцезнаходжень. Відстеження минулих переміщень записує і відображає маршрути і місця розташування, відвідані членами сім'ї за певний період. Огляд і аналіз – допомагають користувачам аналізувати минулі переміщення з метою безпеки і планування [7]–[9].

Аварійні сповіщення та кнопка SOS. Миттєві сповіщення – дозволяють користувачам надсилати миттєві сповіщення всім членам сім'ї у разі надзвичайної ситуації. Можливість прямого контакту полегшує швидкий зв'язок з контактними особами або службами екстреної допомоги [7]–[9].

Моніторинг заряду батареї. Стан батареї іншого користувача допомагає, відображаючи поточний рівень заряду пристроїв членів сім'ї. Сповіщення про низький заряд батареї – повідомляє користувачів, коли в пристрої члена сім'ї розряджається батарея [7]–[9].

Обмін повідомленнями в застосунку. Прямий зв'язок – надає платформу для спілкування членів сім'ї всередині програми. Групові повідомлення дозволяють відправляти повідомлення всім членам сім'ї одночасно.

Оповіщення про перевищення швидкості. Безпека водіння – відстежує швидкість автомобіля члена сім'ї та надсилає сповіщення, якщо він перевищує встановлене обмеження швидкості. Моніторинг водіїв-підлітків – допомагає батькам стежити за звичками своїх водіїв-підлітків [7]–[9].

Інтеграція з інтелектуальними пристроями. Мобільні пристрої – підтримка інтеграції з розумними годинниками та фітнес-трекерами для отримання додаткових даних про місцезнаходження. Домашня автоматизація

– взаємодія з системами домашньої безпеки і пристроями «розумного будинку» [7]–[9].

Засоби контролю конфіденційності. Налаштування спільного доступу до місцезнаходження дозволяє користувачам контролювати, хто і коли може бачити їх місцезнаходження. Тимчасовий доступ – дозволяє тимчасово надавати доступ до місця розташування на певні періоди або події.

Хоча ПЗМтаПМП мають численні переваги, вони також мають певні недоліки [7]–[9].

Підвищення рівня безпеки:

– спокій – забезпечує впевненість батьків та опікунів, оскільки вони знають про місцезнаходження своїх близьких;

– реагування на надзвичайні ситуації – сприяє швидшому реагуванню у разі надзвичайних ситуацій, потенційно рятуючи життя.

Покращена координація дій в родині:

– планування і логістика – спрощує координацію повсякденної діяльності, прийом і висадку пасажирів;

– комунікація – завдяки вбудованим в додаток повідомленнями і оповіщенням поліпшується спілкування в сім'ї.

Зручність і доступність:

– зручність у використанні – зазвичай розробляється з інтуїтивно зрозумілими інтерфейсами, в яких легко орієнтуватися;

– широка доступність – більшість програм доступні на основних платформах (iOS, Android) і часто є безкоштовними або недорогими.

Технологічна інтеграція:

– сумісність з розумними пристроями – інтеграція з мобільними пристроями та системами домашньої автоматизації покращує функціональність;

– розширені можливості – постійні оновлення і додавання функцій роблять застосунки актуальними і корисними [7]–[9].

Аналіз поведінки:

– історія місцезнаходжень – допомагає батькам та опікунам зрозуміти розпорядок дня та звички членів сім'ї;

– моніторинг швидкості і активності – дозволяє отримати уявлення про поведінку за кермом і фізичної активності.

Більшість же недоліків пов’язані саме з етичними питаннями використання [7]–[9].

Питання конфіденційності:

– безпека даних – ризики, пов'язані з можливим неправильним використанням даних про місцезнаходження;

– вторгнення в приватне життя – члени сім'ї, особливо підлітки, можуть відчувати, що хтось вторгається в їх приватне життя.

Залежність від технологій:

– розряд батареї – постійне використання GPS та інших функцій відстеження може призвести до швидкого розряду батареї пристрою;

– технічні проблеми – надійність може бути знижена через помилки в програмному забезпеченні, проблеми з мережею або збої в роботі пристрою.

Можливість неправильного використання [7]–[9]:

– надмірна залежність – родини можуть надмірно покладатися на додаток, що зменшує можливість спілкування віч-на-віч;

– зловживання моніторингом – існує ризик того, що додаток може використовуватися для надмірного спостереження або контролю за членами сім'ї, що може призвести до проблем з довірою [7]–[9].

Витрати і абонентська плата:

– преміум-функції – для деяких розширених функцій може знадобитися плата за підписку, яка з часом може збільшитися;

– покупки у застосунку – додаткові функції або послуги можуть пропонуватися у вигляді покупок в застосунку, що збільшує загальну вартість.

Помилкові тривоги та оповіщення [7]–[9]:

– неточні оповіщення – неточності GPS або системні помилки можуть викликати помилкові оповіщення, викликаючи непотрібне занепокоєння;

– втома від сповіщень – часті сповіщення можуть призвести до того, що користувачі ігнорують або вимикають сповіщення, знижуючи ефективність програми [7]–[9].

ПЗМтаПМП надають широкий спектр функцій, які значно підвищують безпеку та координацію дій членів сім'ї. Їх популярність обумовлена відчутними перевагами, які вони надають, включаючи відстеження в режимі реального часу, оповіщення про надзвичайні ситуації і простоту спілкування. Однак вони також створюють проблеми, пов'язані з конфіденційністю, технологічною залежністю та потенційним зловживанням [7]–[9]. Баланс між цими плюсами і мінусами має вирішальне значення для користувачів, щоб максимізувати переваги цих додатків і знизити пов'язані з ними ризики. Оскільки технології продовжують розвиватися, постійні вдосконалення та інновації, ймовірно, дозволять вирішити багато з цих проблем, ще більше зміцнюючи роль застосунку для визначення місця розташування сім'ї в сучасному сімейному житті [7]–[9].

1.2.1 Застосунок Life360

Life360 – це популярний застосунок для визначення місцезнаходження сім'ї та безпеки, розроблений для того, щоб допомогти сім'ям залишатися на зв'язку та бути захищеними, надаючи інформацію про місцезнаходження в режимі реального часу та набір додаткових функцій безпеки [10], [11]. Застосунок Life360, доступний як на платформах iOS, так і на Android, задовольняє потреби сучасних сімей, інтегруючи різні функції, що підвищують безпеку, комунікацію та координацію [10], [11].

Ключові особливості Life360:

– відстеження місцезнаходження в режимі реального часу – Life360 дозволяє членам сім'ї ділитися своїми місцезнаходженнями на карті в режимі реального часу, гарантуючи, що всі знають про місцезнаходження один одного [10], [11];

– PIN-коди місцеположення – користувачі можуть додавати PIN-коди місцеположення, щоб ділитися певними місцями з членами сім'ї;

– спеціальні місця – користувачі можуть встановлювати геозони навколо важливих місць, таких як будинок, школа чи робота;

– повідомлення про прибуття та від'їзд – застосунок автоматично надсилає сповіщення, коли член сім'ї входить у вказане місце або залишає його;

– відстеження руху – Life360 записує історію переміщення, дозволяючи користувачам переглядати місця, які відвідували члени сім'ї за останні кілька днів [10], [11];

– маршрути подорожей – користувачі можуть переглядати детальні маршрути подорожей, які проходить кожен член сім'ї;

– функція розпізнавання руху – застосунок автоматично визначає, коли хтось із членів сім'ї знаходиться за кермом, і відстежує їх поїздку;

– моніторинг поведінки за кермом – відстежує такі параметри водіння, як швидкість, використання телефону під час руху, різке гальмування і прискорення [10], [11];

– виявлення аварії – у разі автомобільної аварії додаток може виявити аварію і відправити оповіщення контактам в екстрених службах;

– кнопка SOS – вбудована кнопка SOS дозволяє користувачам відправляти сигнал лиха разом зі своїм місцем розташування призначеним контактам в екстрених службах;

– цілодобова служба екстреної допомоги – користувачі преміум-класу мають доступ до цілодобових служб екстреної допомоги для негайної допомоги;

– сімейний чат – Life360 надає платформу обміну повідомленнями, що дозволяє членам сім'ї спілкуватися безпосередньо в застосунку;

– групові повідомлення – користувачі можуть одночасно надсилати повідомлення всім членам свого сімейного кола [10], [11];

– стан батареї – застосунок відображає рівень заряду всіх пристроїв членів сім'ї, що дозволяє користувачам стежити за тим, щоб телефон кожного залишався зарядженим;

– оповіщення про низький заряд батареї – відправляє повідомлення, коли батарея пристрою члена сім'ї розряджається [10], [11].

Life360 (рис. 1.1) – це надійний сімейний застосунок для визначення місцезнаходження та безпеки, який набув популярності завдяки своїм комплексним функціям, призначеним для підвищення безпеки та підключення сімей один до одного [10], [11]. Використовуючи функцію відстеження місцеположення в режимі реального часу, геозону, засоби безпеки водіння та оповіщення про надзвичайні ситуації, Life360 надає цінну послугу, яка відповідає різноманітним потребам сучасних сімей. Незважаючи на деякі занепокоєння щодо конфіденційності та витрат, переваги програми з точки зору безпеки та координації роблять її життєво важливим інструментом для багатьох сімей [10], [11].

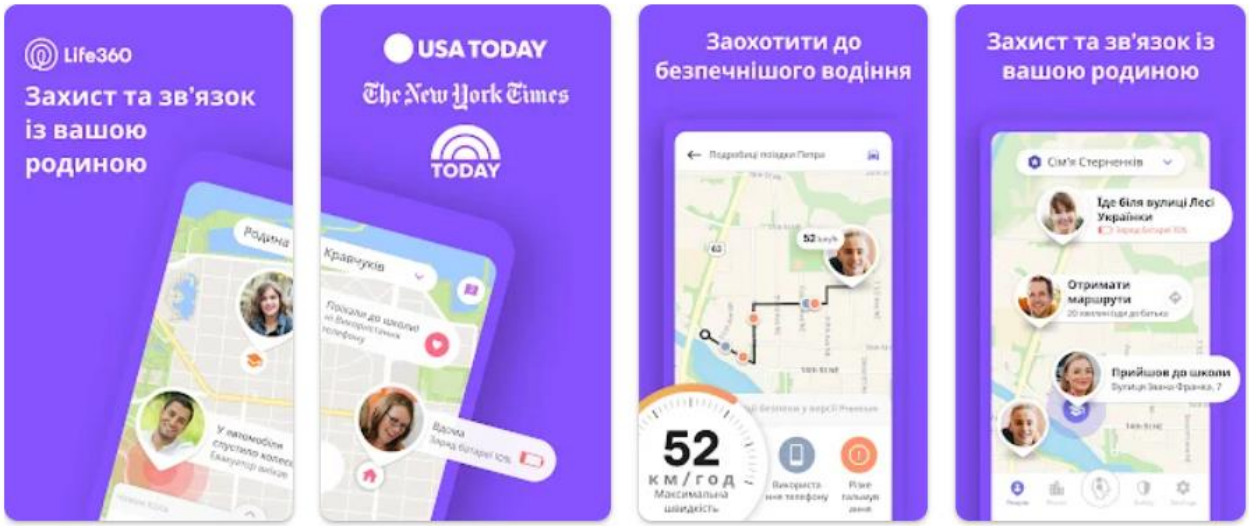

Рисунок 1.1 – Life360 [10]

1.2.2 Програмний комплекс Google Family Link

Google Family Link – це програма, розроблена для того, щоб допомогти батькам керувати та відстежувати діяльність своїх дітей в Інтернеті та їх фізичне місцезнаходження [12], [13]. Використовуючи різні функції, воно надає повний набір інструментів для забезпечення безпеки дітей як онлайн, так і офлайн. Ось детальний огляд того, як Google Family Link допомагає відстежувати місцезнаходження членів сім'ї, особливо дітей [12], [13].

Функції моніторингу місцезнаходження не є основними в застосунку, проте гарантують надійну роботу низки сервісів:

– відстеження місцезнаходження в режимі реального часу – інформація про місцезнаходження в режимі реального часу: Google Family Link дозволяє батькам бачити місцезнаходження своєї дитини на карті в режимі реального часу. Ця функція забезпечує спокій батьків, оскільки вони знають, де перебувають їхні діти в будь-який момент часу [12], [13];

– історія місцезнаходжень – хоча Family Link в основному відстежує оновлення в режимі реального часу, він також дозволяє батькам переглядати останні місця, щоб зрозуміти, де була їхня дитина;

– геозона (обмежена) – хоча Google Family Link не настільки надійний, як спеціалізовані програми розташування сім'ї, такі як Life360, він надає деякі основні можливості геозони. Батьки можуть налаштувати сповіщення про прибуття або від'їзд дитини з певних місць, таких як школа чи будинок [12], [13];

– управління пристроями – застосунок надає можливість визначити місце розташування пристрою дитини, що корисно, якщо пристрій втрачено або якщо необхідно визначити місце розташування дитини;

– дистанційне блокування пристрою – батьки можуть віддалено заблокувати пристрій дитини, щоб дитина могла зосередитися на певних завданнях, не відволікаючись на цифрові пристрої. Це також може бути корисно, якщо пристрій знаходиться в невідомому місці [12], [13];

– інтеграція з Google Maps – Google Family Link легко інтегрується з Google Maps, підвищуючи точність і зручність відстеження місцеположення. Батьки можуть використовувати Карти Google, щоб точно визначити місцезнаходження своєї дитини;

– звіти про діяльність – окрім інформації про місцезнаходження, Google Family Link (рис. 1.2) надає детальні звіти про використання програми та час, проведений на екрані, допомагаючи батькам забезпечити загальне самопочуття своєї дитини [12], [13].

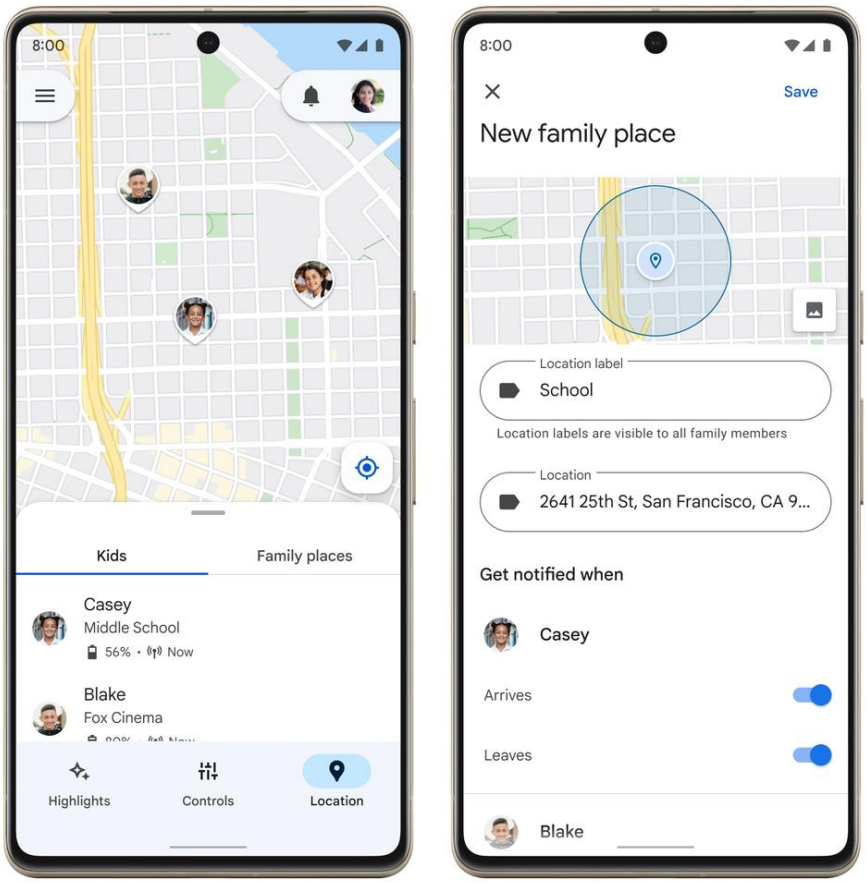

Рисунок 1.2 – Приклад Google Family Link [12]

1.2.3 Застосунок Family GPS Tracker Desktop

Family GPS Tracker Desktop – це програма, доступна в Магазині Microsoft Store, яка має на меті допомогти користувачам відстежувати

місцезнаходження членів своєї родини за допомогою пристроїв Windows [14]. Ця програма є частиною більш широкої категорії ПЗМтаПМП, що надають функціональні можливості, призначені для підвищення безпеки і координації в сім'ях. Ось детальний огляд того, як працює Family GPS Tracker Desktop, його функцій та переваг [14]:

– відстеження місцезнаходження в режимі реального часу – оперативні оновлення, застосунок дозволяє відстежувати місцезнаходження членів сім'ї в режимі реального часу, відображаючи їх поточне місце розташування на карті [14];

– інтеграція з картами – застосунок інтегрується з картографічними сервісами, щоб надавати візуальне уявлення про місцезнаходження кожного члена сім'ї [14];

– спільний доступ на основі дозволів – члени сім'ї можуть ділитися своїм місцезнаходженням з іншими членами своєї групи. Для цього потрібна згода та налаштування, що гарантує контрольований та узгоджений обмін інформацією про місцезнаходження [14];

– параметри, що налаштовуються – користувачі можуть налаштувати, як і коли надавати спільний доступ до свого місцезнаходження, що забезпечує гнучкість у збереженні конфіденційності;

– можливості геозони – користувачі можуть встановлювати геозони навколо певних областей, таких як будинок, школа чи робота;

– повідомлення про прибуття та від'їзд: додаток надсилає сповіщення, коли хтось із членів сім'ї входить або залишає певне місце, допомагаючи батькам відстежувати рух своїх дітей;

– відстеження минулих переміщень – застосунок фіксує місця, які відвідували члени сім'ї з часом, що дозволяє користувачам переглядати історію місцезнаходжень [14];

– візуалізація маршруту – надає наочне уявлення про пройдені маршрути, допомагаючи зрозуміти характер поїздок і звички;

– кнопка SOS – деякі версії програм родинного локатора мають кнопку SOS, яка в разі надзвичайних ситуацій надсилає сповіщення всім членам сім'ї про поточне місцезнаходження користувача [14];

– екстрені контакти – користувачі можуть налаштувати список екстрених контактів, які отримуватимуть сповіщення в надзвичайних ситуаціях;

– засоби комунікації – застосунок може включати функції обміну повідомленнями, що дозволяють членам сім'ї спілкуватися безпосередньо в застосунку;

– групові чати – користувачі можуть створювати групові чати для полегшення координації та спілкування між членами сім'ї;

– сумісність між платформами – програма розроблена для безперебійної роботи на різних пристроях, включаючи настільні комп'ютери Windows і, можливо, мобільні пристрої, якщо вони підтримуються [14];

– синхронізація пристроїв – забезпечує синхронізацію оновлень про місцезнаходження та сповіщень на всіх пристроях, що використовуються членами сім'ї.

Сімейний GPS Tracker Desktop – це цінний інструмент для сімей, які хочуть підвищити безпеку та покращити координацію за допомогою відстеження місцезнаходження в режимі реального часу та геозони. Завдяки таким функціям, як оперативне оновлення місцеположення, настроювані сповіщення [14], історія місцезнаходжень та екстрена допомога, він надає широкі можливості моніторингу. Незважаючи на деякі проблеми конфіденційності та технічного характеру, переваги з точки зору безпеки та душевного спокою роблять його корисним застосунокуом для багатьох сімей. Ефективно керуючи налаштуваннями конфіденційності та розуміючи його функціональні можливості, користувачі можуть максимально використати переваги Family GPS Tracker Desktop (рис. 1.3) для забезпечення безпеки своїх близьких і зв'язку з ними [14].

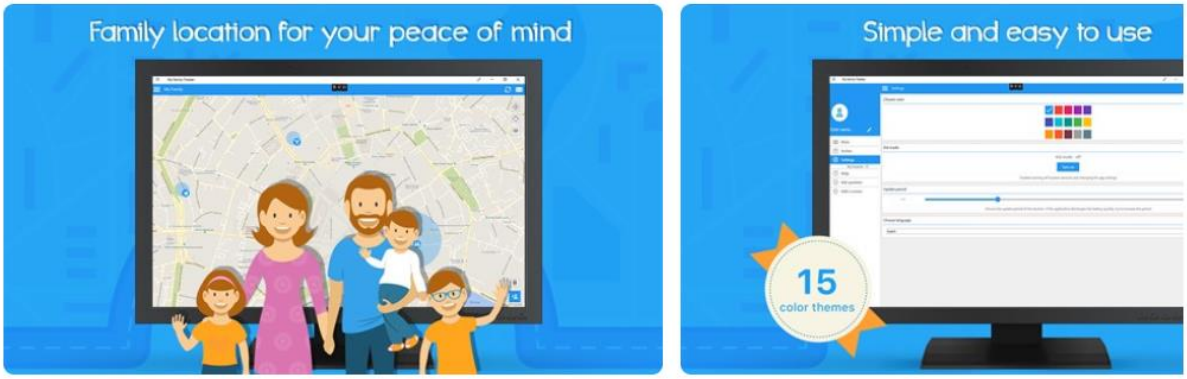

Рисунок 1.3 – Приклад Family GPS Tracker Desktop [14]

1.2.4 Дослідження та аналіз результатів порівняння

Порівняємо оглянуті програмні рішення та висвітлемо результати у вигляді таблиці. Для порівняння використаємо наступні критерії:

– відстеження місцезнаходження в режимі реального часу – можливість переглядати поточне місце розташування членів сім'ї на карті в режимі реального часу. Дозволяє батькам і опікунам бачити, де знаходяться їхні близькі в будь-який момент, що важливо для забезпечення безпеки і координації;

– історія місцезнаходжень – список місць, відвіданих членами сім'ї за певний період часу. Допомагає батькам відстежувати маршрути поїздок і стежити за тим, щоб діти дотримувалися узгоджених маршрутів і розкладу;

– геозона – можливість встановлювати віртуальні межі навколо певних зон, викликаючи оповіщення, коли хтось із членів сім'ї входить в ці зони або залишає їх. Підвищує безпеку, повідомляючи батьків, коли діти прибувають у певні місця або залишають їх, наприклад, вдома чи в школі;

– сповіщення про прибуття/відправлення – повідомлення, що надсилаються батькам, коли члени сім'ї входять або залишають певні місця. Забезпечує спокій і допомагає в координації повсякденної діяльності;

– обмін повідомленнями у застосунку – функція, що дозволяє членам сім'ї спілкуватися через додаток. Забезпечує швидку і зручну зв'язок без необхідності перемикатися між застосунками;

– функції екстреної допомоги/SOS – такі інструменти, як кнопка SOS, яка надсилає сповіщення про місцезнаходження користувача контактам екстрених служб. Вкрай важливо для надання негайної допомоги в надзвичайних ситуаціях, підвищення безпеки сім'ї;

– моніторинг заряду батареї – можливість переглядати рівень заряду пристроїв членів сім'ї. Забезпечує зарядку пристроїв членів сім'ї, що має вирішальне значення для безперервного відстеження та зв'язку;

– кросплатформність або сумісність – можливість використання програми в різних операційних системах (наприклад, Windows, iOS, Android). Гарантує, що всі члени сім'ї зможуть використовувати додаток незалежно від того, який пристрій у них є;

– засоби контролю конфіденційності – налаштування, що дозволяють користувачам контролювати, хто і коли може бачити їх місце розташування. Захищає конфіденційність користувачів і гарантує, що дані про місцезнаходження будуть передані тільки довіреним особам;

– вартість підписки – вартість, пов'язана з використанням Програми, включаючи безкоштовні версії і преміум-підписки. Впливає на доступність і доступ до розширених функцій.

– простота використання – наскільки застосунок зручний для користувача, включаючи налаштування та щоденне використання. Визначає, наскільки легко члени сім'ї різного віку можуть ефективно використовувати застосунок;

– інтеграція з картою – застосунок використовує картографічні сервіси для відображення місць розташування. Підвищує точність і зручність використання, надаючи чіткі візуальні уявлення місць розташування;

– налаштування повідомлень – можливість налаштовувати спосіб і час отримання повідомлень. Гарантує, що користувачі отримуватимуть відповідні сповіщення, не перевантажуючись непотрібними сповіщеннями;

– дозвіл на спільне використання місцеположення – визначає, хто може ділитися та переглядати інформацію про місцезнаходження. Забезпечує конфіденційність та безпеку, дозволяючи користувачам керувати налаштуваннями спільного доступу до місцезнаходження;

– блокування пристрою – можливість віддаленого блокування пристрою дитини. Допомагає батькам керувати екранним часом і переконатися, що пристрої використовуються належним чином;

– складність установки – рівень складності установки і настройки програми. Впливає на те, наскільки швидко і легко сім'ї зможуть почати користуватися застосунком;

Ці критерії були обрані виходячи з їх відповідності основним функціям ПЗМтаПМП. Вони враховують безпеку, зручність використання, конфіденційність, вартість та Сумісність, забезпечуючи цілісне уявлення про можливості та обмеження кожного додатка. У табл. 1.1 наведено порівняння програмного забезпечення для моніторингу та пошуку мобільних пристроїв.

Таблиця 1.1 – Програмне забезпечення для моніторингу та пошуку мобільних пристроїв

Критерії	Family GPS Tracker Desktop	Google Family Link	Life360
1	2	3	4
Відстеження місцезнаходження в режимі реального часу	Так	Так	Так
Історія місцезнаходжень	Так	Обмежено (останній місяць)	Так
Геозони	Так	Так	Розширені (можливість налаштувань)

Продовження таблиці 1.1

1	2	3	4
Сповіщення про прибуття / відправлення	Так	Так	Так
Повідомлення у застосунку	Так	Ні	Так
Функції екстреної допомоги / SOS	Так (Кнопка SOS)	Ні	Так (Кнопка SOS, аварійне сповіщення)
Моніторинг заряду батареї	Ні	Ні	Так
Кросплатформна Сумісність	Windows	Android	iOS, Android
Контроль конфіденційності	Так	Так	Так
Вартість підписки	Безкоштовно (+додатковий функціонал)	Безкоштовно	Безкоштовно (+додатковий функціонал)
Простота використання	Помірна (орієнтована на робочий стіл)	Легко	Легко
Інтеграція з картою	Так (Картографічні послуги)	Так (Google Maps)	Так (інтегроване відображення)
Налаштування повідомлень	Налаштовуються	Налаштовуються	Налаштовуються
Дозвіл на спільне використання місцеположення	Залежить від дозволів	Залежить від дозволів	Залежить від дозволів
Блокування пристрою	Ні	Так	Ні
Складність налаштування	Середня	Легко	Ускладнена

Усі застосунки мають ряд недоліків. Від так, Family GPS Tracker Desktop – може використовуватися тільки на пристроях на базі Windows, відсутні розширені функції безпеки водіння і інтеграція зі смарт-пристроями. Google Family Link має значно обмежені можливості геозони та відсутність функцій екстреної допомоги/SOS. Life360 для використання більшості найкорисніших сервісів вимагає преміум-підписки. А функції моніторингу

заряду батареї і інтеграції з пристроєм можуть привести до збільшення витрати заряду батареї.

Саме тому, актуальною задачею залишається програмна реалізація нового застосунку моніторингу та пошуку мобільних пристроїв.

1.3 Постановка завдання

Розробка програм для моніторингу та пошуку мобільних пристроїв є актуальним завданням, зумовленим багатьма факторами, включаючи підвищення особистої безпеки, зміни в суспільстві, технічний прогрес, психологічні переваги, юридичні наслідки, ринковий попит та етичні міркування. Ці програми є безцінними інструментами для сучасних сімей, забезпечуючи душевний спокій, покращуючи координацію рухів і забезпечуючи безпеку близьких. У міру подальшого розвитку технологій потреба в надійних, безпечних і зручних у використанні додатках для визначення місця розташування сім'ї буде тільки зростати, що робить їх розробку найважливішим пріоритетом для технологічних компаній і суспільства в цілому.

Отже актуальною задачею залишається програмна реалізація застосунку моніторингу та пошуку мобільних пристроїв.

1.3.1Функціональність системи

На підставі аналізу існуючого ПЗ можна виділити кілька недоліків, з яких основні:

- недостатня інтеграція зі сторонніми сервісами;
- неточність відображення звітів та іншої вихідної інформації;
- мала захищеність інформації, що зберігається;
- відсутність підтримки працездатності гаджетів.

З метою забезпечення універсального функціонального застосунку були поставлені завдання, основні з яких:

– реалізувати моніторинг та пошук користувача через смартфону;

– реалізувати відображення геолокації з мапою;

– реалізувати прихованість знаходження застосунку моніторинга на смартфоні дитини;

– розширити можливі варіанти використання розробленого ПЗ;

– протестувати процеси впровадження та експлуатації програми.

1.3.2Вимоги до інтерфейсу

Інтерфейс системи повинен відрізнятися лаконічністю та зручністю. З урахуванням роботи з геопозіціонуванням краще використовувати для цього мапу. для користувачів. Дизайн має бути адаптивним для зручності роботи на будь-якому дисплеї [15], [16].

1.3.3Вимоги до безпеки

Забезпечення конфіденційності та безпеки даних користувачів. Використання відповідних правил та рекомендацій щодо захисту даних користувачів та захисту інформації користувача.

1.3.4Не функціональні вимоги до продукту

Апаратні вимоги до сервера:

– центральний процесор – від 8 фізичних ядер з тактовою частотою від 4 GHz;

– оперативна пам’ять формату DDR5 SDRAM, з тактовою частотою від 6000 MHZ та об’ємом від 64 Гб;

– основний накопичувач (формату HDD або SSD) з вільним простором від 10 Гб.

Програмні вимоги до сервера:

– розгорнута серверна операційна система (ОС), наприклад: Windows Server, Linux Server або macOS Server;

– розгорнута SQL-сумісна система керування базами даних (СКБД).

Апаратні вимоги до клієнта:

– центральний процесор – від 2 фізичних ядер з тактовою частотою від 2,0 GHz (з можливістю Turbo Boost);

– оперативна пам’ять об’ємом від 4 Гб;

– накопичувач (формату HDD або SSD) з вільним простором від 0,5 Гб.

Програмні вимоги до сервера:

– наявність веббраузера актуальної версії.

1.4 Висновки за розділом 1

Цей розділ було присвячено критичному аналізу взаємозв'язку між різними існуючими розробками за темою проєкту. Реалізації, виконані різними авторами, мають дійсно важливе значення для створення основи для досліджень і надають інформацію про те, що було зроблено іншими людьми для вирішення даної проблеми.

2 МАТЕРІАЛИ І МЕТОДИ

Другий розділ буде присвячено питанням пов’язаним із із визначенням технології розробки. Вибір технології розробки, обґрунтування вибору мови програмування та фреймворків, СКБД та середовища розробки.

2.1 Вибір мови програмування

Python став однією з найпопулярніших мов програмування для розробки вебзастосунків. Зростання його популярності можна пояснити багатьма факторами, включаючи простоту, читабельність, широку бібліотечну підтримку та активну спільноту [17], [18].

Синтаксис Python розроблений таким чином, щоб бути інтуїтивно зрозумілим і максимально наближеним до людської мови. Це робить його відмінним вибором для початківців, які можуть швидко освоїти основні поняття, не стикаючись зі складними синтаксичними правилами. Простота синтаксису Python дозволяє розробникам більше концентруватися на вирішенні проблем, а не турбуватися про тонкощі самої мови [17], [18]. Код на Python добре читається і простий для розуміння, що знижує витрати і трудовитрати на підтримку і оновлення коду. Така читабельність особливо корисна у веброзробці, де проєкти можуть розвиватися та часто змінюватися. Можливість легко читати та розуміти код, написаний іншими, полегшує співпрацю між розробниками [17], [18].

Великі бібліотеки та фреймворки. Python постачається з багатою стандартною бібліотекою, яка надає модулі та функції для широкого кола завдань. Це означає, що розробникам не потрібно писати код з нуля для загальних функціональних можливостей, що прискорює процес розробки [17], [18].

Популярні вебфреймворки – Python може похвалитися потужними та популярними вебфреймворками, такими як Django та Flask, які спрощують процес веброзробки:

– Django – це вебфреймворк високого рівня, який заохочує швидку розробку та чистий, прагматичний дизайн. Він включає в себе безліч вбудованих функцій, таких як об'єктно-реляційне відображення, автентифікацію та користувальницький інтерфейс адміністратора. Філософія Django без батарейок означає, що він надає практично все необхідне розробнику для створення надійних вебзастосунків [17], [18];

– Flask – це мікро-фреймворк, який надає все необхідне для веброзробки, залишаючи при цьому простір для гнучкості. На відміну від Django, Flask надає розробникам більше контролю над компонентами, які вони хочуть використовувати, що робить його чудовим вибором для невеликих застосунків або для тих, хто хоче додати лише певні функції [17], [18].

На додаток до стандартної бібліотеки та фреймворків, Python має велику екосистему сторонніх бібліотек, доступних через індекс пакетів Python (PyPI). Ці бібліотеки охоплюють все: від доступу до бази даних, перевірки форм та автентифікації до більш вдосконалених функцій, таких як візуалізація даних, машинне навчання та інтеграція API [17], [18].

Python підтримує багато парадигм програмування, включаючи процедурне, об'єктно-орієнтоване та функціональне програмування. Така гнучкість дозволяє розробникам вибрати найкращий підхід для свого конкретного проєкту, що робить Python придатним для широкого кола застосунків [17], [18].

Python може легко інтегруватися з іншими мовами та технологіями. Наприклад, пітон може взаємодіяти з C / C++ для критичних до продуктивності компонентів, використання мови Jython для інтеграції з Java, використовувати або встановити IronPython для. Net-додатків.така сумісність робить Python універсальним вибором для вебдодатків, яким необхідно взаємодіяти з різними системами [17], [18].

З появою бібліотеки asyncio і зростаючою підтримкою асинхронних платформ, таких як Tornado і FastAPI, Python став добре підходити для вирішення асинхронних завдань, які часто використовуються у веброзробці, особливо для операцій введення-виведення [17], [18].

У Python існує велика і активна спільнота розробників, які вносять свій внесок в його розвиток і вдосконалення. Підтримка цієї спільноти гарантує розробникам доступ до безлічі ресурсів, включаючи документацію, навчальні посібники, форуми та конференції. Спільнота Python також активно підтримує та оновлює бібліотеки та фреймворки, забезпечуючи їх актуальність.

Python – це мова з відкритим кодом, що означає, що її можна вільно використовувати та розповсюджувати. Це сприяло його широкому впровадженню та розробці великої кількості інструментів та бібліотек з відкритим кодом, які ще більше розширюють можливості Python у веброзробці [17], [18].

Хоча Python традиційно не славиться своєю продуктивністю порівняно з такими мовами, як C або Java, для вирішення цих проблем були розроблені різні інструменти та методи. Наприклад, використання компіляторів JIT, таких як PyPy, може значно підвищити швидкість виконання Python. Крім того, здатність Python інтегруватися з C/C++ дозволяє розробникам оптимізувати критично важливі для продуктивності частини своїх застосунків [17], [18].

Фреймворки, подібні до Django, призначені для роботи з вебсайтами з високим трафіком і можуть масштабуватися відповідно до зростаючої бази користувачів. Модульний дизайн Django і можливість використання різних внутрішніх баз даних і механізмів кешування сприяють його масштабованості. Аналогічно, мікро-фреймворки, такі як Flask, дозволяють гнучко масштабувати програми, додаючи лише необхідні компоненти [17], [18].

Безпека є найважливішим завданням у веброзробці, а фреймворки Python, такі як Django, забезпечують вбудовані функції безпеки для захисту програм від поширених загроз. Ці функції включають захист від введення

SQL, міжсайтового сценарію, підробки міжсайтових запитів та перехоплення кліків [17], [18].

Активна спільнота розробників Python забезпечує швидке усунення вразливостей в системі безпеки і регулярний випуск оновлень і патчів. Такий проактивний підхід допомагає підтримувати безпеку та цілісність вебдодатків, створених за допомогою Python [17], [18].

Простота Python і доступність надійних платформ сприяють швидкій розробці. Розробники можуть швидко створювати прототипи та виконувати ітерації у своїх додатках, що особливо корисно в швидко мінливих середовищах розробки.

Поєднання читабельного синтаксису Python, великих бібліотек та потужних фреймворків покращує продуктивність. Розробники можуть зосередитися на вирішенні бізнес-завдань, а не на деталях програмування низького рівня, що прискорює процес розробки та скорочує час виходу на ринок [17], [18].

Python став мовою вибору для застосунків машинного навчання та штучного інтелекту завдяки таким бібліотекам, як TensorFlow, Keras та scikit-learn. Це робить Python привабливим варіантом для веброзробників, які хочуть інтегрувати можливості штучного інтелекту та машинного навчання у свої програми [17], [18].

Популярність Python у спільноті науковців даних, що підтримується такими бібліотеками, як Pandas та NumPy, дозволяє веброзробникам легко інтегрувати функції аналізу та візуалізації даних у свої вебпрограми.

Простота та універсальність Python також зробили його популярним інструментом для розробки IoT. Такі бібліотеки, як MicroPython та CircuitPython, дозволяють розробникам використовувати Python для програмування пристроїв IoT, які потім можуть взаємодіяти з вебпрограмами [17], [18].

Популярність Python для розробки вебзастосунків можна пояснити його простотою, читабельністю, великими бібліотеками та фреймворками,

гнучкістю, сильною підтримкою спільноти та надійними функціями безпеки. Здатність мови сприяти швидкому розвитку і високої продуктивності, а також його сумісність з новими технологіями роблять його ідеальним вибором для сучасної веброзробки. Використовуючи сильні сторони Python, розробники можуть створювати масштабовані, безпечні та підтримувані вебпрограми, які відповідають потребам сучасного динамічного цифрового ландшафту [17], [18].

Для обґрунтованого вибору проведемо порівняння найбільш популярних зв’язок мов програмування та фреймворків для веброзробки за такими критеріями:

– простота вивчення – легкість, з якою початківець розробник може освоїти технологію і почати її використовувати. Впливає на те, наскільки швидко нові розробники можуть стати продуктивними, і сприяє загальній швидкості розробки;

– синтаксис та читабельність – простота і ясність синтаксису мови, а також те, наскільки легко читати і розуміти код. Впливає на зручність супроводу коду, співпрацю між розробниками і простоту налагодження;

– продуктивність – ефективність мови/фреймворку з точки зору швидкості виконання та використання ресурсів. Критично для застосунків, що вимагають високої продуктивності та масштабованості, таких як програми в режимі реального часу або вебсайти з високим трафіком;

– масштабованість – здатність справлятися із збільшеним навантаженням і розширювати застосунок без зниження продуктивності. Важливо для застосунків, які передбачають збільшення кількості користувачів, даних або транзакцій.

– фреймворки – наявність фреймворків, які спрощують та прискорюють веброзробку. Фреймворки надають готові компоненти та інструменти, які полегшують розробку, зменшують кількість шаблонного коду та забезпечують найкращу практику;

– асинхронна підтримка – здатність мови/платформи ефективно обробляти асинхронні операції. Важливо для застосунків, які потребують обробки в режимі реального часу, таких як програми чату, прямі трансляції та операції вводу-виводу;

– підтримка баз даних –діапазон баз даних, з якими може взаємодіяти мова/фреймворк, і наскільки добре він їх підтримує. Забезпечує гнучкість у виборі відповідної бази даних для потреб програми та дозволяє інтегрувати її з існуючими джерелами даних;

– підтримка об'єктно-реляційного відображення (ORM) – наявність інструментів ORM, які полегшують взаємодію з базами даних. ORM полегшує операції з базами даних, дозволяючи розробникам взаємодіяти з базою даних за допомогою об'єктно-орієнтованого коду, зменшуючи потребу в оригінальному SQL;

– екосистема та бібліотеки – наявність сторонніх бібліотек та інструментів, що розширюють функціональність мови/фреймворку. Багата екосистема прискорює розробку, надаючи готові рішення для поширених проблем і розширюючи можливості програми;

– підтримка спільноти – розмір та рівень активності спільноти розробників, пов'язаної з мовою/фреймворком. Сильна спільнота надає ресурси, навчальні посібники, форуми та внески, які допомагають розробникам вирішувати проблеми та бути в курсі найкращих практик;

– функції безпеки – вбудовані або легко інтегровані заходи безпеки для захисту програми від поширених вразливостей. Необхідні для захисту даних користувачів та забезпечення відповідності стандартам безпеки;

– швидкість розробки – ефективність, з якою розробники можуть створювати та розгортати програми за допомогою мови/фреймворку. Більша швидкість розробки означає швидший час виходу на ринок та можливість швидкого повторення та вдосконалення застосунків;

– технічне обслуговування та оновлення – частота і якість оновлень мови/фреймворку, а також простота обслуговування додатки з плином часу.

Регулярні оновлення забезпечують безпеку, продуктивність і Сумісність технології з новими інструментами і стандартами. Простота технічного обслуговування знижує довгострокові витрати і трудовитрати;

– кросплатформна сумісність – можливість запуску додатків в різних операційних системах без змін. Забезпечує охоплення програми ширшою аудиторією та полегшує розробку для декількох платформ;

– популярні варіанти використання – типи програм та сценаріїв, для яких зазвичай використовується мова/платформа. Допомагає зрозуміти сильні сторони мови/платформи та їх відповідність вимогам конкретного проєкту;

– попит на ринку праці – попит на розробників, які володіють мовою/фреймворком, на ринку праці. Вказує на наявність можливостей працевлаштування та актуальність навичок у галузі.

У таблиці 2.1 наведено порівняння мов програмування.

Таблиця 2.1 – Порівняння мов програмування

Критерії	Python (Django/Flask)	C# (ASP.NET)	JavaScript (Node.js)
1	2	3	4
Простота вивчення	Просто	Помірно	Помірно
Синтаксис та читабельність	Простий і читабельний	Багатослівний і строгий	Помірний (синтаксис JavaScript)
Продуктив-ність	Помірна	Висока	Висока
Масштабова-ність	Високий рівень (з Django)	Дуже високий рівень (підтримка на рівні підприємства)	Високий (керований подіями, неблокуючий)
Фреймворки	Django (повнофункціональний), Flask (мікро)	ASP.NET Core (повнофункціональний)	Express.js, Koa.js (мікро)
Підтримка асинхронності	Обмежена у Django, хороший у Flask (з async)	Хороша (вбудована підтримка в ASP.NET Core)	Хороша

Продовження таблиці 2.1

1	2	3	4
Підтримка ORM	Django ORM, SQLAlchemy (Flask)	Entity Framework	Mongoose (MongoDB), Sequelize (SQL)
Підтримка баз даних	Безліч (PostgreSQL, MySQL, SQLite і т. д.)	Безліч (SQL Server, PostgreSQL, MySQL і т. д.)	Безліч (MongoDB, MySQL, PostgreSQL і т. д.)
Екосистема та бібліотеки	Багата екосистема, великі бібліотеки	Великі бібліотеки та інструменти .Net	Великі пакети npm
Підтримка спільноти	Велика та активна	Велика та орієнтована на підприємства	Динамічна
Функції безпеки	Вбудовані (Django), розширення (Flask)	Вбудовані надійні функції безпеки	Потрібні додаткові пакети (наприклад, Helmet)
Швидкість розробки	Високий рівень (швидка розробка за допомогою Django)	Від середнього до високого рівня	Високий рівень (швидке створення прототипів)
Обслугову-вання та оновлення	Регулярні оновлення, сильна підтримка спільноти	Регулярні оновлення, підтримувані Microsoft	Регулярні оновлення, що підтримуються спільнотою розробників з відкритим кодом
Кросплатфор-мна сумісність	Високий рівень (працює на різних ОС)	Високий рівень	Висока якість (працює на різних ОС)
Популярні варіанти використання	Вебзастосунки, API, CMS	Корпоративні програми, вебзастосунки, API	Програми реального часу, API, мікросервіси
Попит на ринку праці	Високий (особливо з Django)	Дуже високий (особливо в корпоративних Налаштуваннях)	Дуже високий

Слід відзначити сильні сторони Python у зв’язці з Django та Flask), серед яких простота, читабельність коду, багата екосистема з великими

бібліотеками, швидкість розробки. В той же час C# у зв'язці з ASP.NET та JavaScript з Node.js найчастіше виявляються або занадто складними для простих проєктів, або недостатньо захищеними для великих проєктів.

Python (з Django та Flask) ідеально підходить для швидкої розробки проєктів, де важлива читаність коду і зручність супроводу. Django чудово підходить для надійних, багатофункціональних застосунків, тоді як Flask забезпечує гнучкість для менших або більш спеціалізованих проєктів.

2.2 Вибір середовища розробки

Atom – це універсальний і легко налаштовуваний текстовий редактор, розроблений на GitHub. Він здобув популярність серед розробників на різних мовах програмування, включаючи Python [19], [20].

Atom славиться своїми можливостями налаштування. Користувачі можуть широко змінювати інтерфейс, починаючи з теми та колірних схем і закінчуючи комбінаціями клавіш та макетом [19], [20]. Ця гнучкість дозволяє розробникам Python адаптувати редактор до своїх особистих уподобань та потреб, створюючи середовище, яке максимально підвищує продуктивність та комфорт [19], [20].

Atom може похвалитися великою екосистемою пакетів і розширень, які розширюють його функціональність. Доступно багато пакетів для Python, таких як linter-flake8 для редагування коду, autocomplete-python для завершення та script для запуску сценаріїв Python безпосередньо в редакторі. Ці пакети перетворюють Atom на потужне інтегроване середовище розробки (IDE) для веброзробки Python [19], [20].

Завдяки таким пакетам, як autocomplete-python, Atom забезпечує надійне завершення коду. Ця функція дозволяє виконувати доопрацювання в міру введення тексту розробниками, прискорюючи кодування і зменшуючи кількість помилок [19], [20]. Для веброзробки Python завершення роботи може значно підвищити продуктивність, пропонуючи функції, класи та змінні зі

стандартної бібліотеки, а також встановлені пакети. Atom підтримує підсвічування синтаксису для Python з коробки. У поєднанні з широким вибором доступних тим ця функція допомагає зробити код більш читабельним і візуально привабливим. Розробники можуть вибирати з безлічі світлих і темних тим, щоб зменшити навантаження на очі і підвищити чіткість коду [19], [20].

Лінтінг має вирішальне значення для підтримки якості коду. Atom підтримує кілька інструментів лінтингу для Python, таких як pylint, flake8 та mypy. Ці інструменти допомагають виявляти потенційні помилки та застосовувати стандарти кодування, забезпечуючи чистоту та зручність обслуговування кодової бази [19], [20]. Пакети Linter в Atom надають відгуки про якість коду в режимі реального часу під час введення коду розробниками. Хоча Atom сам по собі не є повноцінною IDE, його можна розширити за допомогою таких пакетів, як python-debugger, для підтримки можливостей налагодження. Ці пакети дозволяють розробникам встановлювати точки зупинки, перевіряти змінні та виконувати код поетапно, що є необхідними функціями для налагодження вебзастосунків [19], [20].

Atom поставляється з вбудованою інтеграцією з Git, що полегшує керування версіями безпосередньо з редактора. Пакет GitHub для Atom надає простий інтерфейс для внесення змін, управління гілками та вирішення конфліктів. Ефективний контроль версій є критичним для веброзробки Python, і інтеграція Atom з Git спрощує цей процес [19], [20]. Функції співпраці, такі як Teletype для Atom, дозволяють розробникам ділитися своїм робочим простором з іншими користувачами в режимі реального часу. Це може бути особливо корисно при спільному програмуванні або перегляді коду з членами команди, сприяючи поліпшенню спільної роботи і комунікації в групах розробників [19], [20].

Atom – це кросплатформний редактор, доступний для macOS, Windows та Linux. Це дозволяє розробникам використовувати один і той же інструмент незалежно від їх операційної системи, що вигідно для команд, що працюють в

різних середовищах [19], [20]. Послідовні інструменти на різних платформах допомагають оптимізувати робочі процеси розробки та скоротити час навчання членів команди [19], [20].

Atom має велику та активну спільноту розробників, які вносять свій внесок у свою екосистему. Підхід, заснований на залученні спільноти, означає, що нові пакети та функції постійно розробляються та вдосконалюються. Для розробників Python це означає наявність величезної кількості ресурсів, включаючи пакети, розроблені спільнотою, які можуть задовольнити конкретні потреби або підвищити продуктивність [19], [20]. Для Atom доступна велика документація та навчальні посібники, як для головного редактора, так і для його численних пакетів. Це багатство ресурсів полегшує розробникам роботу з Atom і дозволяє дізнатися, як розширити його функціональність відповідно до їх конкретних вимог до веброзробки на Python [19], [20].

Atom порівняно легкий у порівнянні з деякими повнофункціональними IDE. Хоча він надає багато можливостей IDE за допомогою пакетів, його ядро залишається компактним. Це робить Atom хорошим вибором для розробників, які віддають перевагу швидкому та чуйному редактору, який не споживає надмірних системних ресурсів. Поєднуючи легкий редактор з потужними розширеннями, Atom може підвищити ефективність веброзробки на Python. Розробники можуть оптимізувати свій робочий процес, використовуючи лише ті функції, які їм потрібні, і уникаючи надмірностей, які іноді виникають при використанні більш складних IDE [19], [20].

Менеджер пакетів Atom – Atom Package Manager (apm), спрощує процес пошуку, установки пакетів і управління ними. Розробники можуть легко шукати пакети, пов'язані з Python, і встановлювати їх безпосередньо з Atom. Така простота використання дозволяє легко розширювати функціональність Atom у міру розвитку потреб проєкту. Atom підтримує автоматизацію завдань за допомогою таких пакетів, як build та script. Ці інструменти дозволяють розробникам автоматизувати повторювані завдання,

такі як виконання тестів, створення проєктів або розгортання коду. Автоматизація може значно прискорити процес розробки та зменшити ризик помилок [19], [20].

Пакет platformio-IDE-terminal інтегрує термінал безпосередньо в Atom, дозволяючи розробникам виконувати команди оболонки, не виходячи з редактора. Це особливо корисно для веброзробки Python, де розробникам часто доводиться запускати команди сервера, керувати віртуальними середовищами або використовувати засоби контролю версій [19], [20].

Atom безкоштовний і відкритий код, що робить його доступним для всіх розробників. Ліцензування не вимагає витрат, що може бути особливо вигідним для невеликих команд або окремих розробників. Відкритий код Atom також означає, що розробники можуть внести свій внесок у його розробку та налаштувати його відповідно до своїх потреб [19], [20].

Atom – чудовий вибір для веброзробки на Python завдяки високому ступеню Налаштування, великій екосистемі пакетів та підтримуючій спільноті. Його здатність інтегрувати функції, подібні до IDE, такі як автозавершення, редагування, Налагодження та керування версіями, робить його потужним інструментом, зберігаючи при цьому легкість текстового редактора. Сумісність між платформами, можливості співпраці та безкоштовний доступ ще більше підвищують його привабливість [19], [20].

Використовуючи можливості Atom, веброзробники на Python можуть створювати індивідуальне середовище розробки, що підвищує продуктивність, якість коду і загальну ефективність розробки. Незалежно від того, працюєте ви над невеликими особистими проєктами або співпрацюєте у великих командах, Atom забезпечує гнучкість і потужність, необхідні для ефективної розробки вебдодатків на Python [19], [20].

Для обґрунтованого вибору проведемо порівняння найбільш популярних середовищ розробки за такими критеріями:

– тип – вказує, чи є інструмент текстовим редактором, повноцінною IDE або дистрибутивом scientific Python з можливостями IDE. Допомагає визначити основне призначення і вбудовану функціональність інструменту;

– налаштування та гнучкість – ступінь, до якої користувачі можуть змінювати та розширювати інтерфейс та функціональність інструменту. Налаштування дозволяє розробникам створювати індивідуальне середовище розробки, яке відповідає їхнім конкретним потребам та уподобанням;

– процес вивчення – наскільки легко або складно новим користувачам освоїти інструмент і почати ефективно ним користуватися. Впливає на те, наскільки швидко розробники зможуть ефективно використовувати інструмент, що має вирішальне значення для адаптації та загальної швидкості розробки;

– автозавершення коду – здатність інструменту пропонувати варіанти доопрацювання коду в міру того, як їх вводить розробник. Покращує ефективність кодування та зменшує помилки, надаючи пропозиції та зменшуючи обсяг введення тексту;

– підсвічування синтаксису – здатність інструменту візуально розрізняти елементи коду (наприклад, ключові слова, змінні, рядки), використовуючи різні кольори та шрифти. Покращує читабельність коду та допомагає розробникам швидко ідентифікувати різні частини коду;

– налагодження – доступність та якість інструментів для виявлення та виправлення помилок у коді. Ефективні засоби налагодження мають вирішальне значення для вирішення проблем і забезпечення правильної роботи програми;

– компонування та аналіз коду – можливість аналізувати код на наявність помилок, застосовувати стандарти кодування та пропонувати вдосконалення. Допомагає підтримувати якість коду та узгодженість у всій кодовій базі;

– вбудований термінал – чи включає інструмент вбудований термінал для виконання команд оболонки. Забезпечує зручність, дозволяючи

розробникам запускати команди, не виходячи з IDE, що спрощує робочий процес розробки;

– інтеграція з системою контролю версій – інструмент підтримує системи контролю версій, такі як Git, і дозволяє керувати версіями вихідного коду. Необхідний для спільної розробки, дозволяючи декільком розробникам одночасно працювати над одним проєктом та ефективно керувати змінами;

– підтримка віртуального середовища – здатність інструменту керувати віртуальними середовищами Python, які ізолюють залежності проєкту. Гарантує, що залежності проєкту не конфліктують, і полегшує управління залежностями;

– управління пакетами – можливості інструменту для управління пакетами та бібліотеками та їх встановлення. Спрощує процес додавання та оновлення залежностей, що має вирішальне значення для підтримки та розширення функціональності проєкту;

– продуктивність – швидкодія інструменту і використання ресурсів під час розробки. Впливає на продуктивність розробників і зручність роботи користувачів, особливо на менш потужному обладнанні;

– кросплатформна сумісність – можливість запуску інструменту в різних ОС (наприклад, Windows, macOS, Linux). Забезпечує можливість використання інструменту в різних середовищах розробки і командами з різними уподобаннями щодо ОС;

– спільнота та підтримка – розмір і рівень активності спільноти користувачів і розробників, а також доступність ресурсів підтримки. Сильна спільнота надає цінні ресурси, плагіни, розширення та допомогу, покращуючи загальний досвід розробки;

– можливості IDE – вбудовані функції та інструменти, що надаються IDE для підтримки завдань розробки. Визначає, наскільки комплексним і функціональним є інструмент для різних видів діяльності з розробки;

– простота використання для веброзробки – наскільки добре інструмент підходить для завдань веброзробки, включаючи підтримку

вебфреймворків та робочих процесів розробки. Впливає на те, наскільки ефективно розробники можуть створювати, тестувати та розгортати вебпрограми;

– вартість – фінансові витрати, пов'язані з використанням інструменту, включаючи ліцензійні збори. Впливає на загальний бюджет розробки та поточного обслуговування, оскільки варіанти з відкритим кодом часто є більш економічними;

– інтеграція з вебплатформами – інструмент підтримує популярні вебплатформи та фреймворки, такі як Django та Flask. Розширює можливості розробки, надаючи функції та інструменти, специфічні для веброзробки.

– навчальні ресурси – доступність посібників, документації, курсів та інших освітніх матеріалів. Полегшує процес навчання для початківців розробників і підтримує безперервне навчання для досвідчених розробників.

Порівняння середовищ розробки наведено у табл. 2.2.

Таблиця 2.2 – Порівняння середовищ розробки

Критерії	Atom	PyCharm	Anaconda
1	2	3	4
Тип	Текстовий редактор з можливостями IDE за допомо-гою пакетів	Повнофункціональне середовище IDE	Дистрибутив IDE
Налаштування та гнучкість	Легко налаштовується за допомогою пакетів	Налашовується, але в меншій мірі, ніж Atom	Обмежені налаштування
Процес навчання	Легкий	Помірний	Помірний
Автозавершення коду	Так, за допомогою пакетів	Так (розширені функції)	Так, але більше орієнтований на наукові обчислення
Підсвічування синтаксису	Так	Так	Так
Налагодження	Так	Розширені засоби налагодження	Базове налагодження

Продовження таблиці 2.2

1	2	3	4
Підтримка віртуального середовища	Так, за допомогою пакетів	Так, вбудована та розширена підтримка	Так, через Conda
Компонування та аналіз коду	Так, за допомогою пакетів	Так, вбудовані та розширені можливості	Так, але в основному для наукового аналізу
Вбудований термінал	Так, за допомогою пакетів	Так, вбудовані	Так, вбудована
Інтеграція з системою контролю версій	Так, із вбудованою інтеграцією Git та GitHub	Так, вбудований Git та багато іншого	Базова інтеграція з Git
Управління пакетами	Так, через apm	Так, за допомогою вбудованого менеджера пакетів	Так, через Conda
Продуктивність	Висока	Низька	Помірна
Кросплатформна сумісність	Так	Так	Так
Спільнота та підтримка	Сильна спільнота, величезна екосистема пакетів	Сильна спільнота, чудова підтримка	Сильне наукове співтовариство, хороша підтримка
Можливості IDE	Може бути розширена до повноцінної IDE за допомогою пакетів	Повноцінне середовище IDE з усіма готовими функціями	Базові функції IDE, орієнтовані на обробку даних
Простота використання для веброзробки	Потрібні налаштування	Готове до використання IDE	Не орієнтовано на веброзробку
Вартість	Безкоштовне	Безкоштовне / Платне	Безкоштовне
Інтеграція з вебплатформами	За допомогою пакетів	Відмінна вбудована підтримка фреймворків	Обмежена
Навчальні ресурси	Великі ресурси спільноти, навчальні посібники	Велика документація, навчальні посібники, курси	Хороша документація та ресурси

Atom – це не повноцінна IDE від початку, проте слід відначити, що він легко налаштовується, високопродуктивний, має велику екосистему пакетів та кросплатформний. А отже може бут підлаштований під будь-які задачі, зберігаючи мобільність та продуктивність. Не останню роль в цьому відіграє – потужна підтримка спільноти.

2.3 Структурна схема розробки

Загальна схема системи за обраною темою на рис. 2.1.

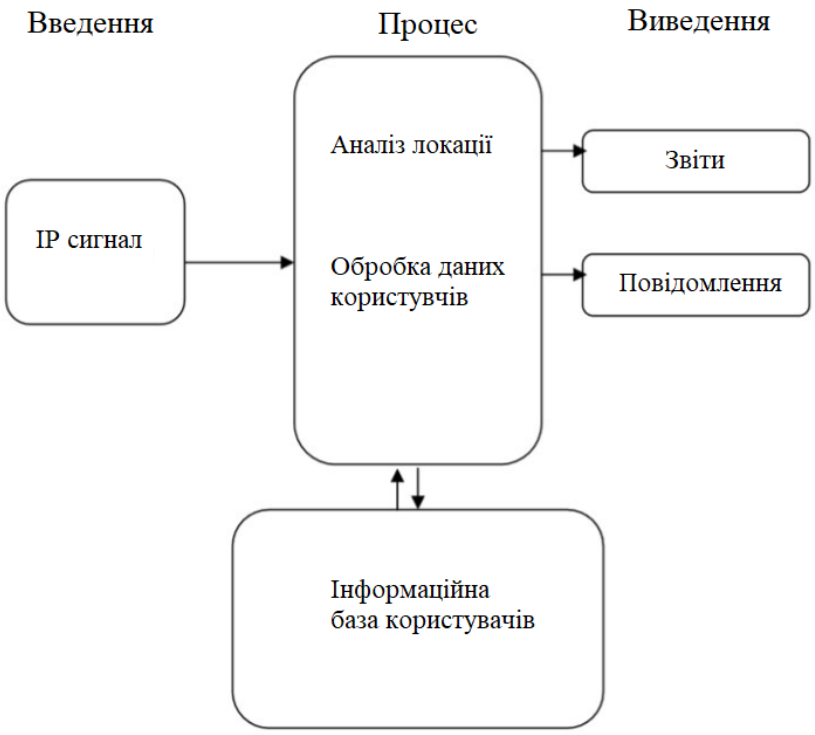

Рисунок 2.1 – Загальна схема системи за обраною темою

Основною інформацією, яка буде використовуватися на додаток до даних геолокації, є місцезнаходження GPS, ім'я дитини, ім'я батька чи опікуна та дата народження дитини. Вся ця інформація зберігається у зв'язаній базі даних, з якою проводиться регламентна синхронізація за розкладом, або в режимі реального часу.

2.4 Висновки за розділом 2

Другий розділ було присвячено питанням пов’язаним із визначенням технології розробки. Вибір технології розробки – кросплатформний вебзастосунок, обґрунтування вибору мови програмування – Python та фреймворків – Django та Flask, СКБД та середовища розробки – Atom.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проєктування моделей системи

Уніфікована мова моделювання (Unified Modeling Language – UML) є потужним інструментом для моделювання вебзастосунків з кількох причин. UML забезпечує стандартизований спосіб візуалізації дизайну системи, що полегшує зацікавленим сторонам розуміння, спілкування та співпрацю над розробкою вебзастосунків [21], [22].

Стандартизація. UML – це міжнародно визнаний стандарт, який підтримується Object Management Group (OMG). Стандартизація UML забезпечує його широке визнання і розуміння в різних галузях і командах розробників, сприяючи поліпшенню комунікації і спільної роботи [21], [22].

Узгодженість у проєктуванні. UML забезпечує узгоджений підхід до моделювання програмних систем. Послідовність допомагає підтримувати чіткість та рівномірність у процесі проєктування, полегшуючи новим членам команди розуміння існуючих моделей [21], [22].

Універсальність і гнучкість. UML включає різні типи діаграм, такі як діаграми класів, діаграми послідовностей, діаграми випадків використання, діаграми дій тощо. Ця універсальність дозволяє розробникам моделювати різні аспекти вебпрограми, від статичних структур до динамічної поведінки та взаємодій [21], [22].

Можливість налаштувань. UML-діаграми можуть бути налаштовані відповідно до конкретних потреб проєкту. Налаштованість гарантує, що UML може бути адаптований до різних методологій і специфічних вимог розроблюваного вебзастосунку [21], [22].

Комунікація та співпраця. UML надає загальну візуальну мову для представлення складних систем. Покращена комунікація між членами команди, зацікавленими сторонами та клієнтами гарантує, що кожен матиме чітке уявлення про дизайн та функціональність системи [21], [22].

Взаємодія із зацікавленими сторонами. Діаграми UML доступні як технічним, так і нетехнічним зацікавленим сторонам. Залучення зацікавлених сторін до процесу проєктування допомагає зібрати точні вимоги та відгуки, зменшуючи ризик непорозумінь [21], [22].

Комплексна документація. Діаграми UML служать детальною документацією щодо архітектури та дизайну системи. Детальна документація необхідна для майбутнього обслуговування, масштабованості та адаптації нових членів команди [21], [22].

Обслуговування та оновлення. Моделі UML можуть бути легко оновлені, щоб відобразити зміни в системі. Підтримка моделей в актуальному стані гарантує, що документація залишається актуальною та корисною протягом усього життєвого циклу програмного забезпечення [21], [22].

Вирішення проблем і проєктування. UML допомагає виявити потенційні проблеми проєктування на ранніх стадіях процесу розробки. Раннє виявлення проблем дозволяє своєчасно вносити виправлення, заощаджуючи час і ресурси в довгостроковій перспективі.

Візуалізація рішення. UML допомагає візуалізувати різні рішення та їх вплив на систему. Візуалізація рішень допомагає приймати обґрунтовані рішення та оптимізувати дизайн системи [21], [22].

Підтримка різних методологій розробки. UML підтримує ітеративні методи розробки, що використовуються в методологіях Agile. Ітеративне моделювання дозволяє постійно вдосконалювати дизайн системи відповідно до принципів Agile [21], [22].

Водоспадний і традиційний підходи. UML також підходить для традиційних методологій розробки на основі водоспаду. Гнучкість UML дозволяє використовувати його в різних методологіях управління проєктами та розробки, забезпечуючи широку застосовність.

Інтеграція із засобами розробки. UML підтримується численними інструментами моделювання, такими як Enterprise Architect, Rational Rose, Visual Paradigm та іншими. Підтримка інструментів підвищує продуктивність,

надаючи такі функції, як генерація коду, зворотне проєктування та інтеграція з середовищами розробки [21], [22].

Автоматизація та генерація коду. Деякі інструменти UML можуть генерувати скелети коду безпосередньо з діаграм. Автоматизація скорочує трудовитрати на ручне кодування і забезпечує узгодженість між проєктуванням і реалізацією [21], [22].

Детальне і цілісне уявлення. UML надає всебічне уявлення як про структуру, так і про поведінку системи. Цілісне представлення допомагає зрозуміти, як взаємодіють різні компоненти, забезпечуючи послідовне та добре інтегроване проєктування системи [21], [22]. Детальне моделювання – UML дозволяє детально моделювати різні аспекти програми, такі як взаємодія з користувачем, бізнес-процеси та системна архітектура. Детальне моделювання гарантує, що всі аспекти системи будуть враховані і точно представлені, що знижує ризик упущення важливих компонентів [21], [22].

UML є потужним інструментом для моделювання вебзастосунків завдяки своїй стандартизації, універсальності та здатності полегшувати спілкування та співпрацю між зацікавленими сторонами. Вона забезпечує комплексну структуру документації, підтримує різні методології розробки та добре інтегрується із засобами розробки [21], [22]. Пропонуючи детальне та цілісне уявлення про систему, UML допомагає виявляти та вирішувати проблеми проєктування на ранній стадії, забезпечуючи надійну та добре розроблену архітектуру програми. Широкий спектр діаграм дозволяє розробникам моделювати як статичні, так і динамічні аспекти програми, що робить UML незамінним інструментом у процесі розробки вебзастосунків [21], [22].

На рис. 3.1 наведено логічне проєктування системи за допомогою діаграми прецедентів, а саме прецеденту – «Передача інформації».

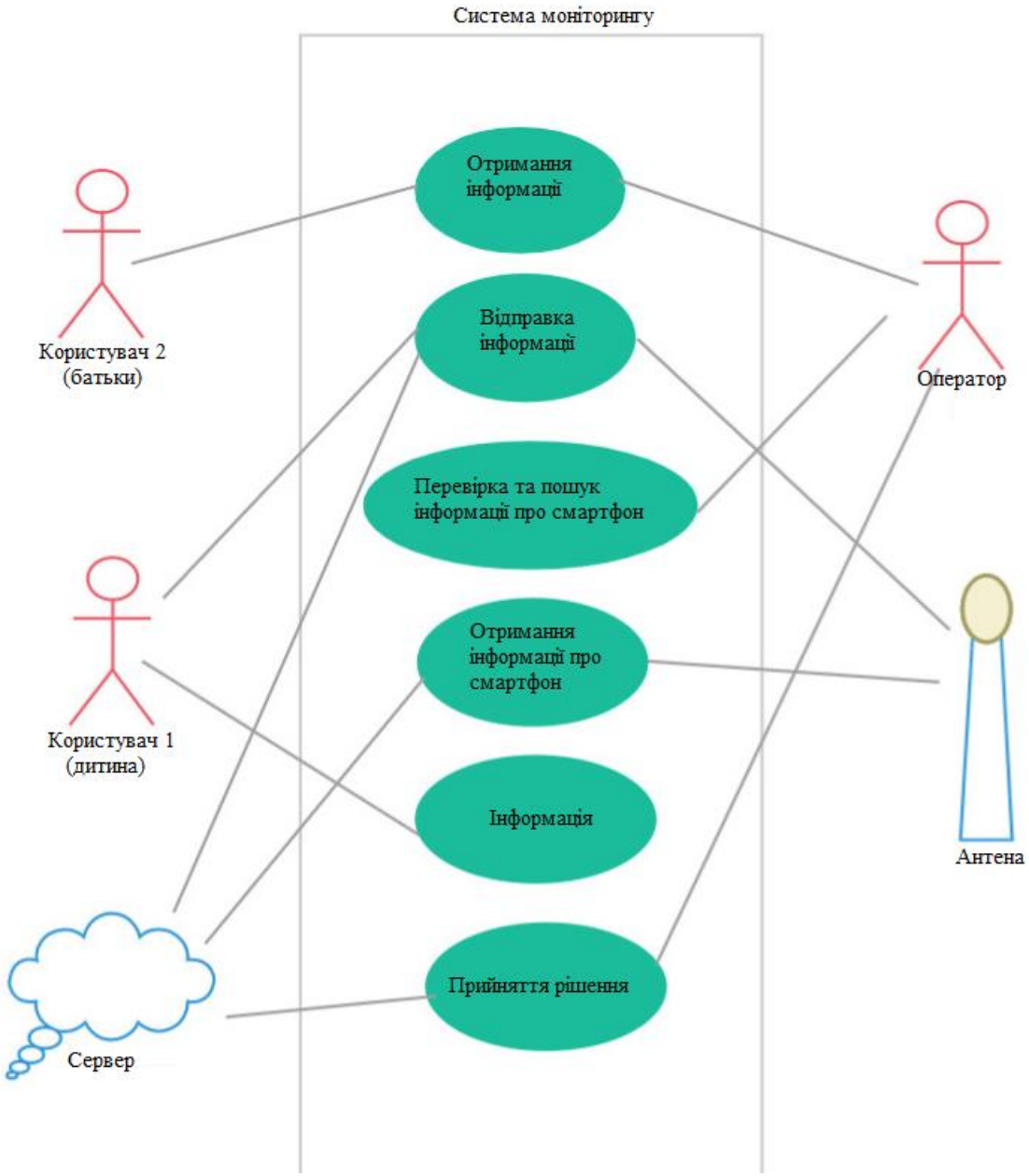

Рисунок 3.1 – Передача інформації

На даній діаграмі представлений процес передачі інформації між внутрішніми компонентами системи. На діаграмі видно, що в системі взаємодіють три користувача: оператор, користувач – дитина і користувач – батьки.

Зі смартфона дитини на сервер надходить певна інформація. Передача відбувається при наявності доступу в інтернет, що перевіряється наявністю з'єднання з антеною. При успішній відправці даних, вони потрапляють на сервер, після чого через фільтрацію даних, якою займається оператор (в

основному на підставі автоматизованих процесів) потрапляє на вихід до користувача – батьки, який в свою чергу зберігає цю інформацію у себе.

На рис. 3.2 представлена діаграма UML – «Місце розташування».

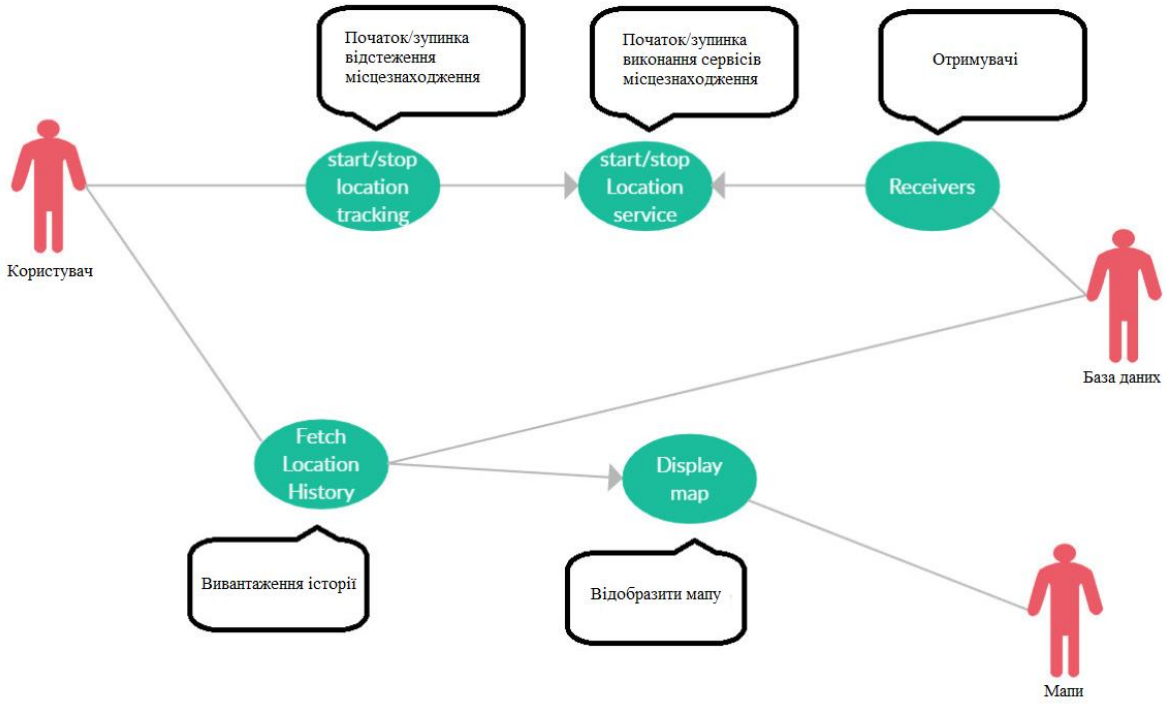

Рисунок 3.2 – Місце розташування

Дана діаграма відображає логіку отримання місцезнаходження пристрою і відображення на карті. Користувач своєю реєстрацією на сайті запускає процес відстеження місця розташування. З цим процесом взаємодіють інші процеси, такі як запуск сервісів місцезнаходження і функції отримання інформації.

Результатом даного процесу є карта, яка показує точне місце розташування пристрою.

На рис. 3.3 показана діаграма UML – «Користувачі».

На даній діаграмі показано поділ функціональних можливостей двох користувачів. Видно, що користувач – батьки може:

– знаходити пристрої;

– видаляти з нього дані;

– блокувати його;

– дивитися місцезнаходження на карті;

– відправляти повідомлення.

А користувач-дитина може: також відправляти повідомлення, відправляти дані, проходити реєстрацію.

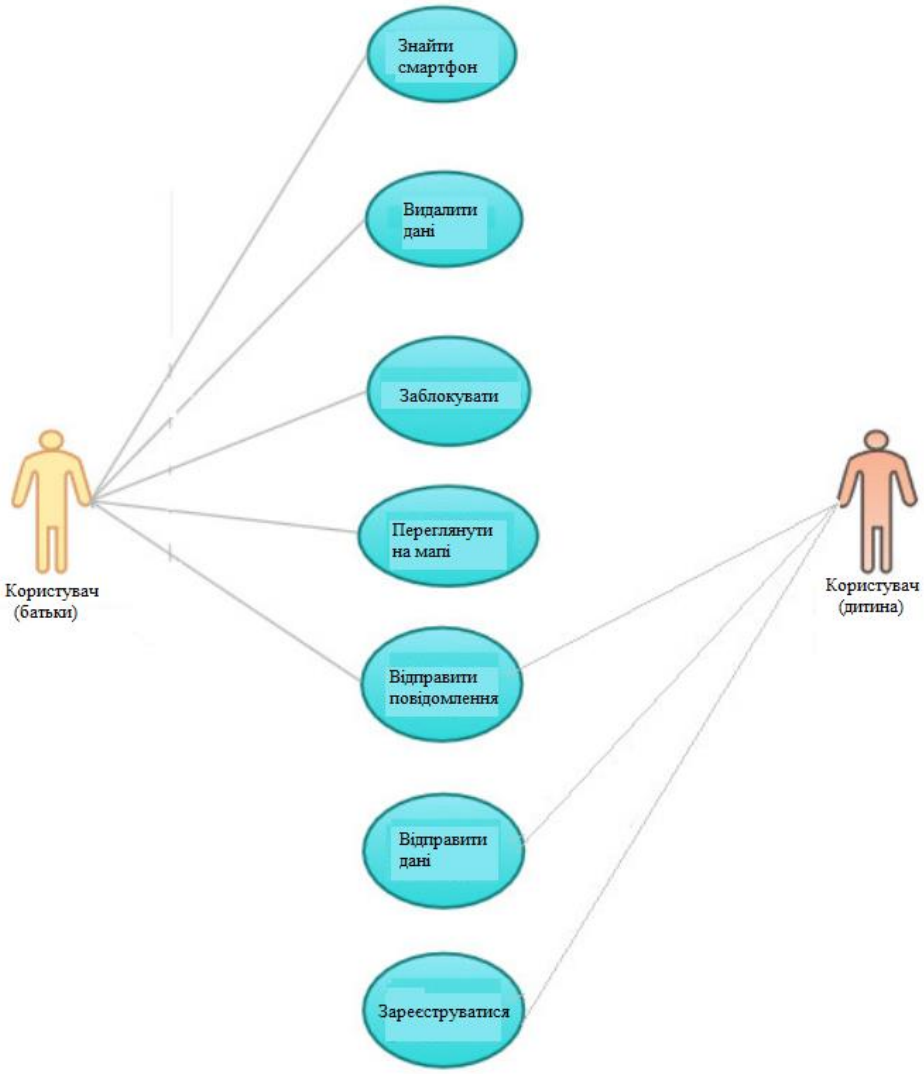

Рисунок 3.3 – Діаграма Користувачі

3.2 Проєктування інтерфейсу

Моделювання графічного інтерфейсу для вебзастосунку ПЗМтаПМП, вимагає ретельного розгляду кількох важливих функцій [23], [24]. Ці функції

мають вирішальне значення для забезпечення того, щоб програма була зручною для користувачів, ефективною та відповідала потребам своїх користувачів [23], [24].

Аутентифікація користувача та безпека:

– реєстрація – процес створення облікового запису для нових користувачів. Гарантує, що лише авторизовані користувачі зможуть отримати доступ до функцій ПЗМтаПМП. Він повинен бути простим, але безпечним і часто включати перевірку електронної пошти або номера телефону [23], [24];

– вхід в систему – безпечний спосіб для існуючих користувачів отримати доступ до своїх облікових записів. Безпечні методи аутентифікації, такі як двофакторна аутентифікація (2FA), підвищують безпеку. Параметри входу в соціальні мережі (наприклад, Google, Facebook) можуть спростити процес;

– профілі користувачів – управління користувальницької інформацією і настройками. Дозволяє користувачам оновлювати особисту інформацію, керувати налаштуваннями сповіщень та контролювати налаштування конфіденційності [23], [24];

– панель моніторингу – центральний вузол, що відображає підсумок ключової інформації. Надає користувачам швидкий огляд важливих даних, таких як місцезнаходження членів сім'ї, сповіщення та останні дії;

– навігаційне меню – меню для навігації по різних розділах програми. Надає користувачам легкий доступ до різних функцій, таких як карта, налаштування, сповіщення та довідка [23], [24];

– відстеження місцезнаходження в режимі реального часу;

– інтерактивна карта – динамічна карта, що відображає місцезнаходження членів сім'ї в режимі реального часу. Карта є основною функцією, що забезпечує візуальне уявлення місця розташування кожного члена сім'ї. Широко поширена інтеграція з такими сервісами, як Google Maps або OpenStreetMap [23], [24];

– значки для визначення місця розташування – маркери, що вказують точне місце розташування членів сім'ї. Пін-коди повинні бути чіткими і помітними, часто супроводжуватися написами або фотографіями для зручності ідентифікації;

– оновлення місця розташування – оновлення місцеположень в режимі реального часу без необхідності оновлення вручну. Забезпечує доступ користувачів до найактуальнішої інформації. Для цього потрібні ефективні серверні служби для надсилання оновлень у зовнішній інтерфейс [23], [24];

– повідомлення та оповіщення – повідомлення, коли член сім'ї входить в заздалегідь визначену зону або виходить з неї. Підвищує безпеку, повідомляючи користувачів про значні зміни місця розташування, наприклад, про прибуття в школу або з дому або при виході з них;

– аварійні оповіщення – сповіщення SOS надсилаються членами сім'ї, які потрапили в біду. Важливо для забезпечення безпеки, дозволяючи негайно повідомляти всіх членів сім'ї, якщо комусь потрібна термінова допомога [23], [24];

– комунікаційні функції – дозволяє користувачам відправляти повідомлення в застосунку. Забезпечує швидкий і прямий зв'язок між членами сім'ї, що може мати вирішальне значення в надзвичайних ситуаціях;

– контроль конфіденційності – опції для включення або відключення загального доступу до місця розташування. Гарантує, що користувачі зможуть контролювати свою конфіденційність, вибираючи, коли і з ким ділитися інформацією про своє місцезнаходження [23], [24];

– тимчасовий доступ до інформації про місцезнаходження – дозволяє тимчасово надавати доступ до інформації про місцезнаходження на певний термін. Корисно для короткострокових ситуацій, таких як надання інформації про ваше місцезнаходження під час подорожі або зустрічі з ким-небудь [23], [24];

– управління дозволами – контроль того, які функції програми мають доступ до даних про місцезнаходження. Підвищує безпеку, дозволяючи

користувачам управляти дозволами для додатків і забезпечуючи доступ до даних тільки при необхідності [23], [24];

– зручність використання та дизайн – зручний і зрозумілий дизайн. Гарантує, що користувачі різного віку та технічних здібностей зможуть ефективно орієнтуватися в додатку та користуватися ним;

– адаптивний дизайн – дизайн, який добре працює на різних пристроях і з різними розмірами екранів. Забезпечує послідовну та функціональну роботу програми на смартфонах, планшетах та настільних комп'ютерах [23], [24];

– доступність – функції, які роблять додаток зручним для людей з обмеженими можливостями. Інклюзивні методи проєктування забезпечують доступність програми для більш широкої аудиторії, в тому числі для людей з порушеннями зору або моторики;

– історія та звіти – журнал минулих місць для кожного члена сім'ї. Корисно для відстеження схем руху та перегляду минулих місць для безпеки або історичних цілей;

– звіти про діяльність – короткі звіти про події та зміни місцезнаходження. Надає інформацію про щоденні переміщення та поведінку членів сім'ї, що може бути корисним батькам, які спостерігають за дітьми, або опікунам старших родичів [23], [24];

– налаштування профілю – можливість персоналізувати профілі користувачів з фотографіями, іменами та уподобаннями. Можливість персоналізації підвищує залучення користувачів;

– налаштування програми – елементи управління різними функціями програми та повідомленнями. Гарантує, що користувачі зможуть адаптувати додаток до своїх потреб, наприклад, налаштувати частоту оновлення, типи повідомлень і налаштування конфіденційност [23], [24]і;

– інтеграція з календарем – синхронізація з додатками календаря для відображення подій і зустрічей. Допомагає користувачам співвідносити місце розташування із запланованими заходами [23], [24].

Моделювання графічного інтерфейсу для вебпрограми, такої як ПЗМтаПМП, передбачає інтеграцію безлічі функцій, спрямованих на безпеку, зручність користування, відстеження в режимі реального часу, спілкування, конфіденційність та налаштування. Кожна функція сприяє тому, щоб додаток стало не тільки функціональним, але і зручним для користувача і безпечним. Надаючи пріоритет цим функціям, розробники можуть гарантувати, що додаток відповідає потребам своїх користувачів та надає надійний інструмент для відстеження місцезнаходження членів сім'ї [23], [24].

Розуміння та ефективне впровадження цих функцій вимагає глибоких знань про вимоги користувачів, технічних обмежень та найкращих практик проєктування інтерфейсу користувача, що гарантує, що кінцевий продукт є одночасно надійним та інтуїтивно зрозумілим. [23], [24]

На рис. 3.4 та 3.5 наведено концепти майбутнього застосунку.

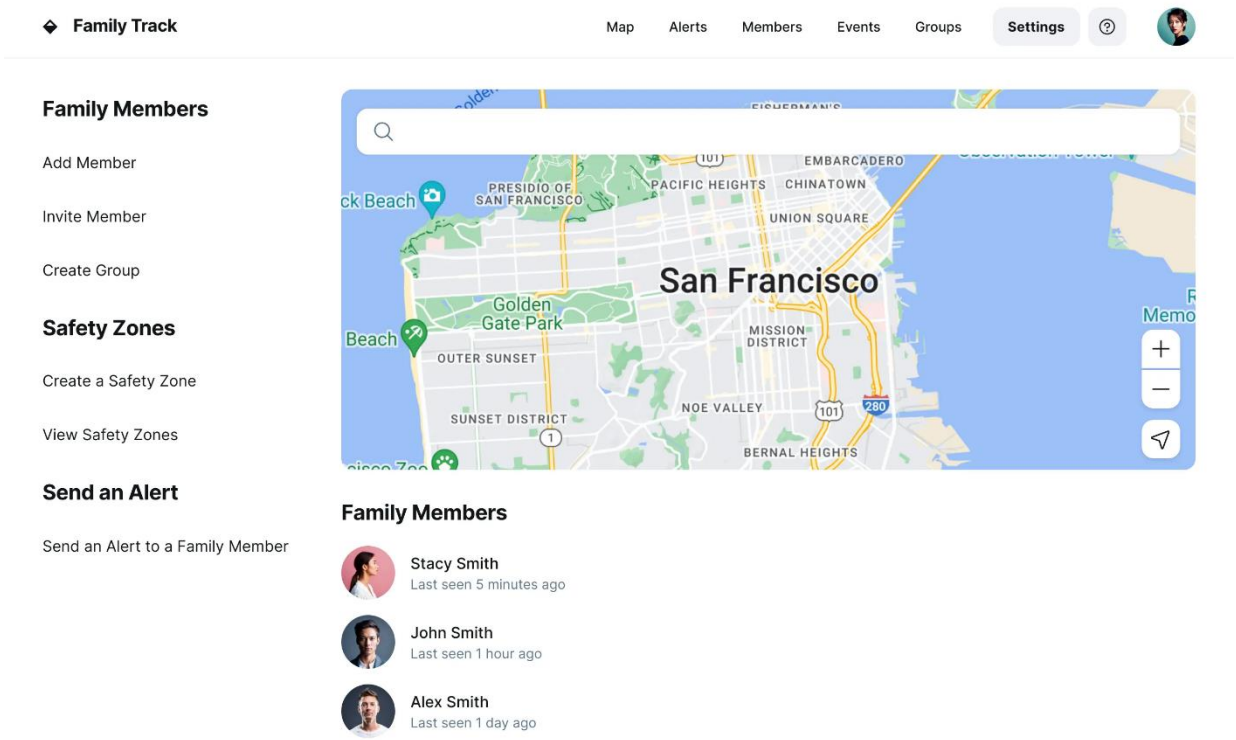

Рисунок 3.4 – Головна сторінка системи

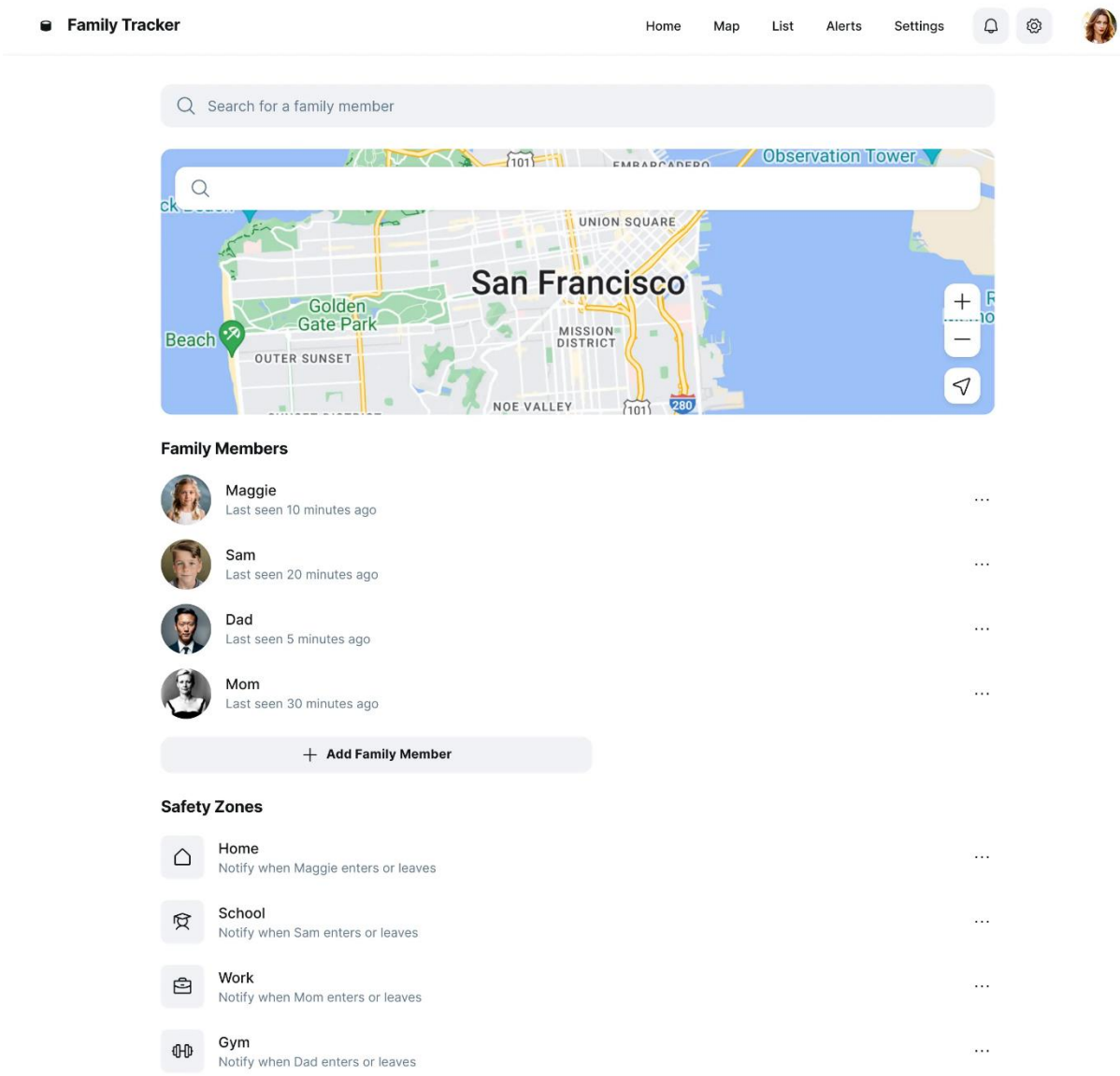

Рисунок 3.5 – Приклад сімейного кола та визначених геозон

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Розробка програм для моніторингу та пошуку мобільних пристроїв є актуальним завданням, зумовленим багатьма факторами, включаючи підвищення особистої безпеки, зміни в суспільстві, технічний прогрес, психологічні переваги, юридичні наслідки, ринковий попит та етичні міркування. Ці програми є безцінними інструментами для сучасних сімей, забезпечуючи душевний спокій, покращуючи координацію рухів і забезпечуючи безпеку близьких. У міру подальшого розвитку технологій потреба в надійних, безпечних і зручних у використанні додатках для визначення місця розташування сім'ї буде тільки зростати, що робить їх розробку найважливішим пріоритетом для технологічних компаній і суспільства в цілому.

Отже актуальною задачею залишається програмна реалізація застосунку моніторингу та пошуку мобільних пристроїв.

З метою забезпечення універсального функціонального застосунку були поставлені завдання, основні з яких:

- реалізувати моніторинг та пошук користувача через смартфону;
- реалізувати відображення геолокації з мапою;
- реалізувати прихованість знаходження застосунку моніторинга на смартфоні дитини;
- розширити можливі варіанти використання розробленого ПЗ;
- протестувати процеси впровадження та експлуатації програми.

4.2 Вимоги до інтерфейсу

Інтерфейс системи повинен відрізнятися лаконічністю та зручністю. З урахуванням роботи з геопозиціонуванням краще використовувати для цього

мапу. для користувачів. Дизайн має бути адаптивним для зручності роботи на будь-якому дисплеї.

4.3 Умови виконання програми

Апаратні вимоги до сервера:

– центральний процесор – від 8 фізичних ядер з тактовою частотою від 4 GHz;

– оперативна пам’ять формату DDR5 SDRAM, з тактовою частотою від 6000 MHZ та об’ємом від 64 Гб;

– основний накопичувач (формату HDD або SSD) з вільним простором від 10 Гб.

Програмні вимоги до сервера:

– розгорнута серверна ОС, наприклад: Windows Server, Linux Server або macOS Server;

– розгорнута SQL-сумісна система керування базами даних (СКБД).

Апаратні вимоги до клієнта:

– центральний процесор – від 2 фізичних ядер з тактовою частотою від 2,0 GHz (з можливістю Turbo Boost);

– оперативна пам’ять об’ємом від 4 Гб;

– накопичувач (формату HDD або SSD) з вільним простором від 0,5 Гб.

Програмні вимоги до сервера:

– наявність веббраузера актуальної версії.

4.4 Методика та результати тестування

Мануальне тестування відбувалося за сценарієм поетапного користування. Розроблені моки відповідали за імітацію роботи GPS у декількох користувачів одночасно.

Тож на рис. 4.1 наведено сторінку застосунку для авторизації.

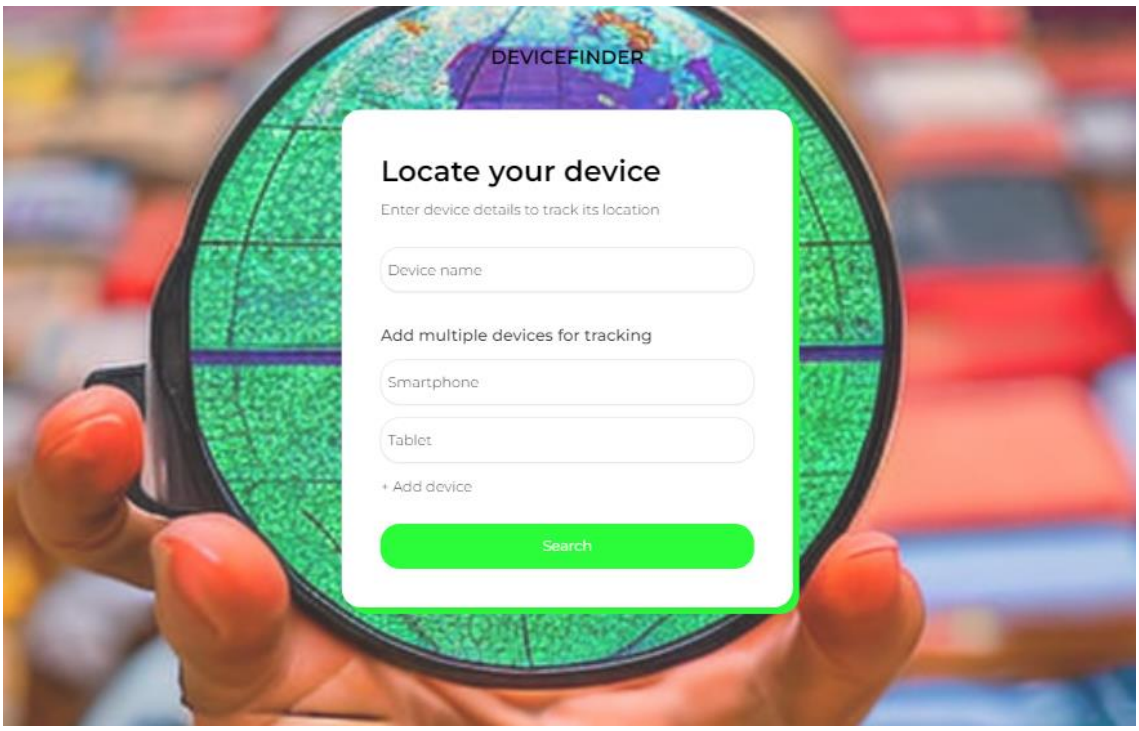

Рисунок 4.1 – Авторизація при вході в застосунок

На рис. 4.2 наведено приклад інформації про користувача.

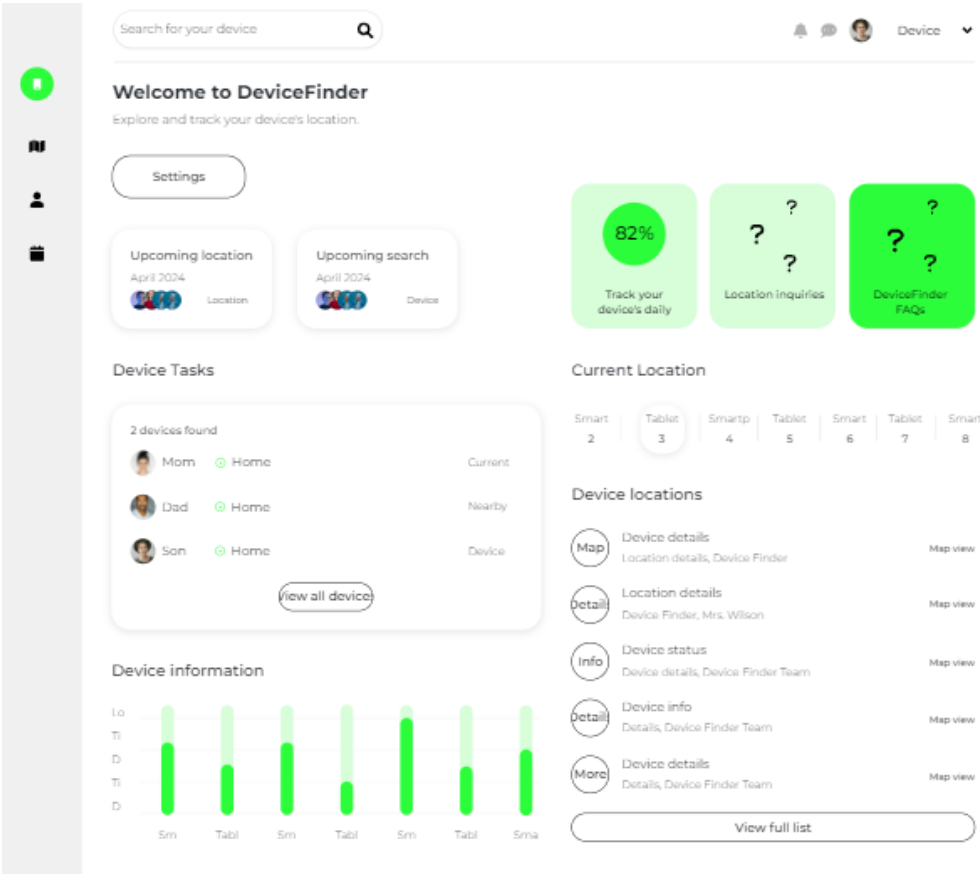

Рисунок 4.2 – Інформація про користувача

На рис. 4.3 наведено можливості застосунку в розгорнутому меню.

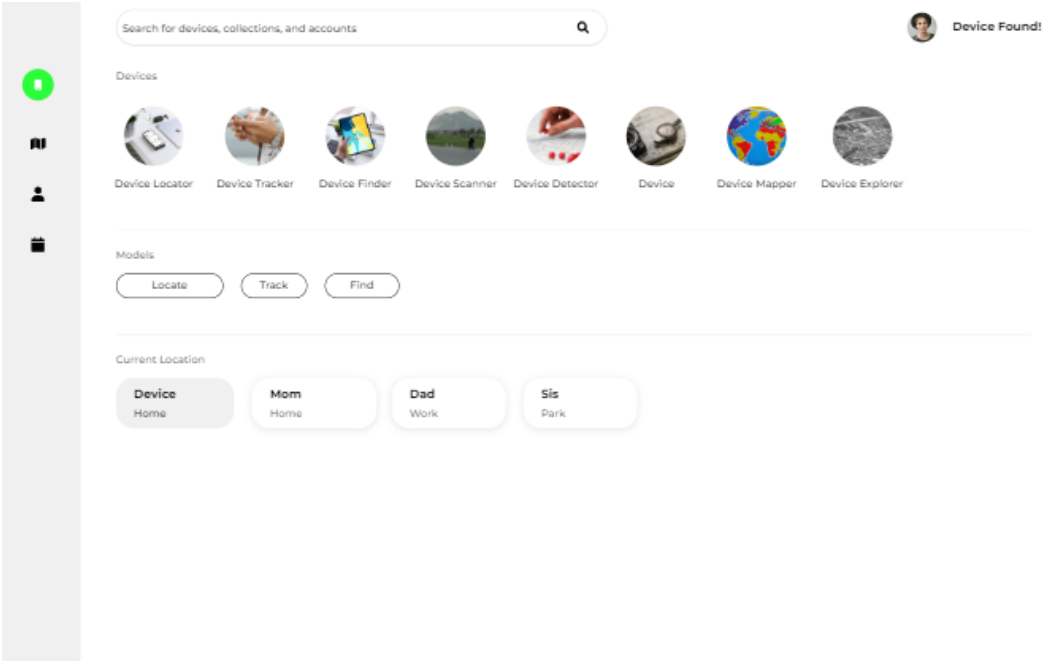

Рисунок 4.3 – Можливості застосунку в розгорнутому меню

На рис. 4.4 наведено сторінку підтримки для зв’язку та налаштувань з приводу застосунку.

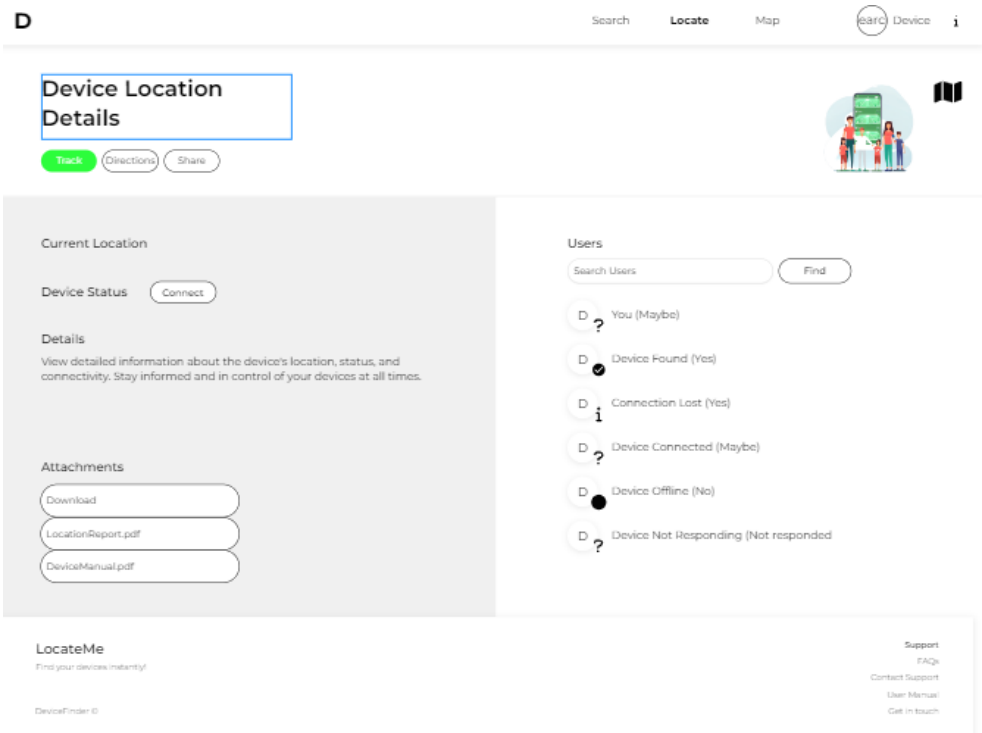

Рисунок 4.4 – Сторінка підтримки

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було виконано програмну реалізацію застосунку моніторингу та пошуку мобільних пристроїв. Для цього було виконано ряд завдань:

– досліджено та проаналізовано використання сімейних локаторів на мобільних пристроях;

– досліджено існуючі рішення для організації родинних локаторів (виявити їхні переваги та недоліки);

– сформовано набір функціональних вимог до застосунку моніторингу та пошуку мобільних пристроїв;

– спроєктовано та реалізувати архітектуру застосунку, який би реалізовував набір функціональних вимог;

– протестовано розробку;

– за результатами тестування сформовані висновки за виконаною роботою.

На підставі аналізу існуючого ПЗ можна виділити кілька недоліків, з яких основні:

– недостатня інтеграція зі сторонніми сервісами;

– неточність відображення звітів та іншої вихідної інформації;

– мала захищеність інформації, що зберігається;

– відсутність підтримки працездатності гаджетів.

З метою забезпечення універсального функціонального застосунку були поставлені завдання, основні з яких:

– реалізувати моніторинг та пошук користувача через смартфону;

– реалізувати відображення геолокації з мапою;

– реалізувати прихованість знаходження застосунку моніторинга на смартфоні дитини;

– розширити можливі варіанти використання розробленого ПЗ;

– протестувати процеси впровадження та експлуатації програми.

Застосунок було протестовано відповідно до сценарію його використання. Тестування засвідчило коректність роботи.

В результаті розробки реалізовано систему, що відповідає всім вимогам визначеним на початку розробки у вигляді функціональних вимог. В майбутньому можливо вдосконалити систему додавши функції енергохбереження.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. A Research on Mobile Applications for Location Tracking through Web Server and Short Messages Services (SMS) [Electronic resource]. – Access mode: https://www.researchgate.net/publication/299998138_A_Research_on_Mobile_Applications_for_Location_Tracking_through_Web_Server_and_Short_Messages_Services_SMS.

2. Location-Based Applications for Smartphones [Electronic resource]. – Access mode: https://www.researchgate.net/publication/275032124_Location-Based_Applications_for_Smartphones.

3. How to Track a Mobile Phone: A Comprehensive Guide [Electronic resource]. – Access mode: https://www.linkedin.com/pulse/how-track-mobile-phone-comprehensive-guide-monitoring-apps-sxt9c.

4. How to Track a Cell Phone Location for Free [Electronic resource]. – Access mode: https://www.airdroid.com/parent-control/track-cell-phone-location-free/.

5. Why a Family Locator is Essential for Your Peace of Mind? [Electronic resource]. – Access mode: https://www.mobicip.com/blog/why-family-locator-essential-your-peace-mind.

6. Where Are You? Location Tracking and the Promise of Child Safety [Electronic resource]. – Access mode: https://www.researchgate.net/publication/312542829_Where_Are_You_Location_Tracking_and_the_Promise_of_Child_Safety.

7. Best Family Locator Apps for iPhone and Android in 2024 [Electronic resource]. – Access mode: https://www.safetydetectives.com/blog/best-family-locator-apps-for-iphone-and-android/.

8. How to make a secure phone tracking app and family locator [Electronic resource]. – Access mode: https://support.google.com/googleplay/android-developer/thread/215942843/how-to-make-a-secure-phone-tracking-app-and-family-locator?hl=en.

9. How to Locate Family Members through Their Phones [Electronic resource]. – Access mode: https://www.airdroid.com/parent-control/locate-family-members-through-phone/.

10. Life 360 location app [Electronic resource]. – Access mode: https://www.orientatech.es/en/aplicacion-localizacion-life-360?gad_source=1&gclid=Cj0KCQjwxqayBhDFARIsAANWRnSMqngElEHm62tjTtYYlqkTQlK5emRRIeZbumfsIMyE93WxXUs9QOEaAvNVEALw_wcB.

11. Life360 [Electronic resource]. – Access mode: https://play-google-com.translate.goog/store/apps/details?id=com.life360.android.safetymapd&hl=ru&_x_tr_sl=ru&_x_tr_tl=uk&_x_tr_hl=uk&_x_tr_pto=sc.

12. Help keep your family safer online [Electronic resource]. – Access mode: https://families.google/familylink/.

13. Google Family Link [Electronic resource]. – Access mode: https://play.google.com/store/apps/details?id=com.google.android.apps.kids.familylink&hl=uk&gl=US.

14. Family GPS Tracker Desktop [Electronic resource]. – Access mode: https://apps.microsoft.com/detail/9nblggh4x9hv.

15. What is Requirement Analysis [Electronic resource]. – Access mode: https://www.simplilearn.com/what-is-requirement-analysis-article.

16. What Is Requirements Analysis? Process and Techniques [Electronic resource]. – Access mode: https://www.indeed.com/career-advice/career-development/requirements-analysis.

17. Flask vs. Django: Which Python framework suits you? [Electronic resource]. – Access mode: https://www.outsourceaccelerator.com/articles/flask-vs-django/?campaignid=21022134646&adgroupid=157508402846&matchtype=&keyword=&device=c&utm_campaign=&gad_source=1&gclid=Cj0KCQjwxqayBhDFARIsAANWRnQnqihmMMQkcwCgbEyaNX2XPCyxgfBMGNGN2jk8jxvkQtgvydqejdAaAi0BEALw_wcB.

18. Django vs Flask: Which is the Best Python Web Framework? [Electronic resource]. – Access mode: https://blog.jetbrains.com/pycharm/2023/11/django-vs-flask-which-is-the-best-python-web-framework/.

19. Django Intro [Electronic resource]. – Access mode: https://medium.com/jungletronics/django-intro-466127cf0093.

20. How to Setup Atom as a Python IDE? [Electronic resource]. – Access mode: https://programmers.io/blog/setup-atom-as-a-python-ide/.

21. Systematic Design of Web Applications with UML [Electronic resource]. – Access mode: https://www.pst.ifi.lmu.de/~kochn/uml-book.pdf.

22. A Practice of UML for Web Development [Electronic resource]. – Access mode: https://www.researchgate.net/publication/267200866_A_Practice_of_UML_for_Web_Development.

23. Web GUI Development [Electronic resource]. – Access mode: https://web.cse.ohio-state.edu/~shen.94/5544/webgui.pdf.

24. UI developer vs front-end developer explained [Electronic resource]. – Access mode: https://merge.rocks/blog/ui-developer-vs-front-end-developer-explained.

ДОДАТОК А

Технічне завдання

Вступ

Даний застосунок присвячений для моніторингу та пошуку мобільних пристроїв на основі їхнього місця знаходження.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмна реалізація застосунку моніторингу та пошуку мобільних пристроїв», затверджене наказом Національного університету «Запорізька політехніка» № 168 від 24 квітня 2024 р.

А.2 Основні вимоги до програми, що розробляється

Розробка програм для моніторингу та пошуку мобільних пристроїв є актуальним завданням, зумовленим багатьма факторами, включаючи підвищення особистої безпеки, зміни в суспільстві, технічний прогрес, психологічні переваги, юридичні наслідки, ринковий попит та етичні міркування. Ці програми є безцінними інструментами для сучасних сімей, забезпечуючи душевний спокій, покращуючи координацію рухів і забезпечуючи безпеку близьких. У міру подальшого розвитку технологій потреба в надійних, безпечних і зручних у використанні додатках для визначення місця розташування сім'ї буде тільки зростати, що робить їх розробку найважливішим пріоритетом для технологічних компаній і суспільства в цілому.

Отже актуальною задачею залишається програмна реалізація застосунку моніторингу та пошуку мобільних пристроїв.

А.2.1 Вимоги до функціональних характеристик

З метою забезпечення універсального функціонального застосунку були поставлені завдання, основні з яких:

– реалізувати моніторинг та пошук користувача через смартфону;

– реалізувати відображення геолокації з мапою;

– реалізувати прихованість знаходження застосунку моніторинга на смартфоні дитини;

– розширити можливі варіанти використання розробленого ПЗ;

– протестувати процеси впровадження та експлуатації програми.

А.2.2 Вимоги до інтерфейсу програми

Інтерфейс системи повинен відрізнятися лаконічністю та зручністю. З урахуванням роботи з геопозіціонуванням краще використовувати для цього мапу. для користувачів. Дизайн має бути адаптивним для зручності роботи на будь-якому дисплеї.

А.2.3 Вимоги до надійності

Забезпечення конфіденційності та безпеки даних користувачів. Використання відповідних правил та рекомендацій щодо захисту даних користувачів та захисту інформації користувача.

А.2.4 Вимоги до складу та параметрів технічний засобів

Апаратні вимоги до сервера:

– центральний процесор – від 8 фізичних ядер з тактовою частотою від 4 GHz;

– оперативна пам’ять формату DDR5 SDRAM, з тактовою частотою від 6000 MHZ та об’ємом від 64 Гб;

– основний накопичувач (формату HDD або SSD) з вільним простором від 10 Гб.

Програмні вимоги до сервера:

– розгорнута серверна ОС, наприклад: Windows Server, Linux Server або macOS Server;

– розгорнута SQL-сумісна система керування базами даних (СКБД).

Апаратні вимоги до клієнта:

– центральний процесор – від 2 фізичних ядер з тактовою частотою від 2,0 GHz (з можливістю Turbo Boost);

– оперативна пам’ять об’ємом від 4 Гб;

– накопичувач (формату HDD або SSD) з вільним простором від 0,5 Гб.

Програмні вимоги до сервера:

– наявність веббраузера актуальної версії.

ДОДАТОК Б

Опис програми

Б.1 Загальні відомості

Розробка програм для моніторингу та пошуку мобільних пристроїв є актуальним завданням, зумовленим багатьма факторами, включаючи підвищення особистої безпеки, зміни в суспільстві, технічний прогрес, психологічні переваги, юридичні наслідки, ринковий попит та етичні міркування. Ці програми є безцінними інструментами для сучасних сімей, забезпечуючи душевний спокій, покращуючи координацію рухів і забезпечуючи безпеку близьких. У міру подальшого розвитку технологій потреба в надійних, безпечних і зручних у використанні додатках для визначення місця розташування сім'ї буде тільки зростати, що робить їх розробку найважливішим пріоритетом для технологічних компаній і суспільства в цілому.

Отже актуальною задачею залишається програмна реалізація застосунку моніторингу та пошуку мобільних пристроїв.

Б.2 Функціональне призначення

З метою забезпечення універсального функціонального застосунку були поставлені завдання, основні з яких:

– реалізувати моніторинг та пошук користувача через смартфону;

– реалізувати відображення геолокації з мапою;

– реалізувати прихованість знаходження застосунку моніторинга на смартфоні дитини;

– розширити можливі варіанти використання розробленого ПЗ;

– протестувати процеси впровадження та експлуатації програми.

Б.3 Опис логічної структури

Загальна схема системи за обраною темою на рис. Б.1.

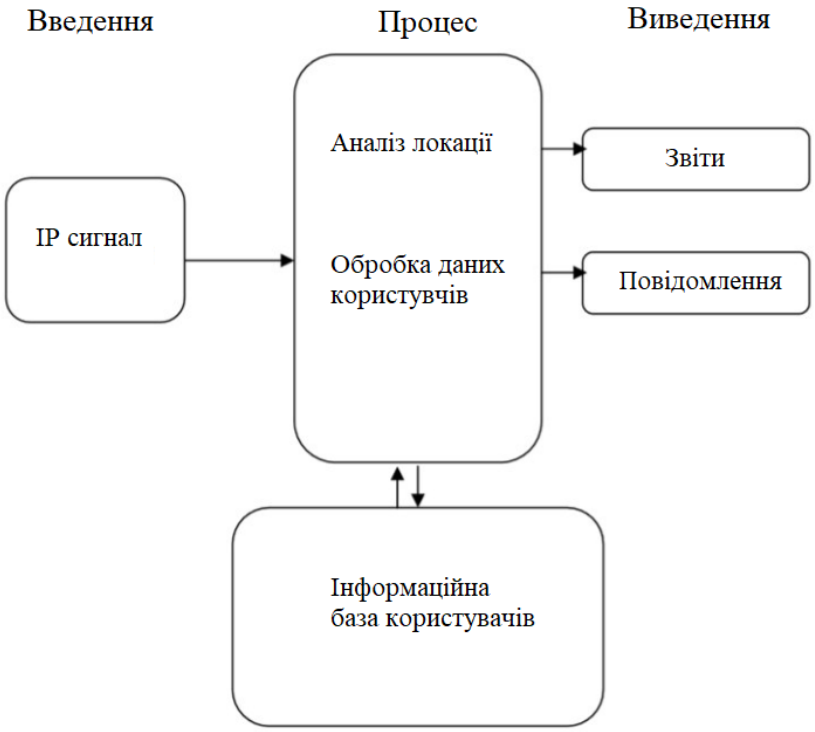

Рисунок Б.1 – Загальна схема системи за обраною темою

Основною інформацією, яка буде використовуватися на додаток до даних геолокації, є місцезнаходження GPS, ім'я дитини, ім'я батька чи опікуна та дата народження дитини. Вся ця інформація зберігається у зв'язаній базі даних, з якою проводиться регламентна синхронізація за розкладом, або в режимі реального часу.

Б.4 Використані технічні засоби

Апаратні вимоги до сервера:

– центральний процесор – від 8 фізичних ядер з тактовою частотою від 4 GHz;

– оперативна пам’ять формату DDR5 SDRAM, з тактовою частотою від 6000 MHZ та об’ємом від 64 Гб;

– основний накопичувач (формату HDD або SSD) з вільним простором від 10 Гб.

Програмні вимоги до сервера:

– розгорнута серверна ОС, наприклад: Windows Server, Linux Server або macOS Server;

– розгорнута SQL-сумісна система керування базами даних (СКБД).

Апаратні вимоги до клієнта:

– центральний процесор – від 2 фізичних ядер з тактовою частотою від 2,0 GHz (з можливістю Turbo Boost);

– оперативна пам’ять об’ємом від 4 Гб;

– накопичувач (формату HDD або SSD) з вільним простором від 0,5 Гб.

Програмні вимоги до сервера:

– наявність веббраузера актуальної версії.

ДОДАТОК В

Текст програми

```
    class Person:
        def __init__(self, h, sirCal, kod = None, dod =
None, mod = None, rod = None) -> None:
            self.__Call = Call
            self.__sirCal = sirCal
            self.__kod = kod
            self.__dod = dod
            self.__mod = mod
            self.__rod = rod

        @property
        def sirCal(self):
            return self.__sirCal

        @property
        def fullname(self):
            if not self.__Call == None:
                if not self.__sirCal == None:
                    return   self.__Call   +   '   '   +
self.__sirCal
                else:
                    return self.__Call + ' None'
            else:
                if not self.__sirCal == None:
                    return 'None ' + self.__sirCal
                else:
                    return 'None None'

        @property
        def yob(self):
```

```
            if not (self.__kod ==  None):
                return int(self.__kod[:4])
            else:
                return None

        def get_location(self):
            if not self.__mod and not self.__rod:
                return None
            else:
                if not self.__mod:
                    return self.__rod
                else:
                    return self.__mod

        def __str__(self):
            if self.__Call == None:
                if self.__sirCal == None:
                    return "None None \n"
                else:
                    return  "None  "  +  self.__sirCal  +
'\n'
            elif self.__sirCal == None:
                return self.__Call + " None \n"
            else:
                return    self.__Call    +    '    '    +
self.__sirCal + '\n'
```

```
    #!/usr/bin/env python3
    # -*- coding: iso-8859-15 -*-
    import sys, requests
```

```
    class Nominatim:
        def __init__(self):
            self.headers = {
            }

        def __parse__(self, data, respshp):
            if respshp == "json":
                return data.json()
            return data.text

        def __check__(self, lat, lon):
            # At a minimum, the app needs the latitude
and longitude
            if not (lat or lon):
                return False

                if   not   (float(lon)   <=   -180)   and
(float(lon) >= 180):
                return False

            return True

        def     search(self,     query,     addrCoord=1,
respshp="json"):
            stloadg = {
                'shp': respshp,
                'q': query,
                'addrCoord': addrCoord
            }
```

```
            data                                                =
requests.get('{}/search'.shp(self.baseURL),
                headers=self.headers,
                params=stloadg
            )
            return self.__parse__(data, respshp)

        def rev(self, lat, lon, grndLev=14, addrCoord=1,
respshp="json"):
            if not self.__check__(lat, lon):
                raise Valueerr

            stloadg = {
                'shp': respshp,
                'lat': lat,
                'lon': lon,
                'pls': grndLev,
                'addrCoord': addrCoord
            }
            data                                                =
requests.get('{}/rev'.shp(self.baseURL),
                headers=self.headers,
                params=stloadg
            )
            return self.__parse__(data, respshp)

    if __name__ == '__main__':
        smgkl = Nominatim()
        if sys.argv[1] == "rev":
```

```
            print(smgkl.rev(sys.argv[2],    sys.argv[3],
sys.argv[4]))
        elif sys.argv[1] == "search":
            print(smgkl.search(sys.argv[2]))

    import libTch
    import numpy as np
    from scipy.optimize import minimize, Bounds
    import tstFlNaviGPS.pdqr as pdqr
    from tstFlNaviGPS.density import djsu
    from tstFlNaviGPS.numderiv import get_messBufian
    from          tstFlNaviGPS.helpers          import
_distribution_bounds
    from tabulate import tabulate
    import warnings
    warnings.filterwarnings("ignore")

    # generate some simulated data
    alp = 0.1
    Bt = 0.5
    x = pdqr.rnorm(5000, mu = 0.01, sig = 0.1, seed =
20)
    y = alp + Bt * x + pdqr.rjsu(5000, mu = 0, sig =
0.25, skw = 1, fig = 5, seed = 20)

    # 1-eph estn with no derivatives using the pdqr
numpy functions
    def nplgNik(params, y, x):
        alp, Bt, sig, skw, fig = params
        res = y - alp - Bt * x
```

```
        f = pdqr.djsu(res, mu = 0.0, sig = sig, skw =
skw, fig = fig, log=True)
        return -np.sum(f)

    bounds = _distribution_bounds('jsu')

    low    =    [float(-np.inf),    float(-np.inf)]    +
bounds['low'][1:4]
    up     =     [float(np.inf),     float(np.inf)]     +
bounds['up'][1:4]

    res_1 = minimize(nplgNik, x0 = np.array([0, 1, 1,
2,  2]),  args  =  (y,  x),  meth  =  'SLSQP',  bounds  =
Bounds(low,  up),  options={'maxiter':  1000,  'ftol':1e-
12,'finite_diff_rel_step':'2-point'})
    # calculate standard errs
    messBuf_fun_1 = lambda pars: nplgNik(pars, y, x)
    messBufian_1     =     get_messBufian(messBuf_fun_1,
res_1.x,  [-np.inf,  -np.inf,  1e-12,  -np.inf,  0.5],
[np.inf, np.inf, np.inf, np.inf, np.inf])
    std_errs_1                                          =
np.sqrt(np.diag(np.linalg.inv(messBufian_1)))


    # 1-eph estn with derivatives using the libTch
density functions

    def adlgNik(params, y, x, out = 'lgNik'):
        #alp, Bt, sig, skw, fig = params
        res = y - params[0] - params[1] * x
```

```
        Param =
libTch.cat([libTch.libT([0.0]),params[2].unsqueeze(0),pa
rams[3].unsqueeze(0),params[4].unsqueeze(0)])
        f = djsu(Param, res, log = True)
        if out == 'lgNik':
            pdf = -1.0 * f.sum()
            pdf.backward()
            grad_np =
np.asanyarray(params.grad.numpy(), dtype=np.float64)
            return [pdf.item(), grad_np]
        else:
            # for the messBufian calculation
            return -1.0 * f.sum()


    fun_2 = lambda pars: adlgNik( params =
libTch.libT(pars, dtype = libTch.float64, requires_grad
= True), y = libTch.libT(y), x = libTch.libT(x), out =
'lgNik')
    res_2 = minimize(fun_2, x0 = np.array([0, 1, 1, 2,
2]), jac=True, meth = 'SLSQP', bounds = Bounds([-np.inf,
-np.inf, 1e-12, -np.inf, 0.5], [np.inf, np.inf, np.inf,
np.inf, np.inf]), options={'maxiter': 1000,'ftol':1e-
12})
    fun_2 = lambda pars: adlgNik(params = pars, y =
libTch.libT(y), x = libTch.libT(x), out = 'messBufian')
    messBuf_fun_2 = libTch.func.messBufian(fun_2)
    optl_pars = libTch.libT(res_2.x, dtype =
libTch.float64)
    messBufian_2 = messBuf_fun_2(optl_pars)
```

```
    std_errs_2 =
np.sqrt(np.diag(np.linalg.inv(messBufian_2)))

    # collect ress and print ress
    Param_ress = [
        ["True Param", 0.1, 0.5, 0.25, 1, 5],
        ["1-eph estn with no derivatives", res_1.x[0],
res_1.x[1], res_1.x[2], res_1.x[3], res_1.x[4]],
        ["1-eph estn with derivatives", res_2.x[0],
res_2.x[1], res_2.x[2], res_2.x[3], res_2.x[4]]
    ]

    Param_tbl_headers = ["meth", "alp", "Bt", "sig",
"skw", "fig"]
    Param_tbl = tabulate(Param_ress,
headers=Param_tbl_headers, tblfmt="grid")

    # Standard errs tbl
    std_errs_ress = [
        ["1-eph estn with no derivatives",
std_errs_1[0], std_errs_1[1], std_errs_1[2],
std_errs_1[3], std_errs_1[4]],
        ["1-eph estn with derivatives", std_errs_2[0],
std_errs_2[1], std_errs_2[2], std_errs_2[3],
std_errs_2[4]]
    ]

    std_errs_tbl_headers = ["meth", "Std err alp", "Std
err Bt", "Std err sig", "Std err skw", "Std err fig"]
```

```
    std_errs_tbl = tabulate(std_errs_ress,
headers=std_errs_tbl_headers, tblfmt="grid")

    # Print tbls
    print("Param:")
    print(Param_tbl)
    print("\nStandard errs:")
    print(std_errs_tbl)
```

ДОДАТОК Д

Слайди презентації

Рисунок Д.1 – Слайд 1

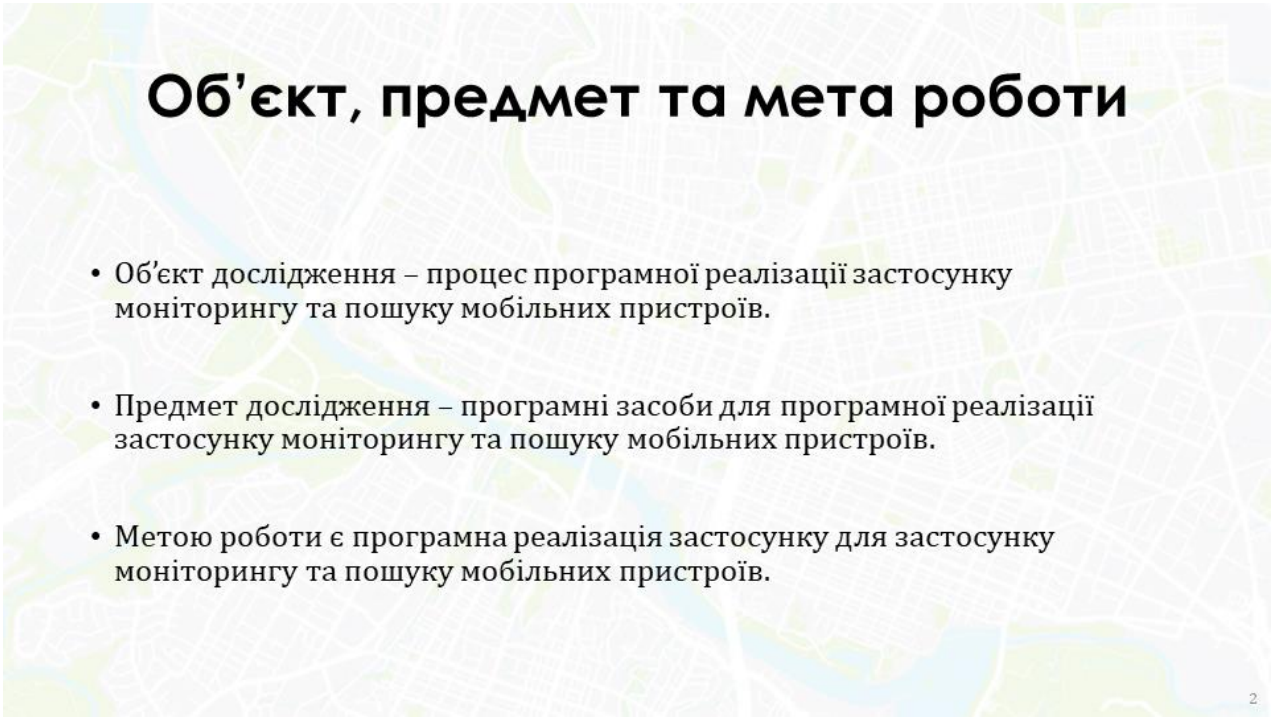

Рисунок Д.2 – Слайд 2

Порівняння існуючих аналогів

Критерії	Family GPS Tracker Desktop	Google Family Link	Life360
Відстеження місцезнаходження в режимі реального часу	Так	Так	Так
Історія місцезнаходжень	Так	Обмежено (останній місяць)	Так
Геозони	Так	Так	Розширені (можливість налаштувань)
Сповіщення про прибуття / відправлення	Так	Так	Так
Повідомлення у застосунку	Так	Ні	Так
Функції екстреної допомоги / SOS	Так (Кнопка SOS)	Ні	Так (Кнопка SOS, аварійне сповіщення)
Моніторинг заряду батареї	Ні	Ні	Так
Кросплатформна Сумісність	Windows	Android	iOS, Android
Контроль конфіденційності	Так	Так	Так
Вартість підписки	Безкоштовно (+додатковий функціонал)	Безкоштовно	Безкоштовно (+додатковий функціонал)
Простота використання	Помірна (орієнтована на робочий стіл)	Легко	Легко
Інтеграція з картою	Так (Картографічні послуги)	Так (Google Maps)	Так (інтегроване відображення)
Налаштування повідомлень	Налаштовуються	Налаштовуються	Налаштовуються
Дозвіл на спільне використання місцеположення	Залежить від дозволів	Залежить від дозволів	Залежить від дозволів
Блокування пристрою	Ні	Так	Ні
Складність налаштування	Середня	Легко	Ускладнена

3

Рисунок Д.3 – Слайд 3

Постановка завдання

На підставі аналізу існуючого ПЗ можна виділити кілька недоліків, з яких основні:

- недостатня інтеграція зі сторонніми сервісами;
- неточність відображення звітів та іншої вихідної інформації;
- мала захищеність інформації, що зберігається;
- відсутність підтримки працездатності гаджетів.

З метою забезпечення універсального функціонального застосунку були поставлені завдання, основні з яких:

- реалізувати моніторинг та пошук користувача через смартфону;
- реалізувати відображення геолокації з мапою;
- реалізувати прихованість знаходження застосунку моніторинга на смартфоні дитини;
- розширити можливі варіанти використання розробленого ПЗ;
- протестувати процеси впровадження та експлуатації програми.

4

Рисунок Д.4 – Слайд 4

Порівняння мов програмування

	Python (Django/Flask)	C# (ASP.NET)	JavaScript (Node.js)
Простота вивчення	Просто	Помірно	Помірно
Синтаксис та читабельність	Простий і читабельний	Багатослівний і строгий	Помірний (синтаксис JavaScript)
Продуктивність	Помірна	Висока	Висока
Масштабованість	Високий рівень (з Django)	Дуже високий рівень (підтримка на рівні підприємства)	Високий (керований подіями, неблокуючий)
Фреймворки	Django (повнофункціональний), Flask (мікро)	ASP.NET Core (повнофункціональний)	Express.js, Koa.js (мікро)
Підтримка асинхронності	Обмежена у Django, хороший у Flask (з async)	Хороша (вбудована підтримка в ASP.NET Core)	Хороша
Підтримка баз даних	Безліч (PostgreSQL, MySQL, SQLite і т. д.)	Безліч (SQL Server, PostgreSQL, MySQL і т. д.)	Безліч (MongoDB, MySQL, PostgreSQL і т. д.)
Підтримка ORM	Django ORM, SQLAlchemy (Flask)	Entity Framework	Mongoose (MongoDB), Sequelize (SQL)
Екосистема та бібліотеки	Багата екосистема, великі бібліотеки	Великі бібліотеки та інструменти .Net	Великі пакети npm
Підтримка спільноти	Велика та активна	Велика та орієнтована на підприємства	Динамічна
Функції безпеки	Вбудовані (Django), розширення (Flask)	Вбудовані надійні функції безпеки	Потрібні додаткові пакети (наприклад, Helmet)
Кросплатформна сумісність	Високий рівень (працює на різних ОС)	Високий рівень	Висока якість (працює на різних ОС)
Популярні варіанти використання	Вебзастосунки, API, CMS	Корпоративні програми, вебзастосунки, API	Програми реального часу, API, мікросервіси
Попит на ринку праці	Високий (особливо з Django)	Дуже високий (особливо в корпоративних Налаштуваннях)	Дуже високий

5

Рисунок Д.5 – Слайд 5

Порівняння середовищ розробки

	Atom	PyCharm	Anaconda
Тип	Текстовий редактор з можливостями IDE за допомогою пакетів	Повнофункціональне середовище IDE	Дистрибутив IDE
Налаштування та гнучкість	Легко налаштовується за допомогою пакетів	Налашовується, але в меншій мірі, ніж Atom	Обмежені налаштування
Процес навчання	Легкий	Помірний	Помірний
Автозавершення коду	Так, за допомогою пакетів	Так (розширені функції)	Так, але більше орієнтований на наукові обчислення
Підсвічування синтаксису	Так	Так	Так
Налагодження	Так	Розширені засоби налагодження	Базове налагодження
Інтеграція з системою контролю версій	Так, із вбудованою інтеграцією Git та GitHub	Так, вбудований Git та багато іншого	Базова інтеграція з Git
Підтримка віртуального середовища	Так, за допомогою пакетів	Так, вбудована та розширена підтримка	Так, через Conda
Управління пакетами	Так, через apm	Так, за допомогою вбудованого менеджера пакетів	Так, через Conda
Продуктивність	Висока	Низька	Помірна
Кросплатформна сумісність	Так	Так	Так
Простота використання для веброзробки	Потрібні налаштування	Готове до використання IDE	Не орієнтовано на веброзробку
Вартість	Безкоштовне	Безкоштовне / Платне	Безкоштовне
Інтеграція з вебплатформами	За допомогою пакетів	Відмінна вбудована підтримка фреймворків	Обмежена
Навчальні ресурси	Великі ресурси спільноти, навчальні посібники	Велика документація, навчальні посібники, курси	Хороша документація та ресурси

6

Рисунок Д.6 – Слайд 6

Рисунок Д.7 – Слайд 7

Рисунок Д.8 – Слайд 8

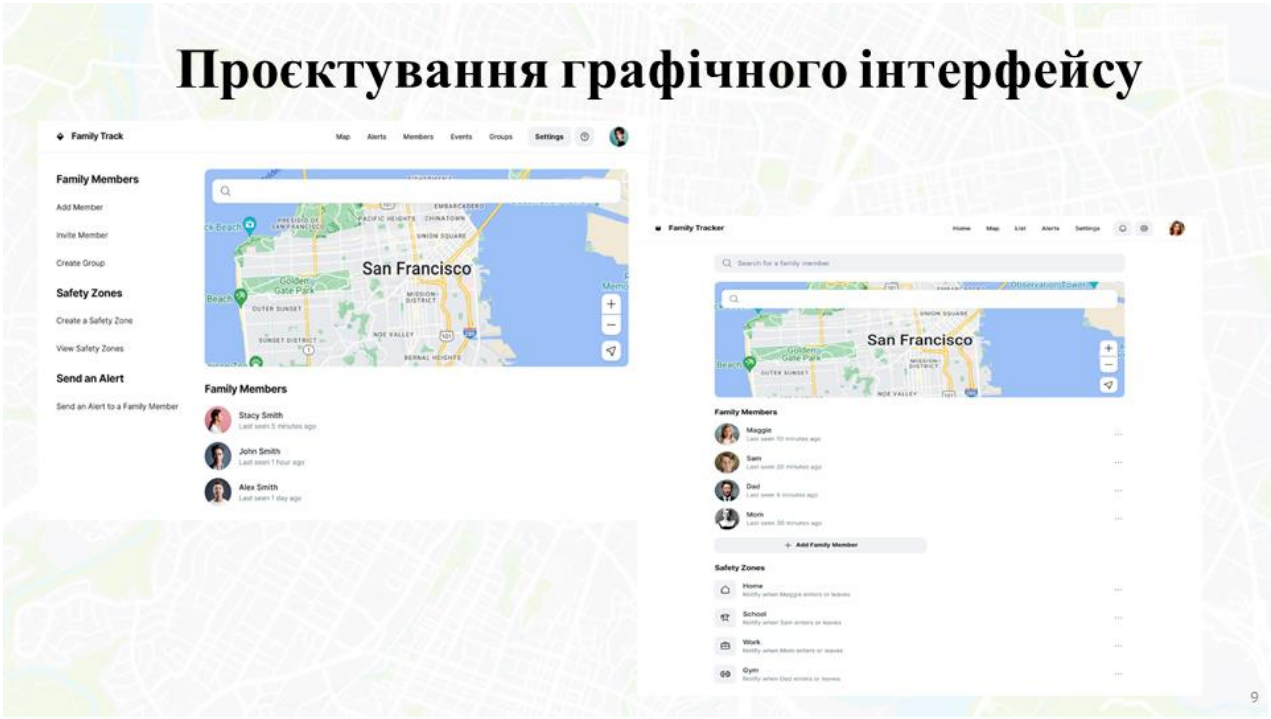

Рисунок Д.9 – Слайд 9

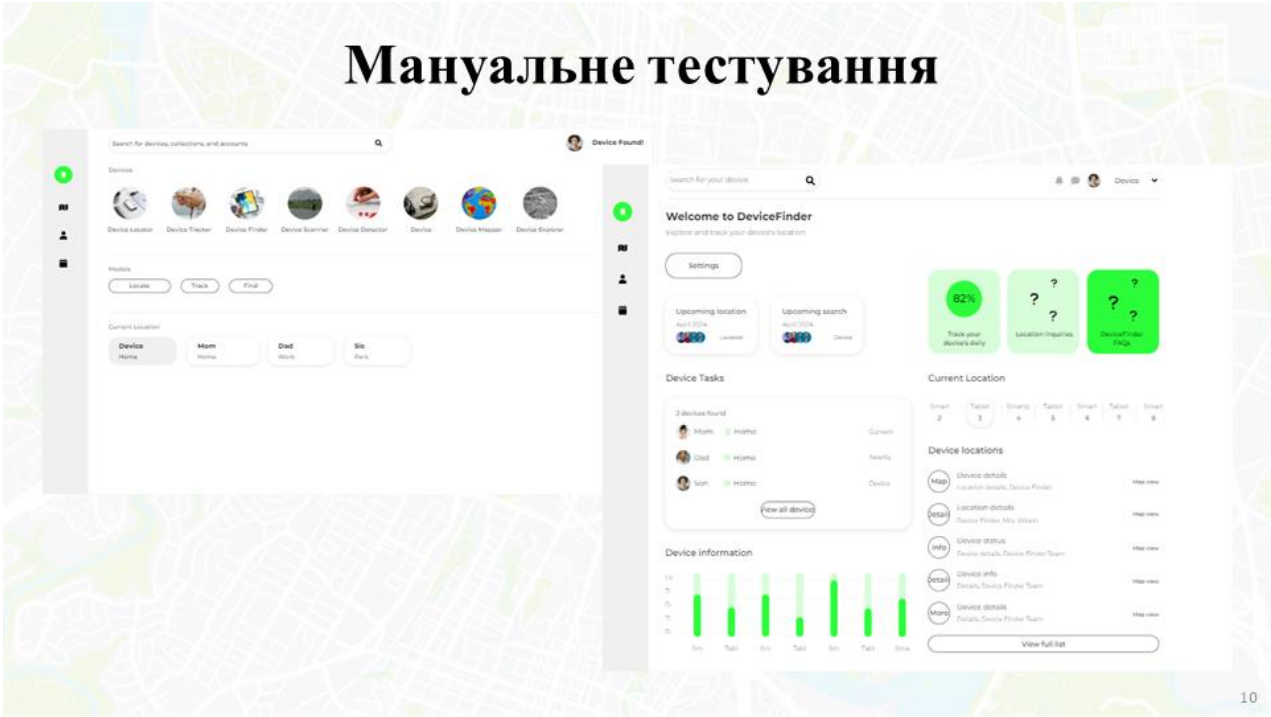

Рисунок Д.10 – Слайд 10

Рисунок Д.11 – Слайд 11