

ПРИНЦИПИ ПОБУДОВИ ГЕОІНФОРМАЦІЙНОЇ СИСТЕМИ ПОШУКУ ЛЮДЕЙ

© Пастернак І., 2017

Розглянуто геоінформаційну систему (ГІС) пошуку людей, за допомогою якої користувач цього сервісу має можливість знайти людину, яка знаходиться поряд з нею на карті, та розпочати спілкування. Запропоновано варіант реалізації геоінформаційної системи пошуку людей. Описано можливості інструментальних ГІС та технології клієнт-сервер і запропонований алгоритм геоінформаційної системи пошуку людей. Запропоновано алгоритм представлення інформації про користувача з використанням маркерів, які розташовуються на карті. Проаналізовано програмне забезпечення клієнта ГІС, яке може використовувати запуск програм на віддалених робочих місцях чи серверах.

Ключові слова: клієнт, сервер, геолокація, позиціонування.

In the article the Geographic Information System (GIS) finding people through which a user of this service has the ability to find the person who is next to him on the map and start communication. An embodiment of a geographic information system finding people. Described opportunities instrumental GIS and client-server technology and geographic information systems запропоновану алгоритм finding people. The algorithm representation statement using markers, which are located on the map. The analysis GIS software client that can be used launch applications on remote servers or workplace.

Keywords: client, server, geolocation, positioning.

Вступ

Розглянуто геоінформаційну систему (ГІС) пошуку людей, за допомогою якої користувач цього сервісу має можливість знайти людину, яка знаходиться поряд з нею на карті, та розпочати спілкування.

Соціальної мережі набирають стрімку популярність, сьогодні багато людей не можуть уявити своє життя без певних з них, одним із видів є геосоцмережа, яку і було розроблено в цьому проєкті. В ній користувачі залишають дані про своє місцезнаходження, що дає змогу об'єднувати і координувати дії користувачів на підставі того, які люди присутні в тих чи інших місцях, або які події відбуваються в цих місцях.

Зважаючи на те, що мережа Internet розвивається стрімкими темпами, ця соціальна мережа є актуальною сьогодні і може використовуватись у повсякденному житті.

Аналіз останніх джерел та публікацій

Сучасні тенденції розвитку інформаційних систем значною мірою зумовлені необхідністю підвищення швидкості доступу кінцевого користувача – фахівця до потрібної інформації. Сьогодні інформаційні системи у фінансових установах основані на концепції відкритих систем, невід'ємною складовою яких є технологія клієнт/сервер. Слід зазначити, що свого часу централізовану обробку даних змінила обробка даних у режимі розподілу часу на центральному комп'ютері. Згодом процедури обробки та доступу до даних почали розділяється між робочими станціями, що зв'язані з центральним комп'ютером. Тому початково поняття клієнт/сервер означало використання

персональних комп'ютерів клієнтів, об'єднаних локальними чи глобальними мережами з центральним комп'ютером або сервером

Принцип використання геоінформаційної системи полягає в тому, що користувач, використовуючи або портативний комп'ютер, або мобільний пристрій, реєструється на сервісі і відзначається в тих місцях, в яких буває: кафе, магазини, офіси та ін.

Поняття геоінформаційних систем увійшло в ужиток у 2000 році, коли було розроблено першу таку систему "Dodgeball", в якій користувачі залишають дані про своє місцезнаходження, що дає змогу об'єднувати і координувати дії користувачів на підставі того, які люди присутні в тих чи інших місцях, або які події відбуваються в цих місцях. Геолокація за допомогою комп'ютерів може використовувати або IP. У мобільних пристроях геолокація реалізується за допомогою GPS, текстової інформації, або LBS.

Постановка завдання

Розробити та проаналізувати геоінформаційну систему (ГІС) пошуку людей як сервіс, з використанням якого користувач сервісу має можливість знайти людину, яка знаходиться поряд з нею на карті, та розпочати спілкування.

Основні результати досліджень

Засоби побудови ГІС за допомогою глобальних мереж

ГІС використовується для місцезнаходження об'єктів на карті, програмно це реалізується так. Браузер може обчислити місцезнаходження людини, він не повідомить його веб-сайту без явного на те дозволу. Веб-сторінка у браузері може отримати дані про місцезнаходження відвідувача. Відвідувач може дозволити або ж не дозволити надавати ці дані. Така поведінка браузера Chrome не є власною ініціативою його розробників, а офіційним правилом стандарту геолокації, що вимагає, щоб кожен веб-сайт, який намагається отримати дані про місцезнаходження людини, отримав для цього його дозвіл. Щоб обчислити розташування відвідувача, браузер заручається допомогою постачальника послуг позиціонування (location provider), наприклад, для Chrome це сервіс Google Location Services.

У разі використання настільного комп'ютера з фіксованим (тобто провідним) підключенням до Інтернету, дані з комп'ютера користувача або локальної мережі направляються (через кабель, виділену лінію або комутуюче телефонне з'єднання) на мережевий апаратний пристрій, який, своєю чергою, направляє їх в Інтернет. Цей апаратний пристрій має однозначну IP-адресу, тобто числовий код, який ідентифікує цей пристрій для інших учасників мережі. Цей пристрій також має MAC-адресу. Постачальник місця розташування використовує ці два типи інформації для визначення географічного розташування відвідувача веб-сторінки. Спочатку обчислюють IP-адресу пристрою, через яке здійснюється підключення, а потім визначають його фізичну адресу. Природно, за такого непрямого підходу можна визначити точне місцезнаходження не користувача, а лише його інтернет-провайдера. Метод визначення місця розташування за допомогою IP-адреси є неточним способом геолокації [1].

Якщо користувач зайшов на сторінку з нетбука або смартфона за бездротовим підключенням, використовуються найближчі точки бездротового доступу. В ідеальному випадку ГІС місця розташування вибирає дані з величезної бази даних, щоб визначити точне місцезнаходження цих точок доступу, а потім використовує отриману інформацію, щоб визначити місце розташування відвідувача методом тріангуляції.

А у випадку підключення з мобільного телефону постачальник місця розташування застосовує подібний метод тріангуляції, але використовує сигнали від різних антен мобільного зв'язку. Ця швидка, порівняно ефективна процедура зазвичай дозволяє визначити місце розташування відвідувача з точністю до 1 км.

Багато мобільних пристроїв мають спеціальні апаратні засоби GPS глобальна система навігації та визначення місця знаходження (GPS), що дозволяє визначити місце розташування необхідних пристроїв з точністю всього лише до кількох метрів. Але цей метод геолокації має свій

недолік – він повільніший і споживає більше енергії, що важливо для пристроїв, що працюють на акумуляторах. Крім цього, він не особливо добре працює в містах з великими і високими будинками з причини відображення сигналу від будівель.

Звичайно ж, можливі й інші способи визначення місця розташування. Ніщо не перешкоджає постачальнику місця розташування використовувати для цього інші джерела інформації, такі як дані від RFID-пристроїв (Radio Frequency Identification – радіочастотна ідентифікація), дані від пристроїв Bluetooth, файли cookies з інформацією від картографічного сайту на зразок Google Maps і т.п. З усього цього можна зробити висновок: незалежно від способу підключення до Інтернету, навіть якщо відвідувач веб-сайту користується настільним комп'ютером, можливість геолокації дозволяє визначити його місцезнаходження з більшою або меншою точністю. А у випадку, якщо користувач вступає до мережі Internet з мобільного телефону або з пристрою, оснащеного апаратним забезпеченням GPS, його місце розташування може бути визначено з більшою точністю.

Клієнт в ГІС

З технічного боку як клієнт можуть бути використані звичайні персональні комп'ютери різних фірм, як, наприклад, IBM, HP, Compaq, Dell тощо, до яких не висуваються специфічні вимоги. Основні функції, які мають забезпечити ці технічні засоби – це обслуговування введення-виведення інформації користувача, локальне збереження особистих даних, використання периферійного устаткування та управління взаємодією з віддаленими ресурсами серверів.

Програмне забезпечення клієнта ГІС може використовувати запуск програм на віддалених робочих місцях чи серверах. Це дозволяє не тільки зберігати фрагменти (модулі) прикладних програм на різних комп'ютерах для розподілу їх між користувачами, але й виконувати їх на різних комп'ютерах. Робоча станція клієнта може не мати жорсткого диска, з якого завантажуються системні програми. В цьому випадку використовується віддалене завантаження операційної системи з віддаленого сервера в операційну пам'ять комп'ютера з використанням мережі передавання даних. Програмне забезпечення клієнта ГІС може надавати користувачу сервіси прямого мережевого обміну з допомогою відомих протоколів: TCP/IP, IPX, NetBios, Ethernet, X.25, тощо.

Користувач має програмні засоби для розроблення прикладних програм за технологією клієнт/сервер та засоби взаємодії з віддаленими базами даних. Сервери баз даних зберігають та оброблюють дані, приймаючи від користувача запити та повертаючи йому результат обробки.

Сервер у ГІС

Як сервер ГІС використовують потужні ЕОМ, що мають великий дисковий простір і процесори з великою швидкістю. Важлива роль серверу ГІС полягає в управлінні клієнтами, які спільно користуються ресурсами системи у певний момент часу: принтерами, БД, зовнішньою пам'яттю, програмами та ін. За функціями сервери поділяють на файл-сервер, обчислювальний сервер, принт-сервер, комунікаційний сервер тощо.

Однчасний доступ багатьох користувачів до інтегрованої бази даних (БД) реалізується в концепції “клієнт-сервер”, згідно з якою серверу належить активніша роль. Запит на оброблення даних посилає клієнт по мережі на сервер. На сервері шукають дані та обробляють їх засобами системи керування БД. Оброблені дані передаються по мережі від сервера до клієнта ГІС. Специфікою архітектури “клієнт-сервер” є використання мови структурованих запитів SQL (Structured Queries Language) до БД, що забезпечує роботу зі спільними даними з різнотипних додатків у мережі [2].

Основна ідея концепції “клієнт-сервер” полягає в тому, щоб сервери розмістити на потужних ЕОМ, а додатки клієнтів — на менш потужних. Завдяки цьому будуть задіяні ресурси більш потужного серверу і менш потужних ЕОМ клієнтів. Введення-виведення до бази ґрунтується не на фізичному, а на логічному дробленні даних, тобто сервер ГІС відправляє клієнтам не повну копію бази, а тільки логічно необхідні порції. Завдяки цьому скорочується трафік мережі – потік повідомлень. Сервер обробляє запити клієнтів, вибирає потрібні дані з БД, посилає їх клієнтам по мережі, поновлює інформацію, забезпечує цілісність і збереження даних.

Концепція “клієнт-сервер” дала змогу поєднати кращі якості ГІС з одним користувачем (високий рівень діалогової підтримки, дружній інтерфейс, низька ціна) з перевагами великих комп’ютерних систем (підтримка цілісності, захист даних, багатозадачність).

Моделі взаємодії “клієнт-сервер”

У своєму розвитку технології “клієнт-сервер” пройшли кілька етапів, тому є різні моделі технології. Їх реалізація заснована на поділі структури СКБД на три компоненти:

- введення і відображення даних (інтерфейс із користувачем);
- прикладний компонент (запити, події, правила, процедури та функції, які характерні для цієї предметної області);
- функції керування ресурсами (файловою системою, базою даних тощо).

Тому в будь-якому додатку виділяють такі компоненти:

- компонент подання даних;
- прикладний компонент;
- компонент керування ресурсом.

Зв’язок між компонентами здійснюється за певними правилами, які називають “протокол взаємодії”.

Існують різні класифікації, але однією з найпоширеніших є використання чотирьох моделей технології “клієнт-сервер”:

1. Модель файлового сервера (File Server - FS).
2. Модель віддаленого доступу до даних (Remote Data Access - RDA).
3. Модель сервера БД (Data Base Server - DBS).
4. Модель сервера додатків (Application Server - AS).

На практиці зараз зазвичай використовують змішані підходи:

- найпростіші прикладні функції виконуються збереженими процедурами на сервері;
- складніші реалізуються на клієнті безпосередньо в прикладній програмі.

Зараз деякі постачальники комерційних СКБД оголосили про плани реалізації механізмів виконання збережених процедур з використанням мови Java. Це відповідає концепції “тонкого клієнта”, функцією якого залишається тільки відображення даних (модель віддаленого подання даних) [3,4].

Останнім часом також спостерігається тенденція до все більшого використання моделі розподіленого додатка. Характерною рисою таких додатків є логічний поділ додатка на дві та більше частин, кожна з яких можна виконувати на окремому комп’ютері. Виділені частини додатка взаємодіють одна з однією, обмінюючись повідомленнями в заздалегідь погодженому форматі. У цьому випадку дволанкова архітектура “клієнт-сервер” стає триланковою, а в деяких випадках вона може містити і більше ланок.

У задачах обробки інформації, основаних на системах баз даних, існують два варіанти розташування даних: локальний і віддалений. У першому випадку говорять про доступ до локальних даних, у другому – до віддалених даних. Локальні дані, як правило, розташовуються на жорсткому диску комп’ютера, на якому працює користувач, і знаходяться в монопольному керуванні цього користувача. Користувач при цьому працює автономно, не залежачи від інших користувачів та жодним чином не впливаючи на їх роботу. Дистанційні дані розташовуються поза комп’ютером користувача (користувачів) – на файловому сервері мережі або на спеціально виділеному для цих цілей комп’ютері.

Модель файлового сервера (File Server – FS) – це природне розширення персональних СКБД для підтримки багатокористувацького режиму і в цьому плані ще довго зберігатиме своє значення.

У RDA-моделі для обробки даних виділяється спеціальне ядро – так званий SQL-сервер, який бере на себе функції обробки запитів користувачів, іменованих тепер клієнтами. Сервер баз даних являє собою програму, яка виконується, як правило, на потужному комп’ютері. Додатки-клієнти посилають з робочих станцій запити на вибірку (вставляння, оновлення, видалення) даних. При цьому сервер виконує всю “брудну” роботу з відбору даних, відправляючи клієнтові тільки

необхідну “вижимку”. Якщо наведений вище приклад перебудувати з урахуванням клієнт-серверної архітектури, то додаток-клієнт “отримає” від сервера як результат список лише тих працівників, які беруть участь у заданому проекті, і не більше того. Такий підхід забезпечує вирішення трьох важливих завдань:

- зменшення навантаження на мережу;
- зменшення вимог до комп’ютерів-клієнтів;
- підвищення надійності та збереження логічної цілісності бази даних.

Тут додатки також виконуються переважно на робочих станціях.

Додаток містить модулі для організації діалогу з користувачем і бізнес-правила (транзакції). Ядро СКБД є загальним для всіх робочих станцій і функціонує на сервері. Оператори звернення до СКБД (SQL-оператори), закодовані в транзакції, не виконуються на робочій станції, а пересилаються для обробки на сервер. Ядро СКБД транслює запит і виконує його, звертаючись для цього до індексів та інших проміжних даних. Назад на робочу станцію передаються тільки результати обробки оператора.

Для поліпшення попередньої RDA-моделі в сучасних СКБД використовується модель сервера баз даних (DBS), в якій на сервері можуть запускатися так звані збережені процедури і тригери, які разом з ядром СКБД утворюють сервер бази даних.

До збережених процедур можна звертатися з додатків на робочих станціях. Це дає змогу скоротити розмір коду прикладної програми і зменшити потік SQL-операторів з робочої станції, оскільки групу необхідних SQL-запитів можна закодувати у збереженій процедурі. Тригери виконуються ядром СКБД перед або після оновлення (UPDATE, INSERT, DELETE) таблиці бази даних. Вони дають змогу автоматично підтримувати цілісність бази даних [5].

Програмна реалізація ГІС

Аналогом запропонованої геоінформаційної системи є додаток “Tinder”, який стрімко набирає популярність у США. Програму розроблено під платформи Android і Apple iOS, призначено для підбору локальних кандидатур для кола спілкування користувача відповідно із заданими параметрами.

Відмінності розробленої геоінформаційної системи від цього додатка визначаються трьома факторами (рис. 1). По-перше, для визначення місцезнаходження використовують інший алгоритм, за яким можна достатньо точно визначити координати користувача, який зайшов на сайт із персонального комп’ютера. По-друге, буде використовуватись інший алгоритм побудови розмови між двома або більше користувачами. По-третє, реалізація буде здійснюватись не на платформах Android і Apple iOS, а як окремий веб-сайт.

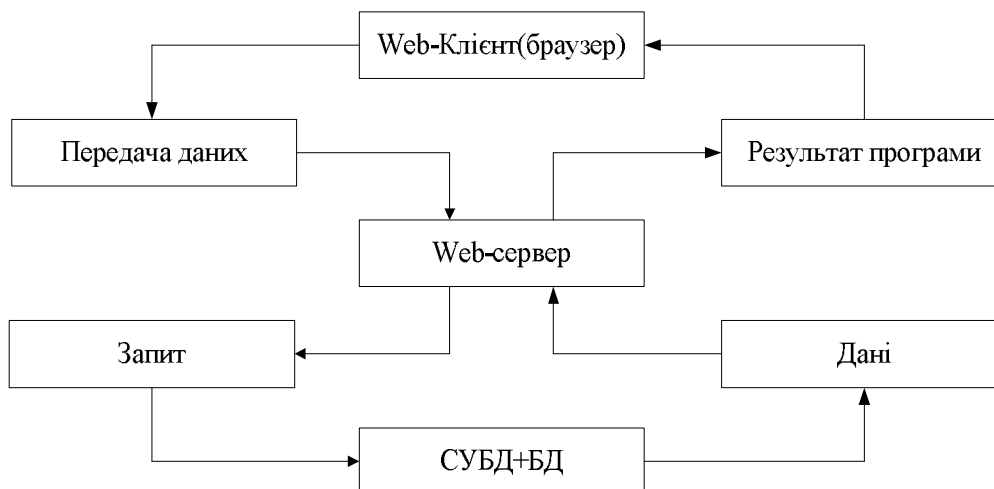


Рис. 1. Взаємодія “клієнт-сервер” у ГІС

З використанням технології “клієнт – сервер” побудовано картографічну ГІС з використанням СУБД. Геоінформаційну систему було розроблено як сервіс. Цей сервіс використовує карти Google для наведення об’єкта, який цікавить. Крім того, за допомогою карти можна легко виявити людей, які знаходяться поряд, для зв’язку із ними. Можливість доступу до БД за допомогою Web істотно змінила стратегії доступу до даних через те, що:

- кількість користувачів може не тільки істотно варіюватися, але і змінюватися без попередження. У звичайних клієнт/серверних додатках адміністратори можуть контролювати число користувачів і можливість їх доступу до даних;

- користувачів фізично неможливо визначити, вони можуть, не закривши однієї програми, підключитися до іншої, що не гарантує правильного відключення від програми;

- у клієнт-серверних додатках при зміні даних декількома користувачами використовуються блокування на рівні запису (песимістичний паралелізм). У Web -додатках застосовується принцип оптимістичного паралелізму, який передбачає фактично монопольний доступ для кожного користувача;

- посередником, який реалізує доступ від клієнта до БД, є спеціальна програма – Web-сервер, що забезпечує функціонування користувацького додатка, що виконує безпосередній запит до даних і повернення відповіді у вигляді HTML-документа.

Доступ до БД на стороні сервера:

- структурна схема доступу;
- структурна схема доступу до бази даних;
- web-сервер. Фактично Web-сервер містить кілька інших серверів, що реалізують необхідні протоколи: HTTP, FTP, NNTP, SMTP [6].

Механізм доступу до БД на стороні сервера реалізується завдяки наявності стандартизованих засобів:

- Підтримки діалогових форм на рівні гіпертекстового документа. З кожним документом пов’язаний URL (Uniform Resource Locator), для доступу до яких використовується об’єктно-орієнтований протокол HTTP;

- Можливості запуску серверних програм, які взаємодіють через стандартний інтерфейс CGI або прикладні інтерфейси Web-сервера.

Підтримку HTTP-протоколу в складі Web-сервера забезпечує HTTP-сервер. HTTP-клієнт/серверна взаємодія складається з одного обміну запит/відповідь. Це коротка транзакція між клієнтом і сервером. Фактично HTTP – це PRC (Remote Procedure Call) у Web поверх TCP/IP. HTTP не має стану, не пам’ятає попередніх станів і має важливу властивість передавати самоописувани дані.

HTTP-запит складатиметься із:

1. Структури запиту (HTTP-запит):

- Рядка запиту (request line);
- Одного або декількох необов’язкових полів заголовка (request header fields);
- необов’язковості тіла (entity body).

2. Рядок запиту складається з трьох текстових частин, розділених пробілами. Перша частина визначає метод (команду), який буде застосований до серверного ресурсу. Найзагальніший метод – це GET, який просить сервер надіслати копію ресурсу клієнту.

Друга частина визначає ім’я цільового ресурсу – URL, що складається з протоколу та імені домена сервера.

Третя частина визначає версію протоколу, використовуваного клієнтом, наприклад, HTTP/1.0.

3. Поле заголовка запиту передає додаткові дані про запит і самого клієнта сервера. Кожне поле заголовка складається з імені з двокрапкою і значенням поля.

4. Тіло запиту використовує клієнт для передавання даних серверу.

GET / path / file.html HTTP/1.0 // метод, ресурс і протокол

Accept: text / html // клієнт може працювати з текстом

User – agent: MacWeb // ім'я реалізації клієнта

Назва MIME – типу Тип даних

application / x – javascript JavaScript -модуль

image / jpeg JPEG – зображення

HTTP відповідь

1. HTTP – відповідь складається з:

- Рядка заголовка (статусу) відповіді;
- Однієї або декількох необов'язкових полів відповіді;
- необов'язкового тіла відповіді.

2. Рядок заголовка відповіді повертає версію HTTP, статус відповіді і пояснення повернутого статусу.

Версія – HTTP SP Статус – Код SP Фраза – Пояснення

3. Поля заголовка відповіді повертають відомості, що описують атрибути сервера і повернений HTML-документ для клієнта і містить ім'я, двокрапку і значення.

Public | Retry – After | Server | WWW – Authenticate | extension – header

4. Тіло відповіді містить власне HTML – документ, запитуваний клієнтом.

HTTP/1.0 200 OK // код 200 – відповідь успішна

Server: NCSA/1.3 // ім'я сервера

Mime_version: 1.0 // Mime версії 1.0

Context_type: text / html // повертається текст

Context_length: 2000 // повертається 2000 байт

Реалізація взаємодії БД із сервером

Для реалізації взаємодії БД із сервером використано сервер Apache, як сервер баз даних використовуються MySQL.

Web-сервер Apache – HTTP сервер з відкритими вихідними текстами для сучасних операційних систем, включаючи UNIX і Windows. Apache володіє великими можливостями конфігурації, є дуже продуктивним, підтримує всі відомі протоколи і використовується для організації більшості Web-серверів [7, 8].

Серед переваг Apache – висока швидкість обробки запитів і ефективність використання ресурсів обчислювальної установки, переносимість, підтримка з боку колективу розробників, стабільність і надійність, постійне оновлення, розширюваність і простота адміністрування.

Ефективність сервера визначається якістю його коду – приблизно 25 000 рядків мовою Сі. Головним завданням проекту Apache було і залишається створення високопродуктивного сервера, побудованого за модульною схемою, що дає змогу забезпечувати його компактність і налагодження відповідно до потреб того чи іншого Web-вузла.

Сьогодні Apache може працювати на всіх платформах, які так чи інакше використовуються в Internet: всі без винятку Unix-системи, системи на основі Windows і NT. При цьому сервер легко переноситься і на інші платформи.

Архітектура Apache робить його одним з найбільш гнучких серверів Мережі. Специфікація API-інтерфейсу сервера дає змогу нарощувати його потужність за рахунок додаткових модулів без зміни ядра системи.

MySQL є найбільш пристосованою для застосування в середовищі web СУБД (системою управління базами даних). Не секрет, що для виконання додатків клієнта на більшості хостинг-майданчиків провайдери надають невелику кількість ресурсів (як обчислювальних, так і дискових). Тому для цього застосування необхідна високоефективна СУБД, що володіє високою надійністю (більшість web-додатків і сайтів повинні працювати в режимі 24/7).

З усіх цих причин MySQL стала непорушним стандартом в області СУБД для web, а тепер в ній розвиваються можливості для використання її в будь-яких критичних бізнес-додатках, тобто конкурує на рівних з такими СУБД таких виробників, як Oracle, IBM, Microsoft і Sybase [9].

MySQL володіє багатьма перевагами, зокрема високою продуктивністю, низькою вартістю, простою конфігурування та вивчення, переносимістю і доступністю вихідного коду.

Докладніше згадані переваги розглянуто нижче.

- **Продуктивність.** MySQL, без сумніву, працює дуже швидко. Результати порівняльних тестів продуктивності, виконаних фірмою-виробником, можна подивитися на сторінці <http://web.mysql.com/benchmark.html>. Багато з цих порівняльних тестів показують, що MySQL працює на порядок швидше конкуруючих продуктів.

- **Низька вартість.** Пакет MySQL доступний безкоштовно відповідно до ліцензії на програмне забезпечення з відкритим вихідним кодом або, якщо це необхідно для програми, за невелику суму можна придбати комерційну ліцензію.

- **Простота використання.** У більшості сучасних баз даних використовується SQL. Якщо раніше працювали з іншими СУРБД, перехід до цієї системи не повинен викликати будь-яких труднощів. Установка MySQL настільки ж проста, як і встановлення багатьох аналогічних продуктів.

Принцип роботи з базою даних через веб-інтерфейс можна описати таким алгоритмом (на прикладі PHP + MySQL):

- користувач заповнює веб-форму і відправляє її PHP-оброблювачу, заданому в атрибуті action;

- веб-сервер приймає запит і викликає інтерпретатор PHP з ім'ям скрипта і даними веб-форми як параметрами;

- інтерпретатор PHP завантажує на виконання вказаний скрипт і передає йому дані з форми;

- у процесі виконання PHP-скрипта дані з форми підставляються в SQL-запит;

- з скрипта SQL-запит передається серверу СУБД MySQL;

- набір даних, отриманий в результаті виконання SQL-запиту, повертається зухвалому скрипту;

- PHP-скрипт використовує отримані з БД дані для динамічного формування веб-сторінки і повертає її веб-серверу, який, своєю чергою, віддає її клієнту.

База даних програми призначена для зберігання даних і швидкого доступу до них.. Ці дані у нашому випадку – це “Користувачі”, “Користувачі онлайн”, “Користувачі на карті”.

На рис. 2 ми бачимо інформацію для збереження про зареєстрованих користувачів, інформацію про яких використано для зберігання.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 ID	bigint(20)		UNSIGNED	Нет	Нет	AUTO_INCREMENT	Изменить Удалить Первичный
<input type="checkbox"/>	2 user_login	varchar(60)	utf8_general_ci		Нет			Изменить Удалить Первичный
<input type="checkbox"/>	3 user_pass	varchar(64)	utf8_general_ci		Нет			Изменить Удалить Первичный
<input type="checkbox"/>	4 user_nicename	varchar(50)	utf8_general_ci		Нет			Изменить Удалить Первичный
<input type="checkbox"/>	5 user_email	varchar(100)	utf8_general_ci		Нет			Изменить Удалить Первичный
<input type="checkbox"/>	6 user_url	varchar(100)	utf8_general_ci		Нет			Изменить Удалить Первичный
<input type="checkbox"/>	7 user_registered	datetime			Нет	0000-00-00 00:00:00		Изменить Удалить Первичный
<input type="checkbox"/>	8 user_activation_key	varchar(60)	utf8_general_ci		Нет			Изменить Удалить Первичный
<input type="checkbox"/>	9 user_status	int(11)			Нет	0		Изменить Удалить Первичный
<input type="checkbox"/>	10 display_name	varchar(250)	utf8_general_ci		Нет			Изменить Удалить Первичный

Рис. 2. Зареєстровані користувачі

На рис. 3 ми бачимо користувачів, онлайн-інформація про яких використовується для зберігання та виводиться на карті, при натисканні на маркер про користувачів, які в цей момент знаходяться на сайті.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Первичный Уникальный Индекс
2	user_type	varchar(50)	latin1_swedish_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс
3	user_id	int(11)			Нет	Нет		Изменить Удалить Первичный Уникальный Индекс
4	user_name	varchar(200)	latin1_swedish_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс

Рис. 3. Користувачі онлайн

На рис. 4 ми бачимо користувачів, місцезнаходження яких використовується для зберігання інформації, за допомогою якою формується маркер на карті.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Первичный Уникальный Индекс
2	userid	int(11)			Нет	Нет		Изменить Удалить Первичный Уникальный Индекс
3	ip	varchar(50)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс
4	country	varchar(100)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс
5	latitude	varchar(50)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс
6	longitude	varchar(50)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс
7	login	varchar(50)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс

Рис. 4. Користувачі на карті

Схему взаємодії серверного php-додатка з базою даних MySQL проілюстровано на рис. 5.

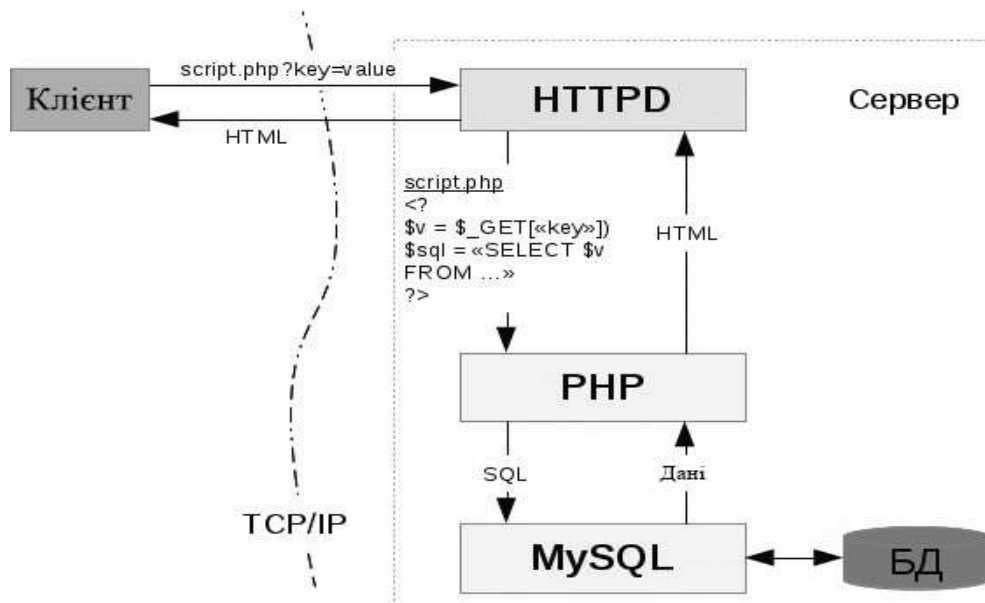


Рис. 5. Взаємодії серверного php-додатка з базою даних MySQL

Отже, для розроблення серверних php-додатків з підтримкою MySQL потрібна така конфігурація системи (на прикладі LAMP):

- Веб-сервер Apache;
- СУБД MySQL;
- PHP з модулями підтримки (бібліотеками) для Apache і MySQL.

У цьому проєкті для того, щоб зчитувати дані з таблиць БД, використовуються такі PHP-скрипти:

Скрипт для зчитування даних із таблиці `usermap`, дані з якої використовують для виведення користувачів на карту у вигляді маркерів.

Для того, щоб звернутись до бази даних, у цьому скрипті використано глобальну функцію Wordpress `wpdb`, після чого формується SQL-запит `SELECT`, для того щоб зчитати дані із стовпців `latitude`, `longitude` і `login`, що відповідають географічній широті і довготі розташування користувача та його реєстраційному імені. Отримані дані використовують для того, щоб розташувати маркер на карті.

Для того, щоб розташувати онлайн-користувачів на карті, ми використовуємо ідентичний скрипт, єдиною відмінністю є те, що зчитування відбувається із таблиці `useronline`.

Взаємодія сервера з клієнтом

Основну структуру взаємодії клієнта із сервером описано раніше. У цьому проєкті основним завданням є визначення географічного місцезнаходження користувача за допомогою технологій HTML5 та JavaScript. Геолокація складається з трьох методів об'єкта `navigator.geolocation`: `getCurrentPosition()`, `watchPosition()` і `clearWatch()`. Об'єкт `navigator` – це порівняно незначна частина JavaScript. Його кілька властивостей надають інформацію про поточний браузер та його можливості. Найкориснішою з них є властивість `navigator.userAgent`, яка надає інформаційний рядок, що містить докладні дані про браузер, його версію, а також операційної системи, в якій він виконується.

Для отримання місцезнаходження відвідувача викликається метод `getCurrentPosition()`. Але слід розуміти, що процес визначення місцезнаходження займає певний час, протягом якого жодний браузер не зупинить всю діяльність на сторінці, поки очікує ці дані. Тому метод `getCurrentPosition()` виконуються асинхронно, тобто після його запуску продовжується виконання наступного за ним коду. Коли визначення місцезнаходження завершиться, для обробки результатів активується інший фрагмент коду.

Можна очікувати, що завершення визначення місцезнаходження супроводжується сповіщенням, багато в чому подібним тому, як повідомляється про закінчення завантаження зображення або читання текстового файлу. Але мову JavaScript можна назвати якою завгодно, але тільки не послідовною. І при виклику методу `getCurrentPosition()` йому передається функція завершення (`completion function`).

При успішному виконанні методу `getCurrentPosition()` код отримує об'єкт `position`, який має дві властивості: `timestamp` (містить час виконання геолокації) і `coords` (містить географічні координати).

Але властивість `coords`, своєю чергою, є підоб'єктом об'єкта `position` і, крім властивостей `latitude` і `longitude`, що визначають географічні координати відвідувача, має ще кілька інших властивостей, що надають додаткову інформацію про місцезнаходження. Це такі властивості, як `altitude` (висота над рівнем моря), `heading` (напрямок руху) і `speed` (швидкість). Але сьогодні цих властивостей не підтримує жодний браузер.

Більш цікавою є властивість `accuracy`, яка вказує на точність певного місцезнаходження в метрах. (Це означає, що в міру зниження точності даних місцезнаходження значення властивості `accuracy` зростає, що може дещо збивати з пантелику). Наприклад, значення властивості `accuracy`, що дорівнює 2135 м, означає, що місцезнаходження користувача визначено в межах цієї відстані.

Тестування ГІС

Постановка експерименту - знайти місцезнаходження людини для зустрічі.

Тестування складається з кількох етапів:

- реєстрація;
- вивчення профілю користувача та його можливостей;

- визначення точного місцезнаходження;
- знаходження користувача на карті онлайн-користувачів;
- написання повідомлення користувачу.

Щоб мати можливість спілкуватись з людьми, які знаходяться поруч, насамперед потрібно зареєструватись. Форма реєстрації складається з таких полів:

- Ім'я користувача;
- E-mail;
- пароль;
- підтвердження пароля;
- ваше ім'я.

Після заповнення полів потрібно натиснути на кнопку “Реєстрація”. Якщо всі поля правильно заповнені, відкриється нова сторінка. Потім відбувається підтвердження електронної пошти, можна заповнити інші поля профіля і завантажити фото. Щоб зайти на сайт, потрібно натиснути на “Вхід”, який знаходиться поряд з “Реєстрацією”, і ввести свої дані. Після введення даних відбувається перенаправлення на головну сторінку, на якій є три кнопки: “Онлайн-користувачі”, “Зареєстровані користувачі”, “Точне місцезнаходження”. Щоб визначити місцезнаходження, потрібно натиснути кнопку “Точне місцезнаходження”.

Висновки

Розглянуто аналіз можливостей інструментальних ГІС та технології клієнт-сервер і описано алгоритм геоінформаційної системи пошуку людей, що реалізований за допомогою методів та об'єктів мови програмування JavaScript, які дають змогу визначити координати користувача сервісу. Було проаналізовано, що геолокація складається з трьох методів об'єкта navigator.geolocation: getCurrentPosition (), watchPosition () і clearWatch (), які брали за основу під час побудови цього сервісу.

Розроблено алгоритм представлення інформації про користувача. Цієї мети було досягнуто за допомогою маркерів, які розташовуються на карті, а саме коли користувач натисне на маркер, який вказує на певну людину, він одержить графічну і текстову інформацію про цього користувача. Завдяки цій інформації можна перейти до профілю користувача та написати йому повідомлення. Наявність цієї інформації на маркері значно полегшує і пришвидшує процес спілкування між користувачами, що є перевагою над аналогічними сервісами. Користувач також має можливість написати повідомлення на стіні, яке можуть бачити всі користувачі цього сервісу; ця особливість дає змогу спостерігати за активністю користувачів.

1. Ярих В. Б. *Якість цифрових карт // Інформаційний бюлетень ГІС-асоціації*. – М.: ГІС-асоціація. – 2002. – № 1–2. – С. 18–21.
2. Бусигін Б.С., Гаркуша І.М., Середінін Е. С., Гаєвенко А. Ю. *Інструментарій геоінформаційних систем: довідковий посібник*. – К.: ІПГ “СБ”, 2000. – 172 с.
3. Світличний О. О., Андерсон В. Н., Плотницький С.В. *Географічні інформаційні системи: технологія та додатки*. – Одеса: Астропринт, 1997. – 196 с.
4. Zeiler M. *Моделювання нашого світу. Посібник ESRI з проектування баз геоданих: пер. з англ.* – К.: ЗАТ ЕСОММ Со, 2004. – 254 с.
5. Мальцев В. А. *Програмний комплекс геостатистичного моделювання та оцінювання GST 3.02: Підручник і керівництво користувача*. – М., 1993. – 153 с.
6. Мітчелл Е. *Керівництво по ГІС-аналізу. Ч. 1: Просторові моделі і взаємозв'язки: Пер. з англ.* – К.: ЗАТ ЕСОММ Со; Стилос, 2000. – 198 с.
7. *Світовий ринок ГІС в 2001 р. // ARCREVIEW. Сучасні геоінформаційні технології*. – 2003. – № 1 (24). – С. 14.
8. *Атлас України. Пілотний проект електронної версії Національного атласу України / Інститут географії НАН України. ТОВ “Інтелектуальні системи ГЕО”*. – К., 2000.
9. Демерс М. Н. *Географічні інформаційні системи*. – М.: Изд-во СП Дата +, 1999. – 491 с.