

Технічні науки

УДК 004.942

**Рижов Антон Ігорович**

*студент Інституту комп'ютерних систем*

*Одеського національного політехнічного університету*

**Рыжов Антон Игоревич**

*студент Института компьютерных систем*

*Одесского национального политехнического университета*

**Ryzhov Anton**

*Student of the Institute of Computer Systems*

*Odessa National Polytechnic University*

**Науковий керівник:**

**Мартинюк Олександр Миколайович**

*кандидат технічних наук, доцент кафедри КІСМ*

*Одеський національний політехнічний університет*

**ДОСЛІДЖЕННЯ МОДЕЛІ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ**

**«ДОВІДНИК МАНДРІВНИКА»**

**ИССЛЕДОВАНИЕ МОДЕЛИ КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ**

**«СПРАВОЧНИК ПУТЕШЕСТВЕННИКА»**

**RESEARCH OF THE CLIENT-SERVER APPLICATION "GUIDE**

**TRAVELER"**

*Анотація. В дослідженні розглянуто аналоги розроблюваного додатку, розроблено модель яка має особливості – введені ділянки позицій та переходів для взаємодії додатку з сервісом Google, що надає користувачу можливість одразу отримувати карту – довідник, не застосовуючи інше*

програмне забезпечення, що дає економію внутрішньої пам'яті пристрою. Побудована модель була використана при проектуванні клієнт серверної системи що забезпечує карту-довідник. А також були розглянуті принципи структурної організації клієнт-серверних систем.

Спроектовано архітектуру системи її клієнтську та серверну частини, у вигляді UML діаграм визначена структура збереження даних користувача, та даних про об'єкти пошуку. Побудована мережа Петрі із графом досяжних маркувань, виконано аналіз та моделювання мережі. Також на основі граф схем алгоритмів було розроблено алгоритми необхідні для реалізації поставлених задач.

Розроблено систему яка реалізує додаток карту довідник яка може бути впроваджена в туристичній галузі для використання туристичними агентствами.

**Ключові слова:** клієнт-сервер, мережа Петрі, веб-сервіс, мережеві протоколи, карта-довідник, telegram-bot.

**Анотація.** В исследовании рассмотрены аналоги разрабатываемого приложения, разработана модель, которая имеет особенности - введены участки позиций и переходов для взаимодействия приложения с сервисом Google, предоставляет пользователю возможность сразу получать карту-справочник, не применяя другое программное обеспечение, дает экономию внутренней памяти устройства. Построенная модель была использована при проектировании клиент серверной системы, обеспечивающей карту-справочник. А также были рассмотрены принципы структурной организации клиент-серверных систем.

Спроектирован архитектуру системы ее клиентскую и серверную части, в виде UML диаграмм определена структура хранения данных пользователя и данных об объектах поиска. Построенная сеть Петри с

*графом достижимых маркировок, выполнен анализ и моделирование сети. Также на основе граф схем алгоритмов был разработан алгоритмы необходимы для реализации поставленных задач.*

*Разработана система реализующая приложение карту, справочник которая может быть внедрена в туристической отрасли для использования туристическими агентствами.*

**Ключевые слова:** *клиент-сервер, сеть Петри, веб-сервис, сетевые протоколы, карта-справочник telegram-bot.*

**Summary.** *The study considers analogs of the developed application, a model has been developed that has features - the sections of positions and transitions are introduced for the interaction of the application with the Google service, provides the user with the opportunity to immediately receive a reference map without using other software, and saves the internal memory of the device. The constructed model was used when designing a client-server system providing a reference map. The principles of the structural organization of client-server systems were also considered.*

*The architecture of the system is designed, its client and server parts, in the form of UML diagrams, the structure of storing user data and data about search objects is determined. The constructed Petri net with a graph of reachable markings, the analysis and modeling of the net is performed. Also, based on the graph of algorithms schemes, the algorithms necessary for the implementation of the tasks were developed.*

*A system has been developed that implements the application of the map reference book, which can be introduced in the tourism industry for use by travel agencies.*

**Key words:** *client-server, Petri net, web service, network protocols, telegram-bot reference map.*

**Постановка проблеми.** На сьогоднішній день жодна людина не відправиться в подорож без карти, так необхідність додатку карти довідника обумовлена великим кроком в розвитку туристичної галузі та розвитку карт в вигляді веб сервісів. Одним з найважливіших переваг клієнт-серверних систем у порівнянні з їхніми аналогами є зниження мережного трафіка при виконанні запитів саме тому користувач повинен мати лише пристрій з доступом в інтернет для отримання можливості працювати з інтерактивною картою довідником. Саме тому додаток що розробляється має велику кількість можливостей для застосування, як в туристичній сфері так і для щоденного використання. Завдяки використанню стандартів протоколу TCP/IP користувач має можливість використання системи з різних пристроїв.

**Аналіз останніх досліджень і публікацій.** Серед прямих аналогів системи розробленої в даному проекті можна відділити тільки один приклад. Цим прикладом є «TravelersBot».

Так «TravelersBot» Реалізує функції пошуку об'єктів, та пошук пробок на дорогах, також є можливість пошуку різних об'єктів за допомогою вводу тексту з клавіатури. Однак мінусом цього проекту є його робота з Yandex.apі. що не дає можливості повного користування функціоналом проекту. Таким чином ми не можемо використовувати функцію пошуку пробок на дорогах, та не можемо отримати більш детальну карту. Також потрібно зазначити що цей проект має функцію пошуку об'єктів схожу з функцією що розробляється проекті, однак вони мають суттєві відмінності, так пошук об'єктів в проекті що розробляється працює за допомогою Google.apі, тож проблем з доступом до карт не буде. Також варто зазначити що має функцію пошуку оптимального маршруту, та функцію додавання об'єктів пошуку в обране [1, с. 137].

**Відокремлення невирішених раніше частин загальної проблеми.** У наш час є багато відомих додатків для користування картами, але всі ці сервіси але для користування цими сервісами необхідні сторонні програми, саме тому створення такого додатку на базі месенджера Telegram вирішує проблему використання зайвого місця на телефоні, а також дає додаткову зручність під час використання.

**Мета і задачі статті.** Дослідження та аналіз моделі клієнт-серверної системи, яка призначена для надання користувачам карти-довідника. Основні задачі: дослідження вже існуючих аналогів, створення моделі майбутнього сервісу, проектування клієнт-серверного додатку.

**Виклад основного матеріалу.** Архітектура клієнт-сервер пред'являє специфічні вимоги як до клієнта, так і до сервера. Програмна система, яка задовольнить цим вимогам, може вважатися клієнт-серверних додатком, який виконує розподілену обробку даних. Під розподіленою обробкою розуміється виконання серверної частиною програми запитів клієнтської частини. Серверна частина програми забезпечує зберігання даних і їх обробку, а клієнтська частина передає серверу відповідні запити [2, с. 35].

Переваги клієнт-серверних систем

- клієнт-серверний підхід - модульний, причому серверні програмні компоненти компактні і автономні;
- оскільки кожен компонент виконується в окремому захищеному процесі призначеного для користувача режиму, збій сервера не вплине на інші компоненти операційної системи;
- автономність компонентів уможливорює їх виконання на кількох процесорах на одному комп'ютері (симетрична багатопроцесорна обробка) або на декількох комп'ютерах мережі (розподілені обчислення);

– обов'язок клієнта, як правило, - надавати призначені для користувача сервіси і, перш за все, призначений для користувача інтерфейс, тобто кошти для прийому, відображення і редагування даних, введених користувачем, які служать основою для запиту серверу. Крім того, клієнт можна налаштувати на обробку частини даних, щоб зменшити навантаження на ресурси сервера [2, с. 42].

Комунікаційні протоколи використовувані у клієнт-серверної архітектури

Комунікаційний протокол — це обумовлені наперед правила передачі даних між двома пристроями. До основних параметрів, які описує протокол, відносяться:

- тип перевірки помилок, що використовується;
  - метод компресії (стискання) інформації (якщо такий є);
  - спосіб визначення передаючим пристроєм завершення передачі
- [2, с. 49].

Протоколи відрізняються своїми характеристиками: одні — більшою надійністю, другі — швидкістю передачі даних, треті — простотою.

Мережеві протоколи поділяють на рівні. Розподіл на рівні — це принцип, який поділяє протоколи на рівні, кожен з яких виконує певну дію та взаємодіє з іншими частинами протоколу за чітко розробленими схемами.

Наприклад, один рівень зайнятий адресами (наприклад IP-адресами Internet protocol), ще один може описувати, як закодувати текст, інший описує, як поводитись з повідомленнями (Simple mail transfer protocol, наприклад), а інший знаходитиме помилки (transmission control protocol), а інший відповідає за формування потоків бітів (point-to-point protocol), завдання другого — електричне кодування бітів. Поділ на рівні дозволяє частинам протоколу розроблятися і тестуватися без величезної кількості

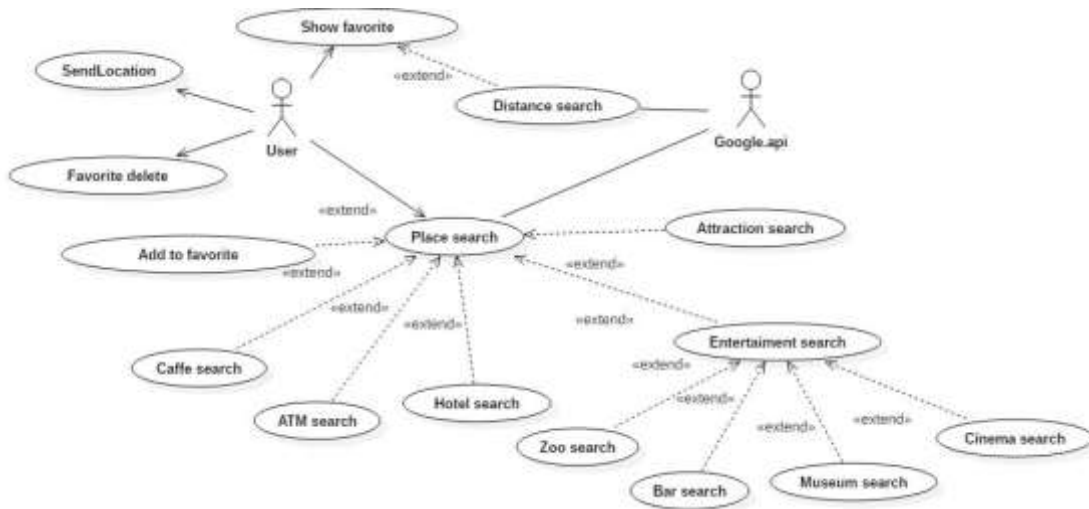
різних варіантів і таким чином зберегти кожний протокол відносно простим. Крім того, поділ на рівні дозволяє близьким протоколам бути пристосованими до незвичайних обставин. Так, наприклад, поштовий протокол може бути пристосований для надсилання повідомлень літаку [2, с. 58].

Архітектура клієнт сервер- архітектурний шаблон програмного забезпечення взаємодію між об'єктами в комп'ютерній мережі. Цей вид взаємодії називають розподіленою обробкою. Розподіленою обробкою називають передачу клієнтською частиною запитів до сервера, та виконання серверною частиною програми запитів клієнтської частини. Серверна частина програми забезпечує зберігання даних і їх обробку [3, с. 62].

Для кращого розуміння архітектури необхідно зазначити що клієнт робоча станція, що взаємодіє з користувачем, часто це інтерфейсний (зазвичай графічний) компонент системи, надається кінцевому користувачу. А сервер це або робоча станція або інша машина, з програмним забезпеченням яка виконує сервісні функції по запити клієнта, надаючи йому доступ до певних ресурсів або послуг. В цьому проекті використовується дворівнева архітектура це значить що уся обробка запитів проходить на одній системі без використання сторонніх ресурсів [3, с. 84].

На рисунку 1 зображено діаграму прецедентів





**Рис. 1. Діаграма прецедентів**

*Джерело: складено автором*

Були реалізовані наступні функції:

- пошук об’єктів на карті;
- пошук маршруту до об’єктів зі списку обраного;
- додавання об’єктів в обране;
- відображення обраного;
- видалення обраного.

Першим актором на діаграмі є актор «Користувач».

Користувач має пряму асоціацію за чотирьома прецедентами «Відправка локації», «Видалити обране», «Відобразити обране», «Пошук об’єктів».

Пошук об’єктів має шість розширень «Додавання в обране», «Пошук кафе», «Пошук банкомата», «Пошук готелів», «Пошук розваг», «Пошук пам’яток». Варіант використання «Додавання в обране» вибір об’єктів користувачем, та додавання їх в таблицю обраного в базі даних. Варіанти «Пошук кафе», «Пошук банкомата», «Пошук готелів», «Пошук пам’яток», реалізують пошук відповідних об’єктів на карті, і є кінцевими варіантами використання. «Пошук розваг» в свою чергу теж має розширення, а саме



«Пошук зоопарків», «Пошук музеїв», «Пошук кінотеатрів», «Пошук барів», «Пошук нічних клубів», усі приведені варіанти також виводять результат пошуку відповідних запитів користувача до системи.

Варіант використання «Відправка локації» не має залежних від нього прецедентів, та виконує запит функцію відправки місцезнаходження користувача в таблицю в базі даних системи.

Варіант використання «Видалити обране» також не має залежних від нього прецедентів, та виконує функцію відправки запиту на видалення об'єктів доданих користувачем в обране за бази даних.

Варіант використання «Відобразити обране» має залежний від нього прецедент, а саме «Пошук маршруту», який в свою чергу реалізує функцію пошуку маршруту до об'єктів на карті зі списку обраного, яка реалізується запитом системи, в Гугл, за допомогою google.apі для отримання результатів пошуку. Сам варіант «Відобразити обране» виконує функцію запиту системи для відображення об'єктів доданих користувачем в обране з бази даних [4, с. 347].

Другим актором на діаграмі є актор «Google.apі» який має непряму асоціацію з варіантами використання «Пошук об'єктів» та «Пошук маршруту» і служить для реалізації запитів до системи Гугл, за допомогою google.apі, він необхідний для отримання результатів запитів пошуку системи, та отримання результатів пошуку.

Система, розроблювальна в проекті є клієнт серверним додатком, це означає що вона складається з клієнтської та серверної частин: [4, с. 362]

- клієнтською частиною виступає веб сервіс – Telegram;
- серверною частиною виступає віддалений – сервер.

В ролі клієнтської частини виступає веб сервіс Telegram.Aпі, з цієї причини для зв'язку серверу з клієнтською частиною використано протокол

tcp/ip http завдяки якому забезпечується взаємодія серверної та клієнтської частин, також для взаємодії з базою даних використано протокол tcp/ip який забезпечує передачу даних між клієнтською та серверною частинами. Клієнтська частина виконує функцію відображення інтерфейсу, завдяки якому користувач має можливість виконувати програмні функції. Серверна частина виконує зберігання об'єктів інтерфейсу користувача, та відправку їх на клієнтську частину, а також зберігання об'єктів які виконують програмні функції, а також зберігання даних.

Для роботи функцій «Add Favorite», «Show favorite», «Favorite delete», «SendLocation», необхідна база даних для зберігання даних.

На рисунку 2 зображена взаємодія клієнтської та серверної частин яка відбувається за допомогою мережевого протоколу «TCP IP», тож на початку роботи клієнтська частина отримує файл користувацького інтерфейсу яке включає в себе такі об'єкти: «Menu», «InlineKeyboardMenu», «TextMessage», за допомогою яких відбуваються відправка та відображення запитів користувача до серверу, також варто зазначити що усі запити та відображення інтерфейсів функціонує за допомогою «Telegram.api» [5, с. 267].

Відправка запиту на пошук об'єктів на карті

При відправці запиту користувача на пошук об'єктів на карті відправляється запит до серверу, а також проходить перевірка локації користувача, з послідуною перевіркою на наявність об'єктів для обраного місцезнаходження.

На рисунку 2 представлена діаграма послідовності для цього варіанту розвитку.

1) Відправка користувачем запиту на сервер для пошуку об'єктів на карті (10);

- 2) Перевірка запиту сервером на коректність (11);
- 3) Відправка сервером Запиту користувача в гугл (12);
- 4) Обробка запиту в гуглі;
- 5) Якщо результат не пустий, гугл відправляє відповідь на сервер, з списком об'єктів відповідних критерію запиту який відправляється користувачу;

б) Якщо результат пустий, то гугл відправляє повідомлення про відсутність об'єктів, на сервер, а сервер відправляє повідомлення з інформацією про те що об'єкти в локації користувача відсутні.

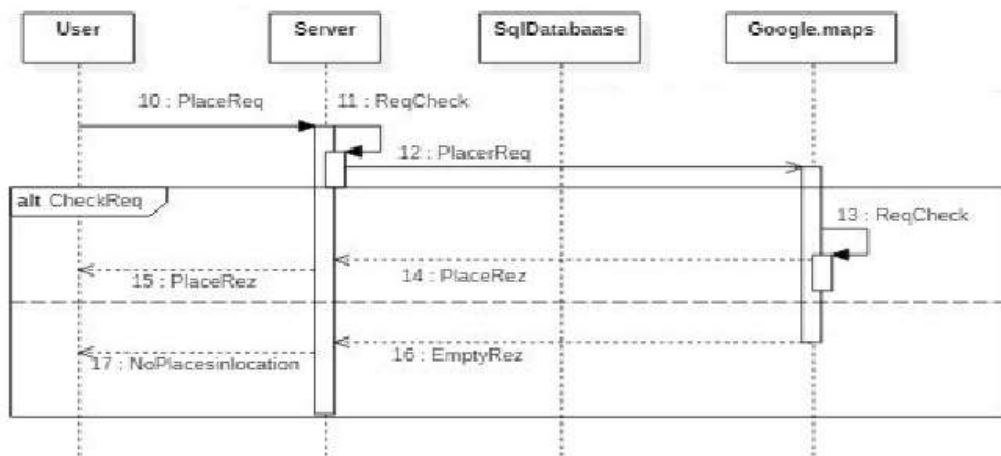


Рис. 2. Діаграма послідовності Відправка запиту на пошук об'єкту на карті

Джерело: складено автором

Для моделювання систему була обрана мережа Петрі. Таким чином мережу Петрі що представлена на рисунку 6 можна поділити на 5 складових:

- користувач клієнт серверного додатку який виступає генератором запитів;
- клієнтська частина додатку – виконує обробку запитів користувача та відправляє їх подальшу обробку до серверної частини;
- серверна частина додатку – обробляє запити що надходять від клієнтської частини, та виконує відповідні запити до бази даних, після

отримання відповіді від бази даних відправляє необхідну інформацію до клієнтської частини системи;

- база даних – виконує роль сховища даних , обробляє запити що надходять від серверної, та після обробки надсилає відповіді до серверної частини системи;
- частина система що відповідає за статистичний аналіз: – виконує збирання даних які модель отримує підчас роботи: загальна кількість оброблених запитів, а також сумарний, максимальний мінімальний середній час обробки запитів

На рисунку 3 наведено фрагмент мережі Петрі, що відповідає клієнтській частині клієнт-серверного додатку.

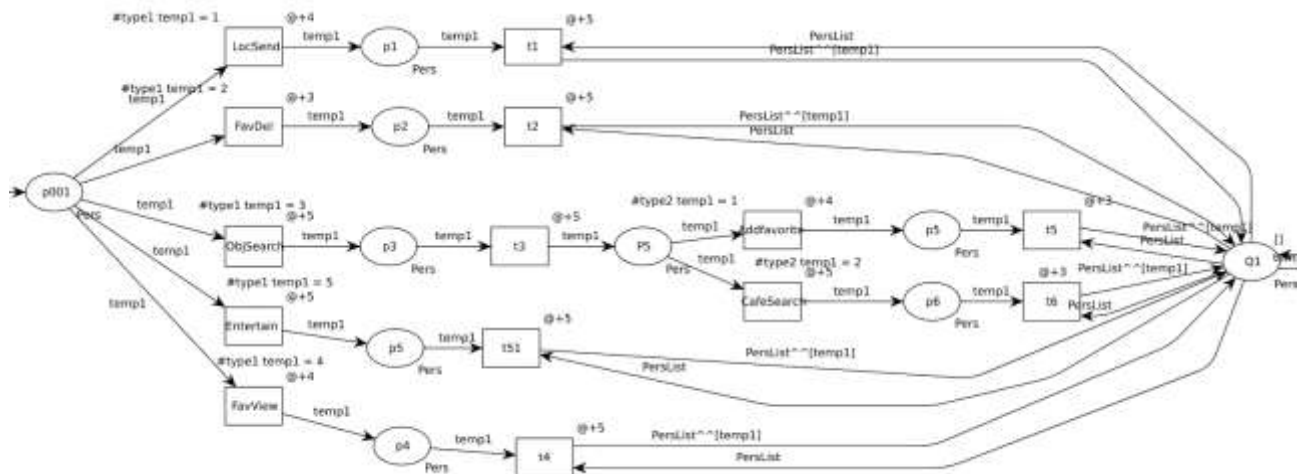


Рис. 3. Клієнтська частина мережі Петрі

Джерело: складено автором

На рисунку 4 наведено фрагмент мережі Петрі, що відповідає серверній частині клієнт-серверного додатку.

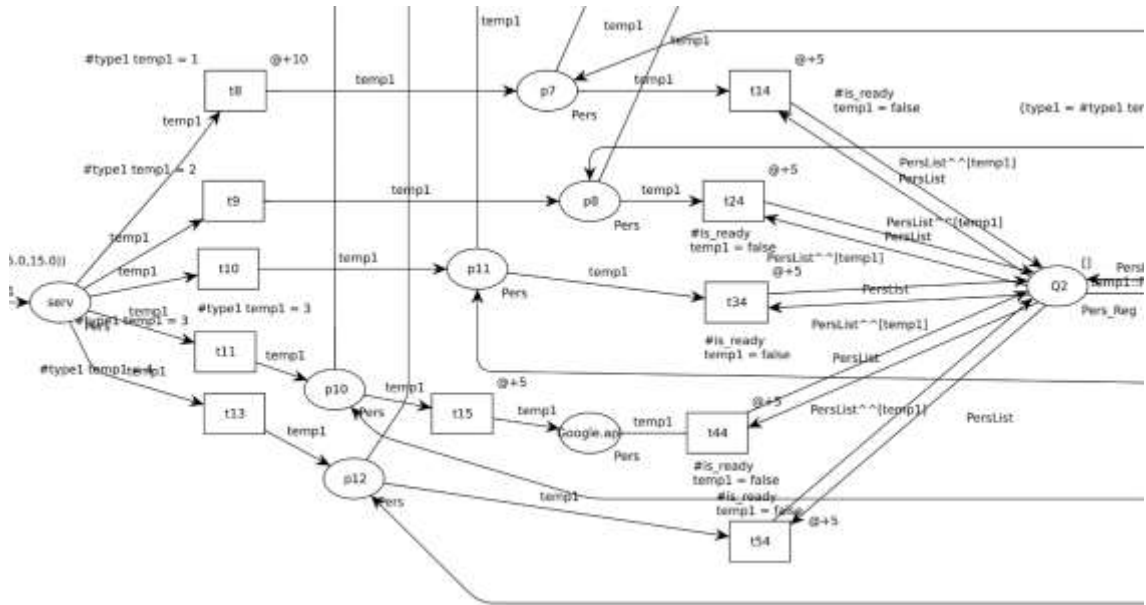


Рис. 4. Серверна частина мережі Петрі

Джерело: складено автором

Формальне подання мережі Петрі виглядає наступним чином:

$$C = (P, T, E),$$

де  $P$  – непушта кінцева множина позицій мережі,

$T$  – непушта кінцева множина переходів,

$E = (P \times T) \cup (T \times P)$  – відношення інцидентності позицій та переходів (множина дуг мережі).

Для отриманої мережі:

$$P = \{user, p_{001}, p_{51}, p_{1...12}, q_{1...3}, server, r_{1...5}, google.api, database, \}$$

$T$

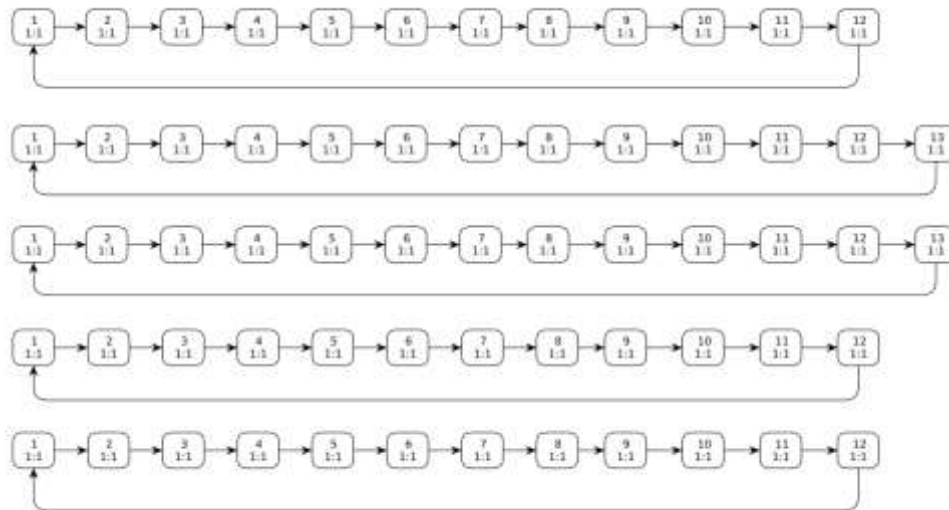
$$= \left\{ \begin{array}{l} choose, locsend, favdel, objsearch, entertain, favview, addfavorite, cafesearch, \\ t_{0...14}, t_{24}, t_{34}, t_{44}, tr_{54}, t_{114}, t_{224}, t_{334}, t_{444}, t_{554}, t_{4444}, t_{5554}, t_{3334}, t_{2224}, t_{1114}, \\ t_{11114}, t_{22224}, t_{33334}, t_{44444}, t_{55554}, t_{111114}, t_{222224}, t_{333334}, t_{444444}, t_{555554} \end{array} \right\}$$

Виділимо ділянку  $E_1 = (t_{15} \times google.api)$  та  $E_2 = (google.api \times t_{44})$ .

Модель відрізняється тим що введені ділянки для взаємодії додатку з сервісом Google, що надає користувачу можливість одразу отримувати карту

—довідник, не застосовуючи інше програмне забезпечення, що дає економію внутрішньої пам'яті пристрою.

Далі було побудовано граф досяжних маркувань мережі Петри що представлений на рисунку 5.



**Рис. 5. Граф досяжних маркувань**

*Джерело: складено автором*

Граф досяжних маркувань розділено на 5 частин кожна з яких відповідає одному з типів запитів користувача. Необхідно зауважити що кожна з частин має циклічний вигляд, це означає що кожна з частин є незалежною, що приводить до неможливості виникнення ситуацій, коли один тип запитів заважає роботі іншого.

### **Реалізація та апробація**

Результати моделювання мережі Петрі наведені у таблиці 1. Моделювання проводилось у середовищі «CPNTools» із наступним апаратним забезпеченням: процесор – «Intel Core i5-9300H», обсяг оперативної пам'яті – 8 Гб, відеоадаптер – «Nvidia Geforce GTX 1660 Super».

**Результати моделювання**

К-сть кроків	Середній час 1, мс	Середній час 2, мс	Середній час 3, мс	Середній час 4, мс	Середній час 5, мс
2000	47	35	42	48	36
6000	48	37	46	46	35
12000	47	37	47	46	36
20000	48	37	47	46	35
26000	46	37	47	46	35

*Джерело:* складено автором

Проаналізувавши результати моделювання можна зробити висновок що найменшим є час відповіді на другий та п'ятий типи запитів – «Додавання в обране» та «Видалення обраного», такі результати через те що вони не потребують складних операцій, а тільки запиту до бази даних, та отримання відповіді користувачу, про вдалу обробку запиту. Найбільшим є час відповіді першого типу запитів а саме – «Відправка місцезнаходження» , такі результати отримані через те що цей запит потребує запиту інформації з клієнтською частини системи, тобто клієнту, після чого необхідно обробити отримані дані, та надіслати та зберегти їх в базі даних. Середній час обслуговування знаходиться в діапазоні від 35 до 48 мс, такий результат є задовільним для сучасних клієнт-серверних систем.

**Висновок.** Під час дослідження було виконано проектування, розробку моделювання та реалізацію клієнт серверної системи яка забезпечує карту-довідник. Для показу моделі системи була побудована та змодельована мережа Петрі, за результатами якої можна зробити висновок що час обробки запитів, та надсилання відповіді користувачам задовільний на сьогоднішній день Одним з найважливіших переваг клієнт-серверних систем у порівнянні з їхніми аналогами є зниження мережного трафіка при виконанні запитів саме тому користувач повинен мати лише пристрій с доступом в інтернет для отримання можливості працювати з інтерактивною картою довідником.



Клієнт серверний додаток реалізує алгоритм пошуку об'єктів на карті, пошук та відображення оптимального маршруту до об'єктів, також користувачу надана можливість додавання об'єктів в персональне сховище даних, для швидкого пошуку об'єктів, та маршруту до них. Саме тому додаток що розробляється в цьому проекті має велику кількість можливостей для застосування, як в туристичній сфері так і для щоденного використання.

### **Література**

1. Машнин, Т.С. Google App Engine Java и Google Web Toolkit: разработка Web-приложений / Т.С. Машнин. СПб.: BHV, 2014. 352 с.
2. Буров С. В. Комп'ютерні мережі: підручник / Євген Вікторович Буров. Львів: «Магнолія 2006», 2010. 262 с.
3. Erik Wilde, Cesare Pautasso. REST: From Research to Practice. Springer Science & Business Media, 2011. 528 p. URL: [https://www.researchgate.net/publication/229431202\\_REST\\_From\\_Research\\_to\\_Practice](https://www.researchgate.net/publication/229431202_REST_From_Research_to_Practice)
4. Чамберс Д., Пэкетт Д., Тиммс С. «ASP. NET Core. Розробка додатків», «Питер» 2017. 464 с.
5. Камер Дуглас, Стівенс Девід Л. «Сети TCP/IP, том 3. Розробка додатків типу клієнт-сервер», видавництво «Вільямс», 2002 г. 592 с.