

ГЕОКОНТЕКСТНИЙ СЕРВІС ОРГАНІЗАЦІЇ ДОСТУПУ МОБІЛЬНИХ СУТНОСТЕЙ ДО СТАЦІОНАРНИХ РЕСУРСІВ

© Бочкаръов О.Ю., Алексейчик І.П., 2016

Розглянуто проблему розробки геоконтекстного сервісу організації доступу мобільних сутностей до стаціонарних ресурсів. Проаналізовано особливості роботи сервісу на прикладі задачі організації доступу до місць паркування. Запропоновано структуру та алгоритми роботи сервісу. Проведено моделювання роботи сервісу та реалізовано його прототип.

Ключові слова: геоконтекстний сервіс, мобільна сутність, стаціонарний ресурс

LOCATION BASED SERVICE FOR PROVIDING ACCESS OF MOBILE ENTITIES TO STATIONARY RESOURCES

© Botchkaryov A., Alekseychik I., 2016

The problem of developing location based service for providing access of mobile entities to stationary resources is considered. The details of service functioning was analyzed using the problem of access to the parking spaces as example. The structure and algorithms of the service are proposed. Service modeling and prototyping are considered.

Keywords: adaptive process of information gathering, mobile entity, stationary resource

1. Задача організації доступу мобільних сутностей до стаціонарних ресурсів.

В загальному вигляді в рамках даної задачі розглядаються 1) множина мобільних сутностей (користувачів сервісу) $K = \{k_1, k_2, \dots, k_m\}$, які можуть вільно або з деякими заданими обмеженнями пересуватися у просторі; 2) множина стаціонарних ресурсів $R = \{r_1, r_2, \dots, r_n\}$, розташування яких у просторі фіксоване і незмінне у часі; 3) множина з'явок (запитів) $Q = \{q_1, q_2, \dots, q_w\}$ на використання ресурсів, які генерують мобільні сутності. Кожна заявка q_x містить координати (або індекс) потрібного мобільній сутності k_i ресурсу r_j та дві часові відмітки (початок та кінець), якими мобільна сутність вказує відрізок часу, на протязі якого вона планує користуватись ресурсом r_j . Завданням сервісу є організація виконання заявок власниками ресурсів (рис.1) з мінімізацією 1) кількості відхилення заявок (за наявності такої можливості з огляду на наявні в даний момент вільні ресурси) та 2) середнього часу обробки одної заявки. При цьому в роботі сервісу за різними сценаріями враховується поточне місце розташування мобільних сутностей (наприклад, для визначення того факту, що мобільна сутність не зможе (не встигне) скористатись віділенням її ресурсом з огляду на її поточне місцерозташування та максимально можливу швидкість її пересування у просторі). Відтак сервіс можна віднести до класу геоконтекстних сервісів (location based service) [1, 2],

темпи розвитку яких на даний момент різко зростають внаслідок широкого розповсюдження мобільних обчислювальних пристроїв.

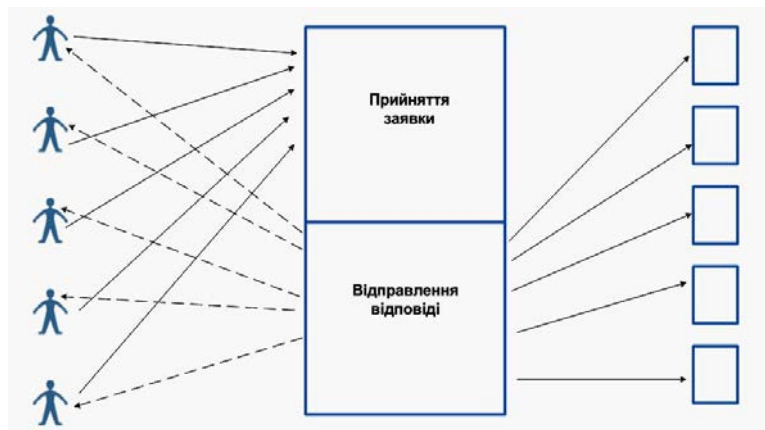


Рис. 1. Схема задачі
(зліва показані мобільні сутності K , справа — стаціонарні ресурси R)

4. Сервіс організації доступу до місць паркування

Одним з найбільш актуальних на даний час варіантів змістовної інтерпретації даної задачі є сервіси організації доступу до місць паркування, в тому числі сервіси в рамках концепції спільного використання приватних місць паркування (shared parking). В даному випадку в ролі мобільних сутностей виступають водії (далі в тексті – клієнти-водії), а в якості стаціонарних ресурсів – місця паркування, безпосередній контроль за використанням яких здійснюють їх власники (далі в тексті – клієнт-парковка). Розглянемо задачу організації доступу мобільних сутностей до стаціонарних ресурсів на прикладі саме цієї змістовної інтерпретації. Сервіс складається з трьох складових частин це: клієнт-водій, сервер і клієнт-парковка. Сервіс передбачає необмежену кількість клієнтів-водіїв так і клієнтів-парковок. Клієнт-водій і клієнт-парковка не можуть спілкуватись безпосередньо, лише через сервер, тобто інформацію про парковки і про водіїв можливо отримати лише через сервер. При цьому клієнт-водій може виконувати такі функції і в такій послідовності: з'єднання з сервером, визначення поточних координат, вибір місця парковки (формування заявки), резервування місця парковки (відправка заявки та отримання підтвердження її виконання).

Якщо клієнт-водій оформив заявку і її підтвердили, то він отримає ключ для ідентифікації і графік де буде вказано час коли він має прибути і покинути стоянку (можливо з незначними змінами значень, які були вказані в його заявці). Припустимо, що в заявці клієнт-водій вказав час прибуття t_{start} і відїзду t_{end} . Система виділить для прибуття проміжок часу (рис.2) від $t_{start} - dt$ до $t_{start} + dt$, щоб клієнт-водій зміг без всяких перешкоди заїхати на своє місце і проміжок часу за який водій повинен покинути: від $t_{end} - dt$ до $t_{end} + dt$. Проміжок часу dt фіксований. Його значення обирається на етапі налаштування роботи сервісу з огляду на доступні часові послаблення щодо часових відміток, які користувачі вказують у заявках.



Рис. 2. Схема часових обмежень на початок та кінець використання ресурсу

Умовно цю часову пряму (рис.2) можна розділити на 4 частини: 1) клієнт-водій прибув до t_{start} ; 2) клієнт-водій прибув після t_{start} , але не пізніше ніж пройшла половина зарезервованого ним часу; 3) клієнт-водій покинув місце паркування пізніше ніж пройшла половина зарезервованого ним часу, але раніше ніж t_{end} ; 4) клієнт-водій покинув місце паркування пізніше t_{end} .

Для частини 1 можливі такі варіанти:

1. $t_{прибуття} < t_{start}$ – даний варіант є сприятливий якщо клієнт-водій прибудить на проміжку часу від $t_{start} - dt$, тоді він зможе без проблем зайняти своє місце паркування. Якщо клієнт-водій прибуде раніше це змусить його чекати щоб зайняти місце паркування, або в кращому випадку система зможе змінити йому місце паркування щоб він зміг припаркуватися.
2. $t_{прибуття} = t_{start}$ – даний варіант є найбільш сприятливий для системи і проблем з паркуванням не має виникнути.

Для частини 2 можливий такий варіант:

$t_{прибуття} > t_{start}$ – даний варіант є сприятливий якщо клієнт-водій прибуде до часу $t_{start} + dt$. Якщо він прибув пізніше, сервіс в кращому випадку зможе змінити йому місце паркування або взагалі позбавить можливості припаркуватись.

Для 3 частини можливі такі варіанти:

1. $t_{відправлення} < t_{end}$ – даний варіант буде сприятливий для системи якщо водій покине місце паркування не раніше $t_{end} - dt$, якщо ж раніше, то це призведе до простою місця паркування.
2. $t_{відправлення} = t_{end}$ – даний варіант є найбільш сприятливий для сервісу.

Для 4 частини можливий такий варіант:

$t_{відправлення} > t_{end}$ – даний варіант буде сприятливий лише якщо клієнт-водій покине місце паркування до $t_{end} + dt$, в іншому випадку це призведе до порушення графіку, що нашкодить роботі сервісу.

По результатах виконання заявки клієнту-водію нараховуються бали. Бали можуть бути як штрафні, так і бонусні. Якщо клієнт-водій прибув в межах від $t_{start} - dt$ до $t_{start} + dt$, то йому нараховуються бонусні бали. Якщо ж водій прибув пізніше, то йому нараховуються штрафні бали. Якщо клієнт-водій покинув місце паркування в межах від $t_{end} - dt$ до $t_{end} + dt$, то йому нараховуються бонусні бали, якщо клієнт-водій покинув місце паркування раніше $t_{end} - dt$, то бали взагалі не нараховуються, якщо пізніше $t_{end} + dt$, то нараховуються штрафні бали.

Клієнт-водій може отримати повідомлення про зміну місця паркування. Зміна місця паркування може статися у випадку, коли клієнт-водій прибув на парковку пізніше $t_{start} + dt$, або якщо попередній клієнт-водій не покинув дане місце паркування вчасно (в умовах нестачі інших місць паркування). Про необхідність зміни місця паркування сервіс інформує клієнт-парковка. Клієнт-парковка відправляє запит на сервер і сервер визначає чи можлива зміна, в свою чергу сервіс інформує про це клієнта-водія. Сервер вирішує чи можлива зміна місця парковки для клієнта-водія на основі бонусних і штрафних балів. Також на зміну місця парковки впливає той факт, хто є її ініціатором. Якщо зміна місця парковки необхідна по вині клієнта-парковки то кількість балів не впливає на вирішення і заміна місця парковки відбудеться, якщо є така можливість. Якщо ініціатором є клієнт-водій то зміна місця парковки базується на сумарній кількості його балів.

Клієнт-водій має можливість скасувати заявку. Скасування заявки може бути двох типів: 1) якщо клієнт-водій відправив заявку, але сервер не встиг її розглянути, то скасування заявки буде без наслідків; 2) якщо клієнт-водій відправив заявку і сервер вже її

розглянув і відправив відповідь, то за скасування заявки можуть нараховуватися штрафні бали у випадку якщо водій скасував заяку пізніше t_{start} . В іншому випадку штрафні бали не нараховуються.

Після реєстрації клієнта-парковки, модератор перевіряє факт існування парковки за вказаною адресою. Тоді і перевіряє кількість місць паркування, створює карту місць паркування і активує клієнта-парковку. Клієнт-парковка може виконувати дві функції, що не зв'язані між собою, а саме: 1) перегляд черг на місця парковки; 2) відправка повідомлення про зміни в черзі. За певних обставин клієнт-парковка має можливість відхилити заявку. Якщо водій прибув пізніше $t_{start} + dt$, клієнт-парковка повідомляє про це сервер і сервер вносить зміни в графік місця паркування. Також клієнт-парковка може скасувати заявки і за інших обставин. В будь-якому випадку клієнт-парковка зобов'язується повідомляти сервер про ситуації, які можуть спричинити порушення графіку. Якщо клієнт-парковка робить зарезеровані клієнтами-водіями місця паркування не можливі в використанні, тобто скасовує заявки, за це клієнту-парковці нараховуються штрафні бали. Якщо кількість штрафних балів стане нижчою заданого порогового значення, сервіс заблокує клієнта-парковку на деякий час.

3. Структура сервісу та алгоритми його роботи

Структура сервісу (рис. 3) складається з трьох частин: серверної частини та двох клієнтських частин (для клієнта-водія та клієнта-парковки). Клієнтські частини сервісу взаємодіють з сервером через мережу Internet. В залежності від типів вхідних повідомлень та попередніх режимів роботи визначається сценарій роботи клієнта.

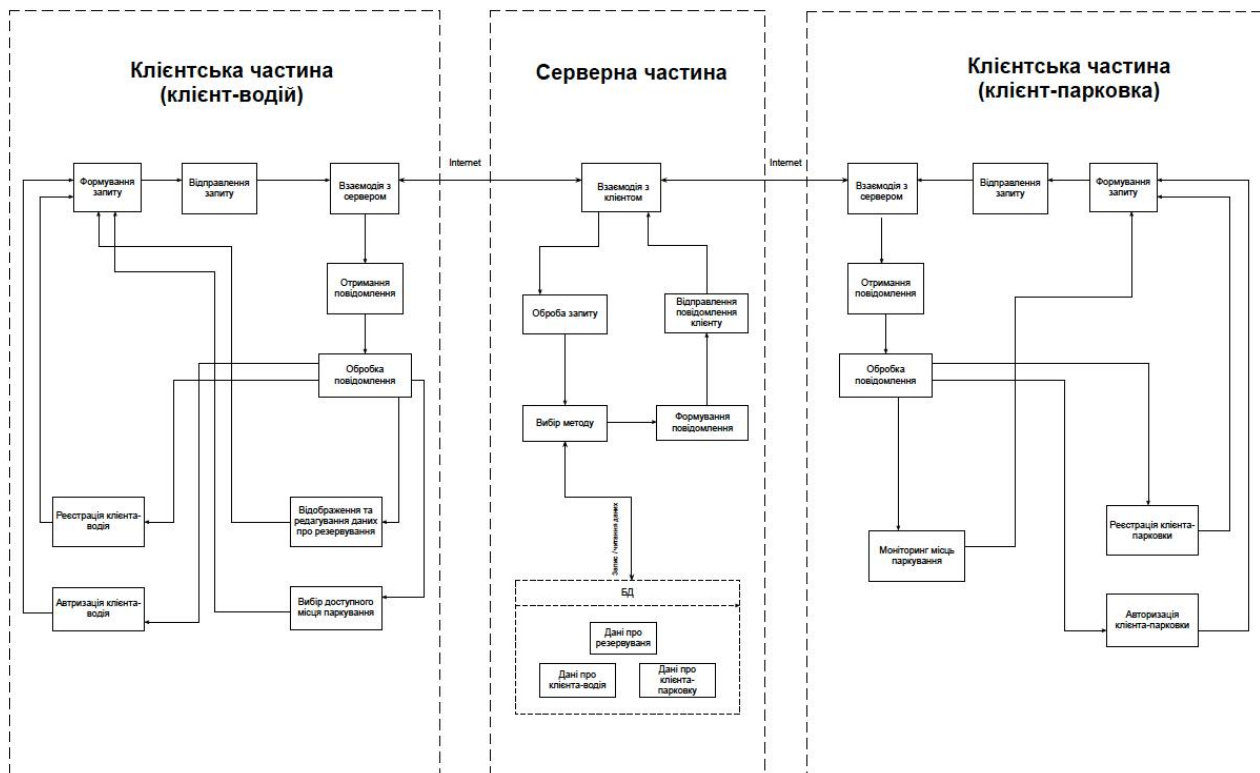


Рис. 3. Структурна схема сервісу

В ході своєї роботи клієнтські частини формують необхідні запити, відправляють їх на сервер та очікують відповіді. Сервер підтримує зв'язок з клієнтами, приймає від них запити

та обробляє їх. Обробка запитів передбачає всі можливі варіанти організації доступу до місць паркування, розглянуті вище. При потребі сервер зчитує або записує дані в базу даних і формує відповідні повідомлення-відповіді на запити клієнтів. Алгоритми роботи клієнтських частин та серверної частини наведені на рис. 4 та рис. 5.

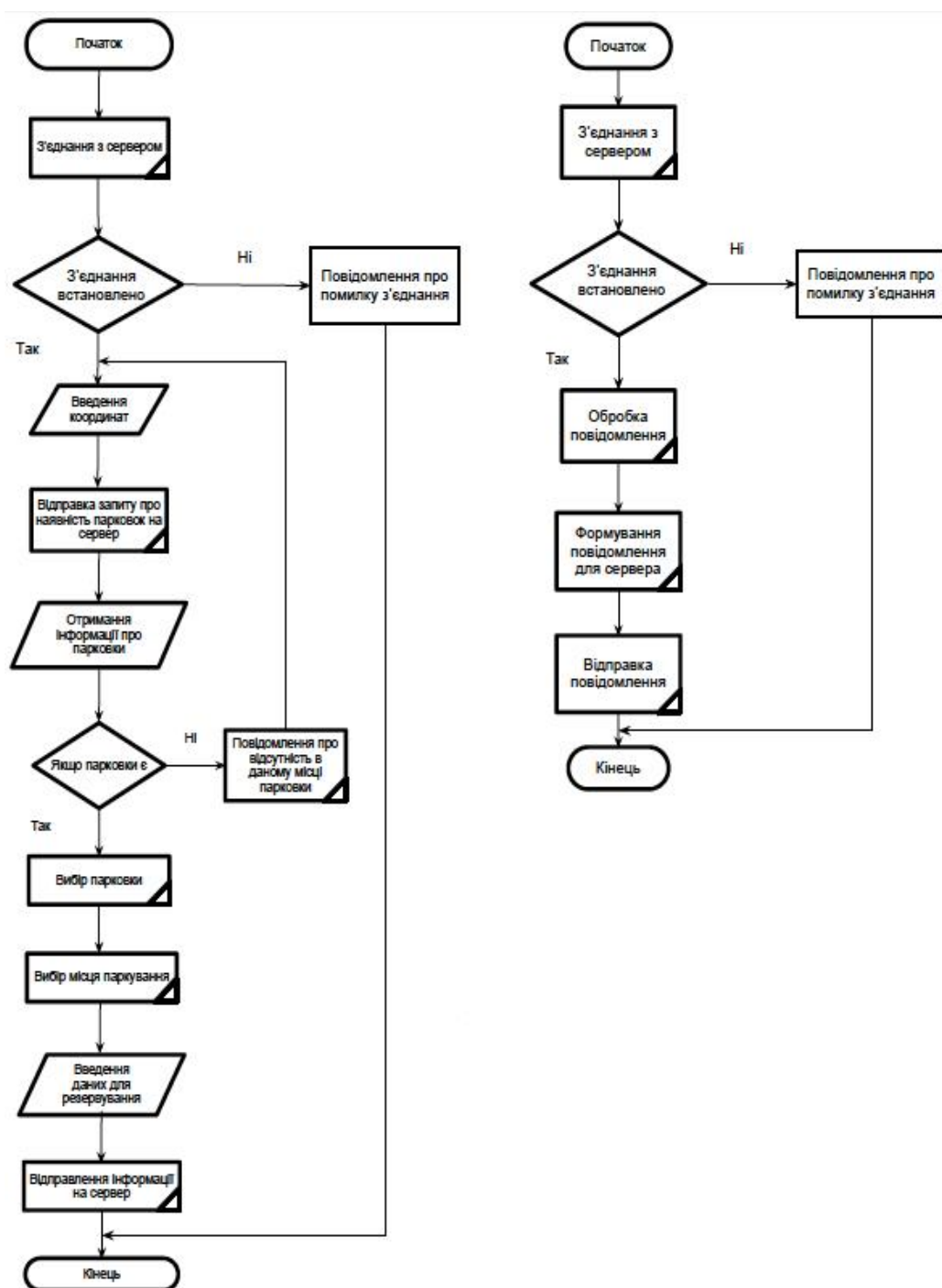


Рис. 4. Алгоритми роботи клієнтських частини сервісу

Було проведено моделювання роботи сервісу з використанням положень теорії масового обслуговування [4,5] та середовища моделювання AnyLogic (рис.6), яке підтвердило його працездатність. Також було реалізовано прототип сервісу на платформі Java. Діаграми класів клієнтської та серверної частини наведені на рис. 7 та рис. 8.

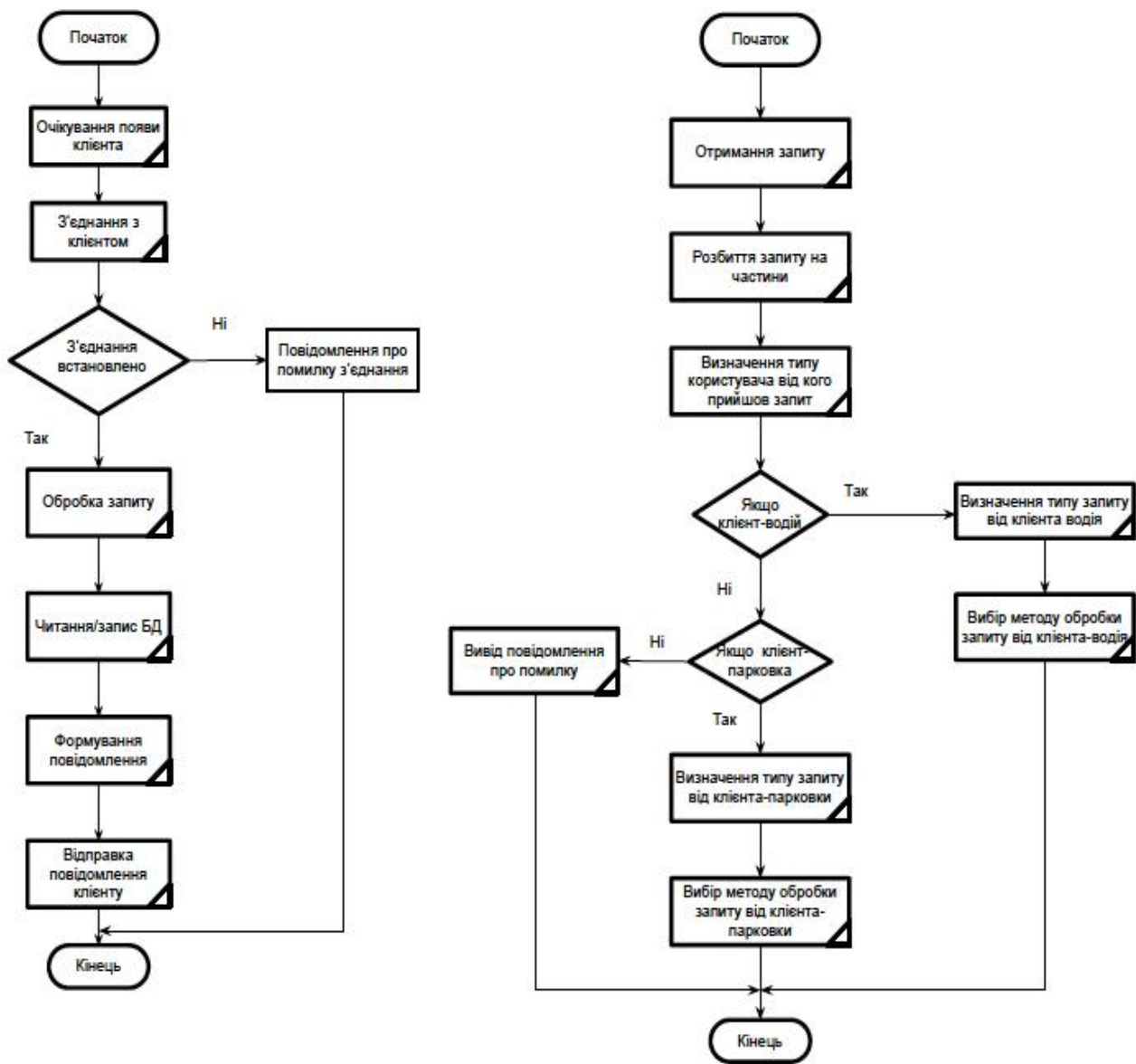


Рис. 5. Алгоритми роботи серверної частини сервісу

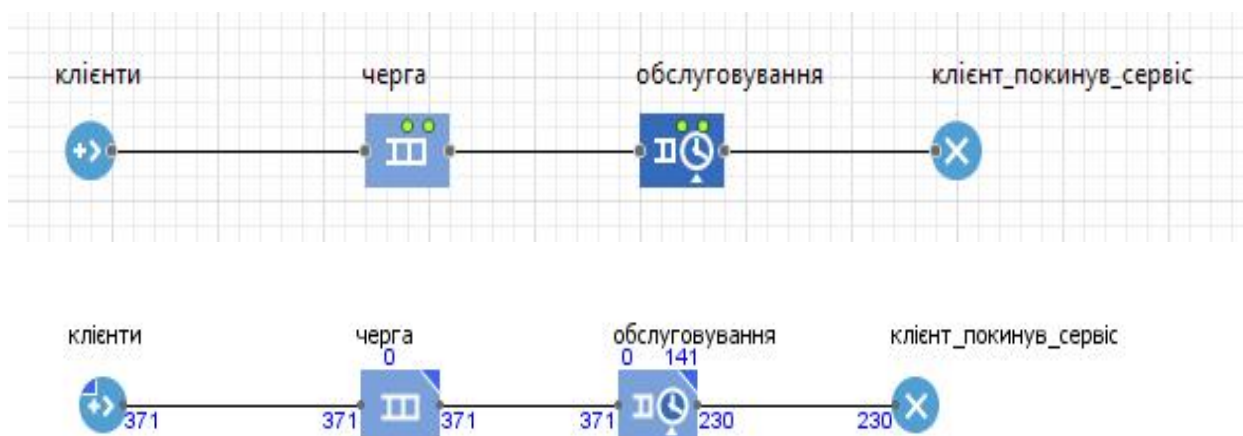


Рис. 6. Моделювання роботи сервісу в середовищі AnyLogic

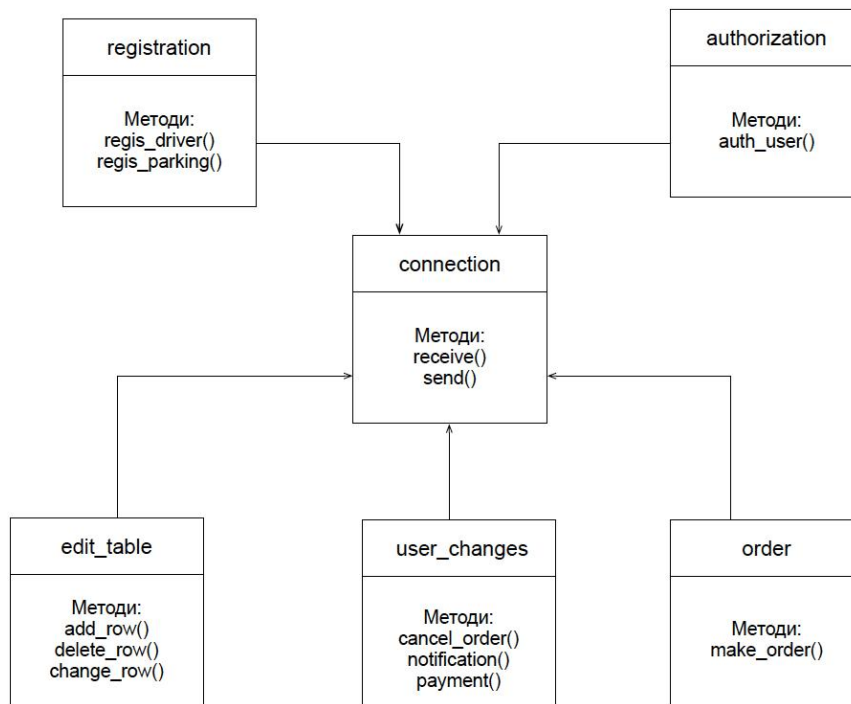


Рис. 7. Діаграма класів клієнтської частини сервісу

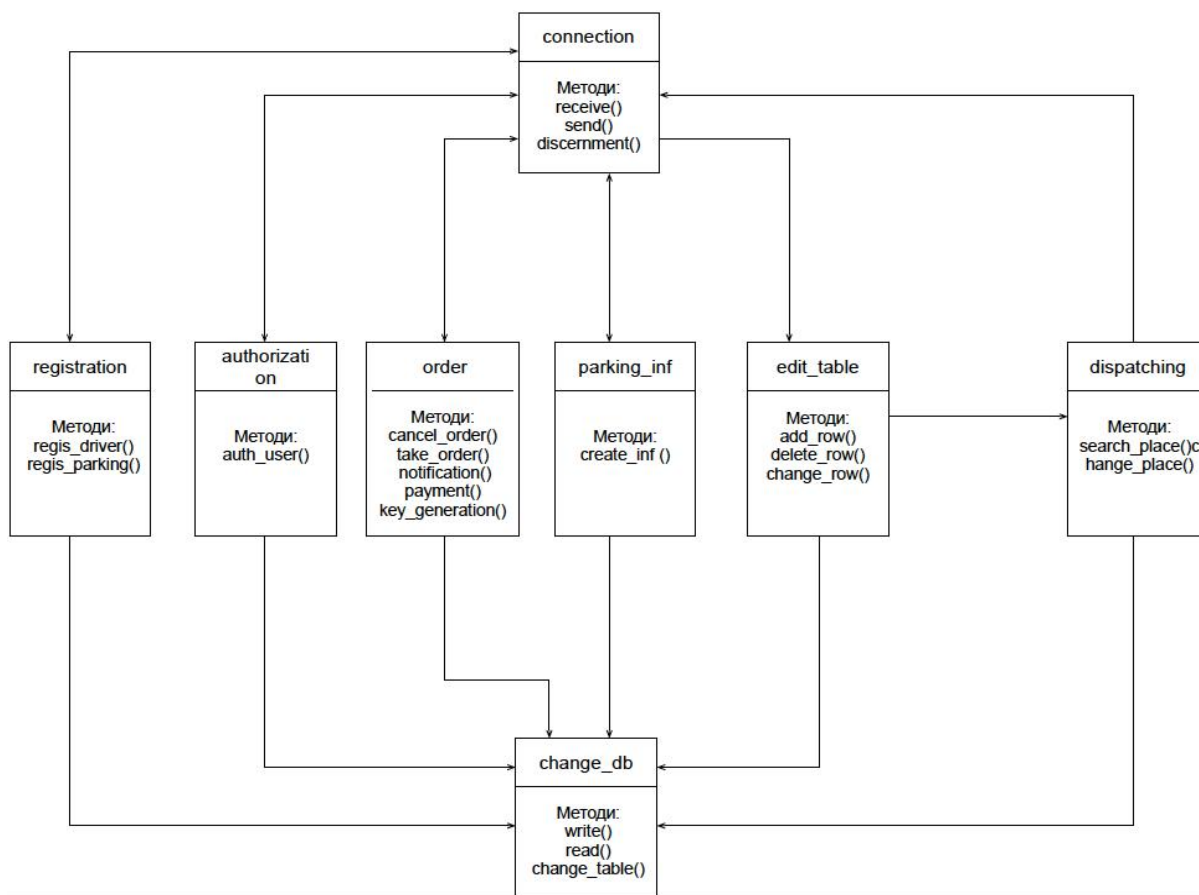


Рис. 8. Діаграма класів серверної частини сервісу

1. Allan Brimicombe, Chao Li, *Location-Based Services and Geo-Information Engineering*.– John Wiley & Sons, 2009. 2. *Location Based Services for Mobiles: Technologies and Standards*. Shu Wang,

Jungwon Min, Byung K. Yi. LG Electronics Mobile Research, USA. 3. Жерновий Ю. В. Марковські моделі масового обслуговування: Тексти лекцій. Львів: Видавничий центр ЛНУ імені Івана Франка, 2004. – 154 с. 4. Вентцель Е.С. Исследование операций: задачи, принципы, методология. Москва: Издательство «Наука». Главная редакция физико-математической литературы, 1988. – 208с.

Наукові результати, подані у цій статті, було отримано в рамках дослідницького проекту ДБ/КІБЕР з реєстраційним номером 0115U000446, 01.01.2015 – 31.12.2017, фінансово підтриманим Міністерством освіти та науки України.