**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”**

**Теплоенергетичний факультет**

**Кафедра автоматизації проектування енергетичних процесів і систем**

# **ЗВІТ З ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ**

### Виконав студент групи ТІ-62 Гавриляк Олександр Володимирович \_\_\_\_\_\_\_\_

### (шифр групи, прізвище, ім’я, по батькові) (підпис)

### Керівник практики від НТУУ “КПІ” Гагарін Олександр Олександрович \_\_\_\_\_

### (прізвище, ім’я, по батькові) (підпис)

### Керівник практики від підприємства

### ТОВ Бріз Софт Сметанюк Віктор Анатолійович \_\_\_\_\_

### (назва підприємства) (прізвище, ім’я, по батькові) (підпис)

Київ — 2020

Зміст

[**План практики** 3](#_Toc40307872)

[ВСТУП 4](#_Toc40307873)

[1. Постановка задачі 6](#_Toc40307874)

[1.1 Основні задачі 6](#_Toc40307875)

[1.2 Задачі для мобільного додатку 6](#_Toc40307876)

[1.2.1 Загальні вимоги 6](#_Toc40307877)

[1.2.2 Головні задачі 7](#_Toc40307878)

[1.2.3 Додаткові задачі 7](#_Toc40307879)

[1.3 Задачі для сервера 8](#_Toc40307880)

[1.3.1 Загальні вимоги 8](#_Toc40307881)

[1.3.2 Головні задачі 8](#_Toc40307882)

[2. Опис предметної області 9](#_Toc40307883)

[2.1 Аналогічні системи 9](#_Toc40307884)

[2.2.1 Система My LA 311 9](#_Toc40307885)

[2.2.2 Система Pakistan Citizen Portal 11](#_Toc40307886)

[2.2.3 Система 2GIS 13](#_Toc40307887)

[3. Засоби розробки 15](#_Toc40307888)

[3.1 Мобільний додаток 15](#_Toc40307889)

[3.1.1Flutter Framework 15](#_Toc40307890)

[3.1.2 Dart – мова програмування 15](#_Toc40307891)

[3.1.3 Visual Studio Code – редактор вихідного коду 15](#_Toc40307892)

[3.2 Сервер 16](#_Toc40307893)

[3.2.1 ASP.NET Core Framework 16](#_Toc40307894)

[3.2.2 MS SQL – база даних 16](#_Toc40307895)

[3.2.3 C# - мова програмування 16](#_Toc40307896)

[3.2 Технічні вимоги до програмного забезпечення сервера 16](#_Toc40307897)

[4.ОПис програмної реалізації 17](#_Toc40307898)

[4.1 Діаграма Use Case 17](#_Toc40307899)

[4.2 Інтерфейс мобільного додатку 19](#_Toc40307900)

[4.2.1 Екран входу 19](#_Toc40307901)

[4.2.2 Навігаційна панель 20](#_Toc40307902)

[4.2.3 Перемикач «Карта / Список» 20](#_Toc40307903)

[4.2.4 Компонент карта 21](#_Toc40307904)

[4.2.5 Компонент фільтрації 22](#_Toc40307905)

[4.3 Схема бази даних системи 24](#_Toc40307906)

[Інсталювання системи 26](#_Toc40307907)

[Висновки 27](#_Toc40307908)

[Список використаних джерел 28](#_Toc40307909)

# **План практики**

**Тема практики:** Огляд та аналіз готових схожих рішень, проектування та реалізація власної системи з серверною та мобільною частинами. Створення супроводжуючої документації.

**Завдання на практику:**

1. Ознайомлення з аналогічними програмними системами та їх документацією;
2. Розробка діаграми прецедентів.
3. Розробка мобільного додатку;
4. Розробка діаграми бази даних.
5. Розробка серверної частини;
6. Тестування й налагодження створеного програмного продукту;

ВСТУП

В наш час діджеталізації створюється потреба в моніторингу господарських проблем та публічних заходів промислового району для ефективного вироблення управлінських рішень. Системи, що вирішують зазначену проблему все частіше реалізуються у вигляді сервісних центрів з віддаленим доступом та спеціалізованими мобільними додатками.

Пропонована система «Awesome Map» вирішить як муніципальні проблеми університету, наприклад повний бак сміття біля гуртожитку, так і життєво небезпечні проблеми забезпечення порядку та закону.

Система, що розробляється має клієнт-серверну архітектуру та вирішує задачі моніторингу наявних господарських проблем (прорив водо-,тепло-, електричних мереж, наявність незручностей, сміття та таке інше), а також планування та проведення публічних заходів на території університету. Серверна частина організує ведення бази даних проблем що виникають, їх фіксацію та доступ до інформації необхідної для вироблення управлінських завдань.

Збір проблемної інформації та зведення її до єдиного сервісного центру виконується користувачами системи за допомогою мобільного додатку.

Подібні мобільні додатки починають розроблятися не тільки в Україні, а й по всьому світу, наприклад додаток для Los Angeles «MyLA311»[1] або для всього Пакистану – «Pakistan Citizen Portal»[2], або звичайна зручна карта мобільного додатка «2GIS» [3]. Всі додатки намагаються вирішувати свої поставлені проблеми за допомогою GIS (Geographic Information System) технологій. На жаль, кожен з них має свої недоліки, але, на щастя, має і свої переваги.

Далі роздивимось більш детально кожен з цих додатків, надамо стислу характеристику та порівняння між собою, виділивши конкретні недоліки та переваги.

Розробимо та проаналізуємо use-case діаграму прецедентів. Переглянемо основний функціонал мобільного додатку. Роздивимось концептуальну схему бази даних.

Система розроблялась за допомогою світових технологій: Google Cloud Platform, сучасного інструмента користувальницького інтерфейсу для мобільних додатків від Google – Flutter, та останніх технологій від Microsoft, для створення серверної частини системи – ASP.NET Core.

Бета версія системи та апробація всіх ії компонентів на даний момент завершені. З повною версією опису системи можна ознайомитися на сайті apeps.kpi.ua у розділі студентські випускні роботи.

1. Постановка задачі

## 1.1 Основні задачі

Система повинна вирішувати задачі моніторингу наявних господарських проблем, (прорив водо-,тепло-, електричних мереж, наявність незручностей, сміття та таке інше), а також планування та проведення публічних заходів на території кампуса університету.

Адміністрація НТУУ «КПІ» ім. Ігоря Сікорського повинна мати змогу відредагувати проблему або захід відповідно до робочих процесів системи.

## Задачі для мобільного додатку

### Загальні вимоги

Мобільній додаток повинен відповідати рекомендаціям дизайну для Android та використовувати загальнодоступні значки та шаблони інтерфейсу користувача.

У додатку не перевизначені звичні функції системних значків (наприклад кнопка "Назад").

У додатку не відбувається зміна системних значків абсолютно новими значками, якщо використовуються стандартні функції інтерфейсу користувача.

Мобільний додаток повинен бути адаптованим для використання його на різних пристроях з різними розмірами екранів.

Якщо в додатку пропонується адаптована версія стандартного системного значка, цей значок повинен в значній мірі нагадувати вихідний системний значок і порядок роботи з ним повинен відповідати поведінки системного значка.

Додаток не має перевизначати або некоректно використовувати стандартні шаблони інтерфейсу користувача Android, наприклад значки або їх дії, щоб не плутати користувачів і не ускладнювати їх роботу.

### Головні задачі

Користувач системи повинен мати змогу переглядати існуючі проблеми та заходи на окремих мапах і на одній загальній.

В системі повинна бути реалізована фільтрація проблем за наступними показниками:

* Назва
* Дата створення (початок проміжка до сьогодні)
* Категорія
* Статус проблеми

та заходів за іншими показниками:

* Назва
* Дата проведення з (початок обраного проміжка)
* Дата проведення до (кінець обраного проміжка)
* Місце
* Категорії

Доцільно зробити перехід від міток на мапі до більш детального списку елементів, відображеного користувачу.

Користувач повинен мати змогу переглянути тільки свої проблеми та заходи, а також відредагувати чи видалити їх, відповідно до робочих процесів системи.

Користувач повинен мати змогу змінити свої особисті та вхідні дані у системі.

### Додаткові задачі

Рекомендовано створити більш гнучке налаштування інтерфейсу, а саме: можливість змінити головну мову інтерфейсу з української на англійську, а також можливість змінити тему інтерфейсу зі стандартної (блакитної) на нічну (темну) тему.

## Задачі для сервера

### Загальні вимоги

Сервер має опрацьовувати http запити та повертати коректні відповіді.

Робота з даними повинна бути запезпечена за допомогою базових операцій REST - архітектурних додатків. [4] Для операцій з мережевим сховищем впроваджуються запити:

* Вилучення даних (GET)
* Збереження даних (POST)
* Зміна (PUT/PATCH)
* Видалення (DELETE)

### Головні задачі

Сервер повинен надавати усі необхідні дані мобільному додатку.

Він відповідальний за авторизацію та реєстрацію користувача в системі. Сервер надає доступ до бази даних, де зберігаються:

* Проблеми
* Типи проблем
* Заходи
* Коментарі
* Іконки
* Файли і таке інше.

1. Опис предметної області

## Аналогічні системи

### Система My LA 311

За допомогою системи MyLA311 у Лос-А́нджелесі, штат Каліфорнія, США, можна дізнатися інформацію про місто та послуги лише за декілька хвилин. [1] Використовувавання цього додатку дає змогу швидкого та легкого доступу до напопулярніших послуг міста, включаючи видалення графіті, ремонт вибоїн та забір об’ємних предметів.

Роздивимось більш детально мобільний додаток.  
На першому екрані (Рис 2.2.1.1) бачимо три пункти:

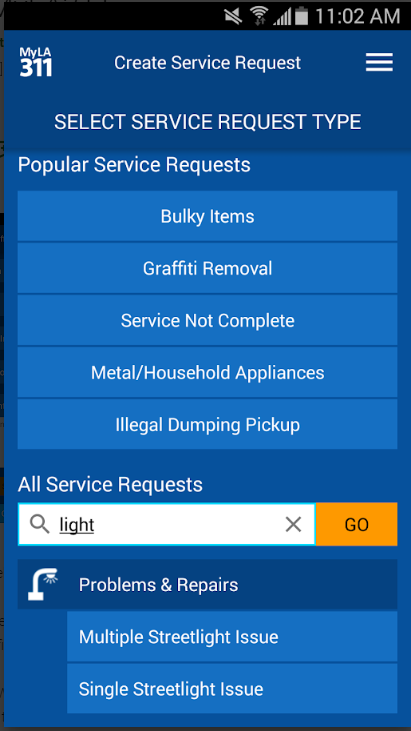
* Створення службового запиту
* Перегляд
* Інші служби

  
Рисунок 2.2.1.1 – Головний екран MyLA311

Головним функціоналом кожного з цих пунктів є:

* Функція «Створення службового запита», що дозволяє швидко та легко запросити найпопулярніші послуги міста, включаючи видалення графіті, ремонт вибоїн та забір об'ємних предметів.
* Функція "Перегляд", яка відображає найближчі парки, бібліотеки, пожежні та поліцейські пункти, громадські басейни, тенісні корти, поля для гольфу, паркінг тощо.
* Функція «Інші службі» дозволяє перейти до інших служб.

Інших сервісів наведено дуже багато (Рис 2.2.1.2). Для цього навіть впроваджено пошук.

  
Рисунок 2.2.1.2 – Усі доступні сервіси в системі

Описувати кожен займе багато часу, тому в загальному завдяки ним можливо:

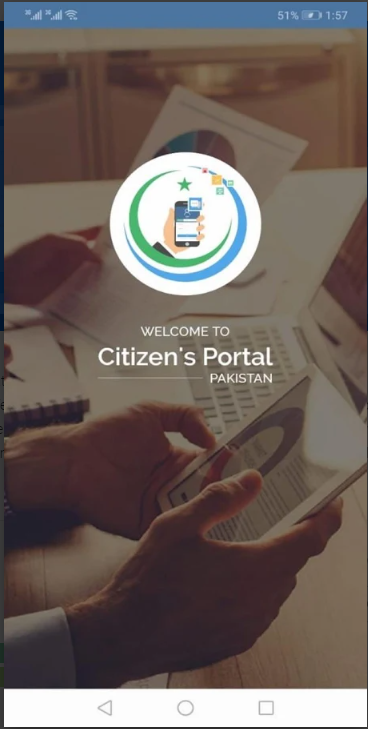
* оплачувати рахунки в місті за воду та електроенергію
* переглядати стрічку новин
* переглядати поточні статуси своїх запитів проблем і таке інше.

**Переваги:** доступ до великої цифрової системи міста, що має багато сервісів; зрозумілий та простий варінт інтерфейсу; можливість залишити мітку анонімно.

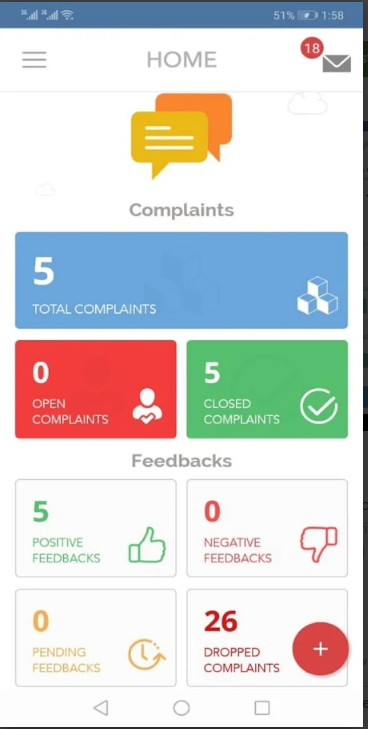
**Недоліки:** застарілий інтерфейс, що не відповідає новітнім вимогам для Android та IOS користувачів. Доступ до встановлення додатку можна отримати лише з території Сполучених Штатів Америки або використовувати VPN сервіси.

### Система Pakistan Citizen Portal

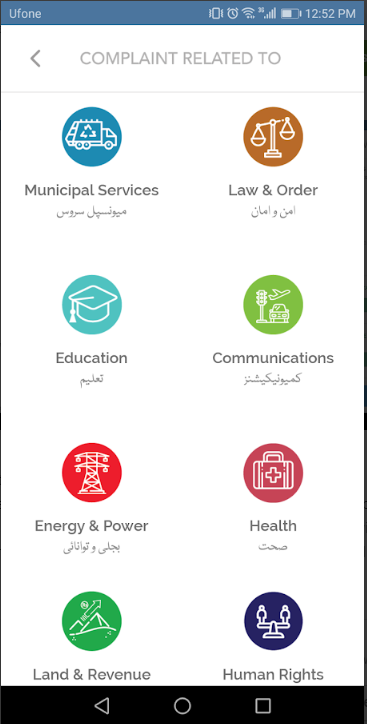
Портал громадянина Пакистану – це інтегрована система відкодування скарги громадян, яка з’єднує всі урядові організації як на федеральному, так і на провінційному рівнях. (Рис. 2.2.2.1) Система служить носієм скарг у відповідні офіси по всьому Пакистану. Додаток слугує додатковим каналом між громадянином та владою. [2]

  
Рисунок 2.2.2.1 – Екран привітання

Не зважаючи на те, що реєстрація обов’язкова, система дозволяє встановити додаток з будь-якої точки світу. В ній існує лише один сервіс «Проблем», що дозволяє створити скаргу та дізнатися її результат на головному екрані (Рис. 2.2.2.2), або більш детально в пунктах меню «Мої проблеми».

  
Рисунок 2.2.2.2. – Головний екран

Кожну скаргу можна віднести до різних категорій (Рис. 2.2.2.3).

  
Рисунок 2.2.2.3 – Категорії скарг

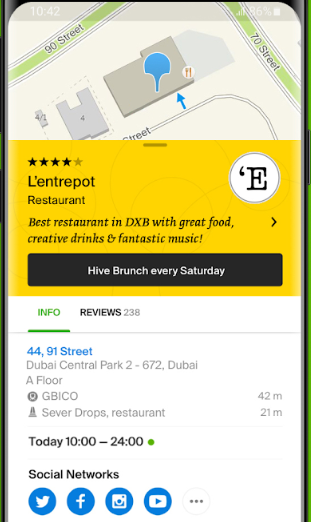
В результаті обраної категорії, відповідні служби повинні вирішувати цю проблему та надавати зворотній зв’язок.

**Переваги:** Доступність додатку та закріплення системи на рівні країни, що дозволяє вдало вирішувати проблеми та впроваджувати будь-які інші сервіси.

**Недоліки:** Система покриває лише одну потребу - скарги. Не повністю інтернаціоналізований додаток.

### Система 2GIS

2GIS - це каталог з картою та GPS навігацією в Україні та за кордоном. Додаток працює без підключення до Інтернету - завантажте міську базу даних і використовуйте її в літаку, метро або в зоні роумінгу.[3] (Рис. 2.2.3.1)

  
  
Рисунок 2.2.3.1 Приклад інтерфейсу

Система впроваджує велику кількість інформаційних сервісів: розклад роботи кафе, поштових відділеннь, лікарень або аптек; дозволяє знаходити та створювати маршрут аж до самої дверей вашого пункту призначення.

**Переваги:** працює на території України та має неперевершену карту з великою кількістю інформації. Додаток відповідає сучасному інтерфейсу.

**Недоліки:** Крім інформаційних сервісів та побудування маршруту не має нічого спільного із допомогою виправлення ситуації навколо.

1. Засоби розробки

## 3.1 Мобільний додаток

Створення мобільного додатку за допомогою Flutter Framework на мові Dart в середовищі Visual Studio Code.

### Flutter Framework

Flutter - це набір для розробки програмного забезпечення з відкритим кодом, що створений Google. Він використовується для розробки програм для Android, iOS, Windows, Mac, Linux, Google Fuchsia та Інтернету.

Перша версія Flutter була відома під кодовою назвою "Sky" і працювала в операційній системі Android. Він був оприлюднений на саміті розробників Dart 2015 року, з наміром мати можливість послідовно рендерувати зі швидкістю 120 кадрів в секунду. 11 грудня 2019 року на заході Flutter Interactive був випущений Flutter 1.12. [5]

### Dart – мова програмування

Dart - оптимізована клієнтом мова програмування для додатків на кількох платформах. Вона розроблена Google і використовується для створення мобільних, настільних, серверних та веб-додатків. Dart - це об'єктно-орієнтована мова, заснована на класах із синтаксисом у стилі C. Dart може компілювати до рідного коду чи JavaScript. [6]

### Visual Studio Code – редактор вихідного коду

Visual Studio Code - це редактор вихідного коду, розроблений Microsoft для Windows, Linux та macOS. Він включає вбудований Git та підтримку налагодження, виділення синтаксису, інтелектуальне завершення коду, фрагменти та рефакторинг коду. Він гнучко налаштовується, що дозволяє користувачам змінювати тему, комбінації клавіш, налаштування та встановлювати розширення, що додають додаткову функціональність. Вихідний код - вільний та відкритий, випущений згідно з ліцензією MIT. Скомпільовані двійкові файли безкоштовні для будь-якого використання. [7]

## Сервер

Сервер створений за допомогою ASP.NET Core Framework з використанням бази даних MS SQL.

### ASP.NET Core Framework

ASP.NET Core - це вільний Web-Framework з відкритим кодом та є спадкоємицею ASP.NET, розроблена Microsoft та громадою. Це модульна основа, яка працює як на повній .NET Framework, так і на Windows, а також на міжплатформованій .NET Core. [8]

### MS SQL – база даних

Microsoft SQL Server - це система управління реляційними базами даних, розроблена Microsoft. Як сервер бази даних, це програмний продукт з основною функцією зберігання та отримання даних, як цього вимагають інші програмні програми, який може працювати або на тому ж комп'ютері, або на іншому комп'ютері в мережі. [9]

### C# - мова програмування

C# - це мова програмування, створена спеціально для роботи у середовищі Microsoft .NET Framework.

## Технічні вимоги до програмного забезпечення сервера

Операційна система: Windows 7 SP1 або новіше (64-bit)

Дисковий простір: 1000 MB (не включаючи дисковий простір IDE/tools).

4.ОПис програмної реалізації

## 4.1 Діаграма Use Case

Проектована система представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою так званих прецедентів.

Актор - це набір ролей, які виконує користувач в ході взаємодії з деякою сутністю (системою, підсистемою, класом). Актор може бути людиною, іншою системою, підсистемою або класом, які представляють щось за межами розглянутої сутності.

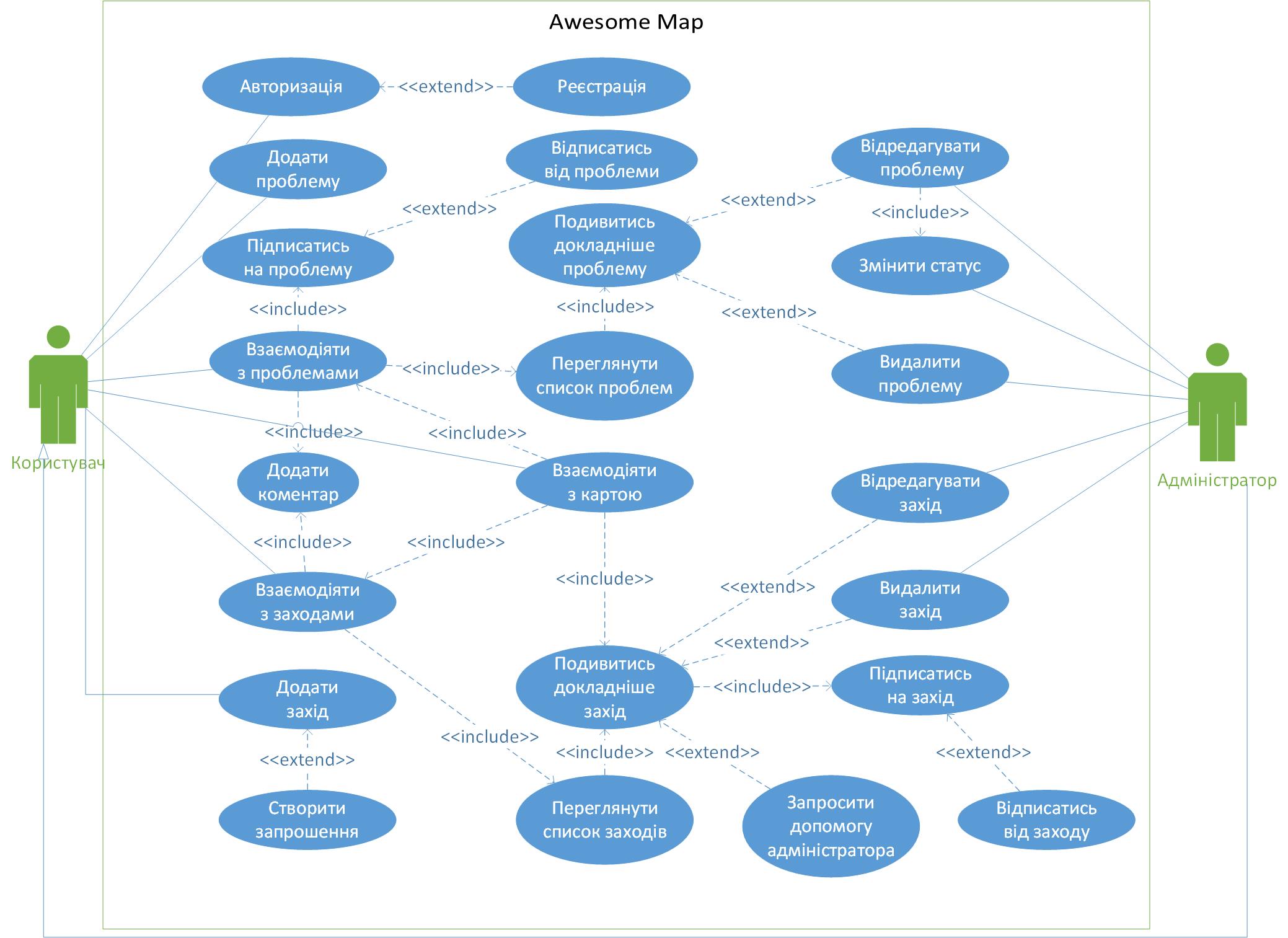
Прецедент - це опис набору послідовних подій (включаючи можливі варіанти), що виконуються системою, які призводять до результату спостережуваного Актором. Прецеденти описують сервіси, що надаються системою Актор, з якими вона взаємодіє. Причому прецедент ніколи не пояснює, "як" працює сервіс, а тільки описує, "що" робиться.

Між елементами діаграми прецедентів можуть існувати різні відносини, які описують взаємодію екземплярів акторів і варіантів використання.

Стандартні види відносин між акторами і прецедентами:

* + асоціації (association relationship);
  + розширення (extend relationship);
  + узагальнення (generalization relationship);
  + включення (include relationship).

Діаграма прецедентів для системи зображено на рис. 4.1.

  
Рисунок 4.1 – Діаграма прецедентів

Аналізуючи діаграму, в системі є дві ролі: Користувач і Адміністратор.

Користувач має змогу авторизуватись, якщо немає ще облікового запису або ж зареєструватись. Реєстрація - це розширення авторизації, а не включення, тому що користувач, який має вже обліковий запис, не повинен мати змогу знову зареєструватись.

Основні дії користувача зводяться до взаємодії з існуючими проблемами та заходами або їх створення. Якщо більш детально, то користувач має змогу додати свою проблему або підписатись на чужу проблему, що дозволить слідкувати за нею та дізнатись результат. Також реалізована можливість відписатись від проблеми, тільки, якщо користувач вже на неї підписався.

Той самий функціонал створений для заходів. Взаємодія з цими двома сутностями реалізовано за допомогою мапи або списку.

У кожному з цих варіантів користувач має змогу подивитись докладніше.

Користувач може відредагувати свою проблему, якщо ії адміністрація не почала читати, що автоматично переведе проблему в статус «В опрацюванні».

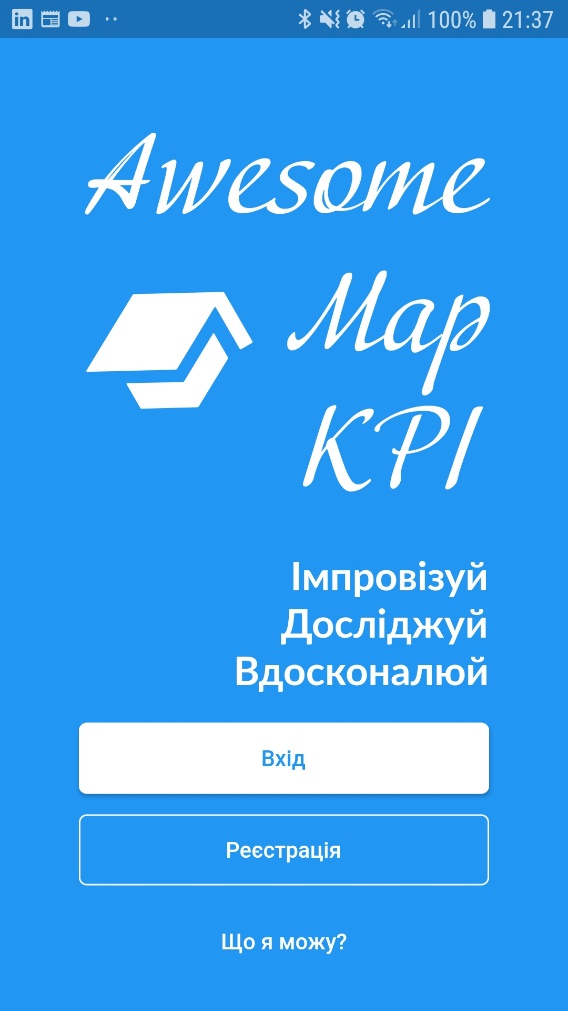
Також є змога відредагувати свій захід.

Адміністрація має всі привілегії, як і користувач, але розширює свій функціонал можливостями відредагувати проблему або змінити її статус, а також видалити проблему чи захід з системи.

## Інтерфейс мобільного додатку

### 4.2.1 Екран входу

При першому запуску, користувачу відображається екран входу та реєстрації (Рис 4.2.1) з додатковою можливістю подивитись що може мобільний додаток.

  
  
Рисунок 4.2.1 – Екран входу та реєстрації

### 4.2.2 Навігаційна панель

Навігаційна панель знизу (Рис 4.2.2) відображає з якими сутностями зараз користувач працює.

  
Рисунок 4.2.2 – Навігаційна панель знизу (виділено червоним)

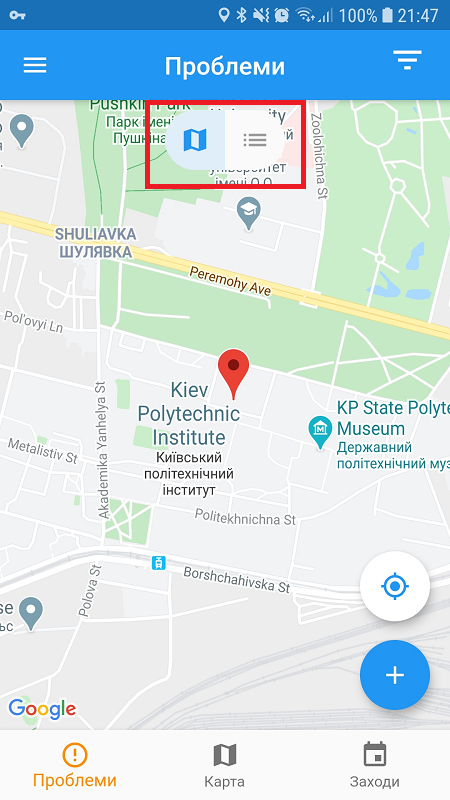
Якщо виділений пункт «Проблеми», як на рисунку 4.2.2, то на карті відображаються лише проблеми, а компонент фільтрації працює лише з ними.

На вкладці «Карта» відображаються і проблеми, і заходи одночасно.

На вкладці «Заходи» відповідно можно побачити лише заходи.

### 4.2.3 Перемикач «Карта / Список»

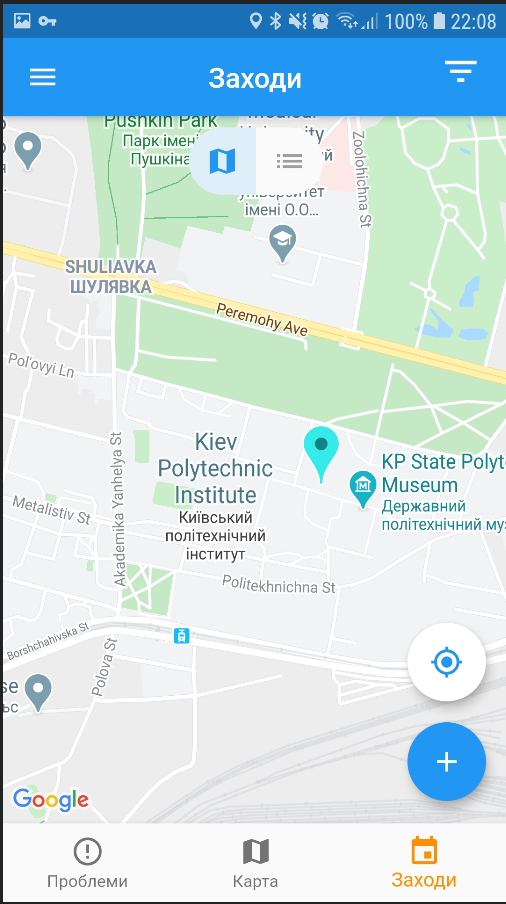
На рисунку 4.2.3 компонент перемикач «Карта / Список» виділений червоним.

  
Рисунок 4.2.3 - Перемикач «Карта / Список»

Компонент на вкладці проблем або заходів дозволяє переключитись до списку цих елементів чи знову до карти.

### 4.2.4 Компонент карта

Екран карти з проблемами (Рис 4.2.3) або з заходами (Рис 4.2.4) має ідентичні елементи управління користувальницького інтерфейсу.

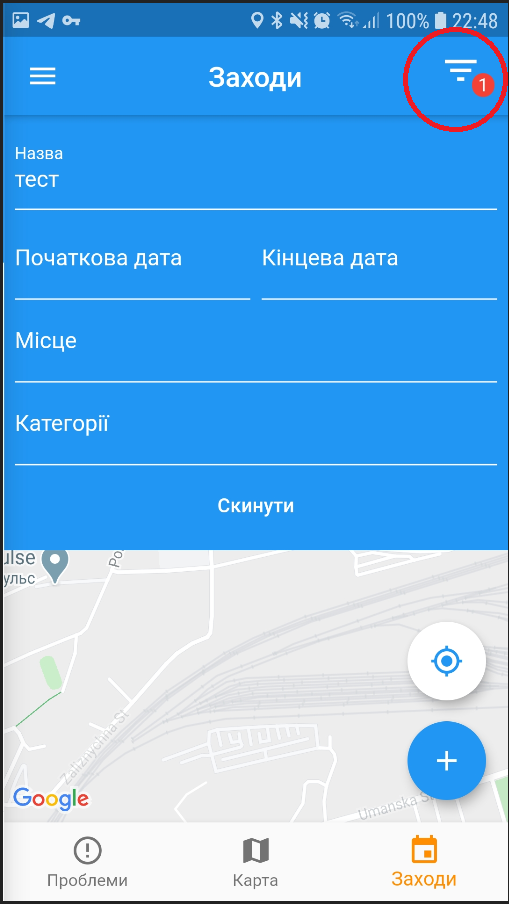
  
Рисунок 4.2.4 – Компонент карта для заходів

На цьому компоненті присутні елементи додавання («+»), пошук своєї локації та вже відомий нам перемикач «Карта / Список».

### 4.2.5 Компонент фільтрації

Компонент фільтрації був розроблений відповідно до технічного завдання з розташуванням у верхньому правому кутку для того, щоб при перемиканні з карти до списку, або навпаки, користувач очікував, що його фільтри досить застосовані.

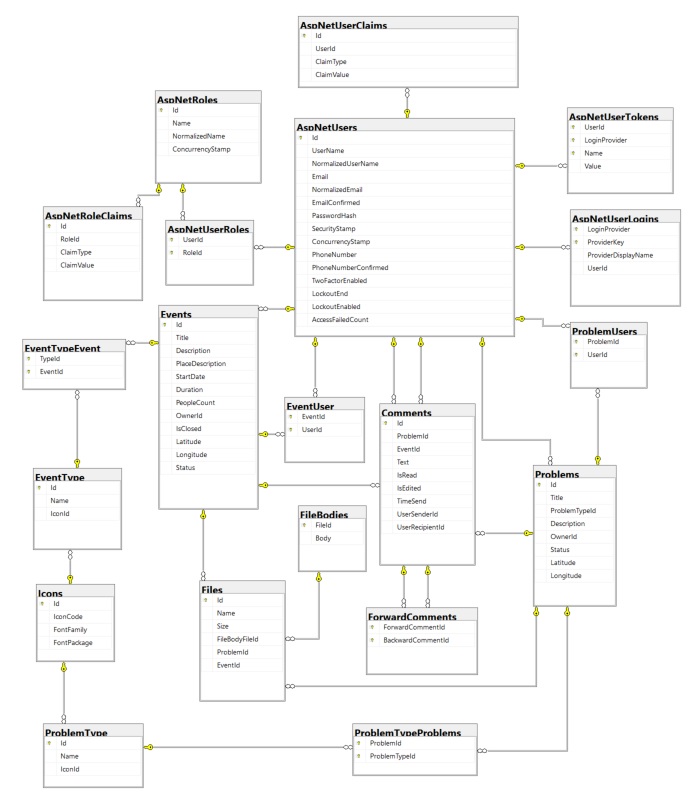
Роздивимось його детальніше на рисунку 4.2.5.

  
Рисунок 4.2.5 – Відкритий компонент фільтрації з одним заповненим елементом

Компонент фільтр має червоний символ «1», тому що заповнено одне поле для фільтрації, а саме - «Назва» із текстом «тест».

## 4.3 Схема бази даних системи

Схема бази даних для серверної частини системи наведена на рисунку 4.2.

  
Рисунок 4.2 – Схема Бази даних

На даній схемі показані зв’язки один до багатьох або один до одного через елементи: іконка ключа - один, нескінченність – багато. Треба звернути увагу на сутність Іконки (Icons), яка була створена спеціально для відображення типів проблем та заходів з відповідною іконкою, що потрібно лише для мобільного додатку. Вона приведена до базової сутності іконки у фреймворку Flutter, що дозволяє відображати будь-яку з них в додатку, якщо цей пакет іконок встановлено заздалегідь.

Також виокремлено файл та тіло файла в дві окремі таблиці (Files та FileBodies відповідно), завдяки чому система набуває легкості при вилученні тільки текстової інформації та окремого завантаження файлу в єдиному незалежному запиті від мобільного додатку.

Інсталювання системи

Для встановлення та використання системи звичному користувачу потрібно встановити мобільний додаток на телефон з Play Market, що носить назву додатку «Awesome Map». Відкрити додаток та зареєструватись.

Якщо ви розробник і намаєтесь встановити систему собі локально, то вам потрібно:

1. Завантажити вихідний код. [10]
2. Налаштувати сервер, указавши рядок підключення в application.config файлі, що розташований у корні проекту сервера.
3. Запустити сервер.
4. Налаштувати мобільний додаток, указавши нову адресу сервера у файлі за шляхом <мобільний додаток>/lib/env/dev.json.
5. Запустити мобільний додаток.

Не потрібно створювати схему бази даних, тому що сервер автоматично створює її при першому запуску.

Висновки

У період практики, я дослідив готові рішення стосовно моєї теми дипломної роботи: вирішення господарських проблем на території кампусу КПІ за допомогою GIS технологій. Кожне готове рішення має свої вади і переваги, завдяки яким я зменшив недоліки у своїй власній системі та отримав для неї мінімальні технічні вимоги. Я створив серверну та мобільну частини системи, яка вирішує задачі моніторингу наявних господарських проблем, (прорив водо-,тепло-, електричних мереж, наявність незручностей, сміття і таке інше), а також планування та проведення публічних заходів на території кампуса університету.

Адміністрація НТУУ «КПІ» ім. Ігоря Сікорського має змогу відредагувати проблему або захід, відповідно до робочих процесів системи.

Я закріпив знання побудування діаграми прецедентів та схеми бази даних, що дозволило більш вдало створити супроводжуючу документацію.

Список використаних джерел

1. Мобільний додаток «MyLa311» від City of Los Angeles [Електроний ресурс] - Режим доступу: https://play.google.com/store/apps/details?id=com.LA.MyLA311
2. Мобільний додаток «Pakistan Citizen Portal» від National IT Board, Government Of Pakistan [Електроний ресурс] - Режим доступу: https://play.google.com/store/apps/details?id=com.govpk.citizensportal
3. Мобільний додаток «2GIS» від DoubleGIS, LLC [Електроний ресурс] - Режим доступу:

https://play.google.com/store/apps/details?id=ru.dublgis.dgismobile

1. Грамотная клиент-серверная архитектура: как правильно проектировать и разрабатывать web API, Владимир, web-developer in Noveo [Електронний ресурс] – Режим доступу:  
   https://tproger.ru/articles/web-api
2. Flutter (software) [Електроний ресурс] – Режим доступу:  
   https://en.wikipedia.org/wiki/Flutter\_(software)
3. Dart (programming language) [Електроний ресурс] – Режим доступу:  
   https://en.wikipedia.org/wiki/Dart\_(programming\_language)
4. Visual Studio Code [Електроний ресурс] – Режим доступу:  
   https://en.wikipedia.org/wiki/Visual\_Studio\_Code
5. ASP.NET Core [Електроний ресурс] – Режим доступу:  
   https://dotnet.microsoft.com/apps/aspnet
6. MS SQL [Електроний ресурс] – Режим доступу:  
   https://www.microsoft.com/en-us/sql-server
7. Awesome Map – Гавриляк Олександр, вихідний код,   
   [Електроний ресурс] – Режим доступу:  
   https://github.com/defa808/awesome\_map