ЗМІСТ

[ВСТУП 4](#_Toc41764641)

[1. ПОСТАНОВКА ЗАДАЧІ 6](#_Toc41764642)

[1.1 Основні задачі 6](#_Toc41764643)

[1.2 Задачі для мобільного додатку 6](#_Toc41764644)

[1.2.1 Загальні вимоги 6](#_Toc41764645)

[1.2.1 Головні задачі 7](#_Toc41764646)

[1.2.2 Додаткові задачі 8](#_Toc41764647)

[1.3 Задачі для сервера 8](#_Toc41764648)

[1.3.1 Загальні вимоги 8](#_Toc41764649)

[1.3.2 Головні задачі 8](#_Toc41764650)

[Висновки до розділу 9](#_Toc41764651)

[2. ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ АНАЛОГІЧНИХ РІШЕНЬ 10](#_Toc41764652)

[2.1 Система My LA 311 10](#_Toc41764653)

[2.2 Система Pakistan Citizen Portal 12](#_Toc41764654)

[2.3 Система 2GIS 15](#_Toc41764657)

[Висновок до розділу 16](#_Toc41764658)

[3. ЗАСОБИ РОЗРОБКИ 17](#_Toc41764659)

[3.1 Технології для реалізації мобільного додатку 17](#_Toc41764660)

[3.1.1 Flutter Framework – крос-платформний фреймворк 17](#_Toc41764661)

[3.1.2 Dart – мова програмування 18](#_Toc41764662)

[3.1.3 Visual Studio Code – редактор вихідного коду 18](#_Toc41764663)

[3.2 Технології для реалізації серверної частини 19](#_Toc41764664)

[3.2.1 ASP.NET Core Framework – Web фреймворк від Microfost 19](#_Toc41764665)

[3.2.2 MS SQL – база даних 20](#_Toc41764666)

[3.3.3 C# - мова програмування 20](#_Toc41764667)

[3.3 Технічні вимоги до програмного забезпечення сервера 21](#_Toc41764668)

[4.ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ 22](#_Toc41764669)

[4.1 Діаграма Use Case 22](#_Toc41764670)

[4.2 Інтерфейс мобільного додатку 24](#_Toc41764671)

[4.2.1 Екран входу 24](#_Toc41764672)

[4.2.3 Перемикач «Карта / Список» 26](#_Toc41764673)

[4.2.4 Компонент карта 27](#_Toc41764674)

[4.2.5 Компонент фільтрації 27](#_Toc41764675)

[4.3 Схема бази даних системи 29](#_Toc41764676)

[Інсталювання системи 31](#_Toc41764677)

[Висновки 32](#_Toc41764678)

[Список використаних джерел 33](#_Toc41764679)

# 

ВСТУП

В наш час діджеталізації створюється потреба в моніторингу господарських проблем та публічних заходів промислового району для ефективного та вдалого створення 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. За допомогою аналізу існуючих рішень, впроваджена система розроблялась з урахування минулого досвіду, таким чином, вона позбулась найпоширеніших недоліків та підкреслила свої унікальні переваги.

Система розроблялась на базі 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

## 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 повинні бути 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 перевизначать 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 картах і на одній загальній.

В системі повинна бути реалізована фільтрація проблем за наступними показниками:

* назва;
* дата створення (початок проміжку до сьогодні);
* категорія;
* статус проблеми.

та заходів за іншими показниками:

* назва;
* дата проведення з (початок обраного проміжку);
* дата проведення до (кінець обраного проміжку);
* місце;
* категорії.

Доцільно зробити перехід від міток на карті 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

### 1.3.1 Загальні вимоги

Сервер має опрацьовувати http запити та повертати коректні відповіді.

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

* вилучення даних – GET запит;
* збереження даних – POST запит;
* зміна даних – PUT або PATCH запит;
* видалення – DELETE запит.

### 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;
* типи заходів;
* коментарі;
* іконки;
* файли і таке інше.

## Висновки до розділу

В цьому розділі були розглянуті загальні вимоги до сучасного мобільного додатку. В першій частині впроваджені головні та додаткові вимоги, які повинен вирішувати мобільний додаток в рамках робочих процесів системи. В другій частині розділу, розглянуті вимоги до архітектури серверної частини та загальна роль сервера в системі «Awesome Map».

1. ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ АНАЛОГІЧНИХ РІШЕНЬ

### 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] Використовування цього додатку дає змогу швидкого та легкого доступу до найпопулярніших послуг міста, включаючи видалення графіті, ремонт вибоїн та забір об’ємних предметів.

Роздивимось більш детально мобільний додаток.  
На першому екрані (Рис 2.1.1) бачимо три пункти:

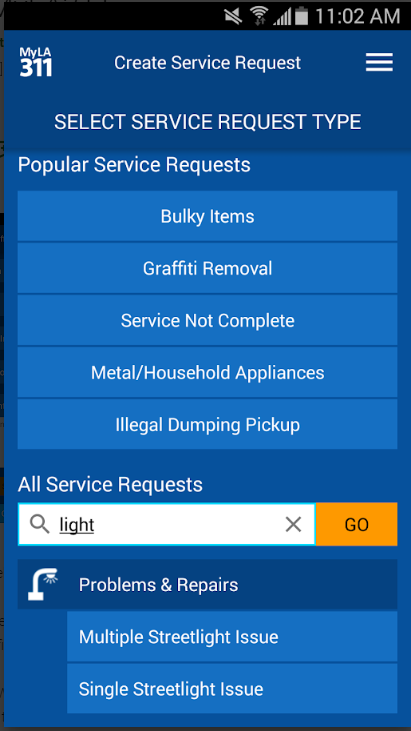
* створення службового запиту;
* перегляд;
* інші служби.

  
Рисунок 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 2.1.2). Для цього навіть впроваджено пошук.

  
Рисунок 2.1.2 – Пошук за доступними сервісами в системі

Завдяки сервісам можливо:

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

Переваги:

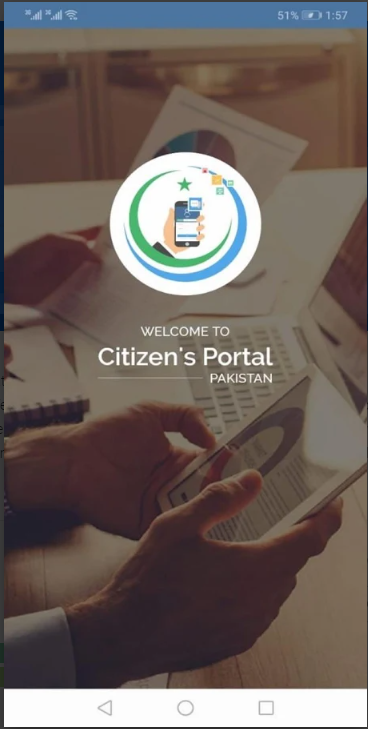
* доступ до великої цифрової системи міста, що має багато сервісів;
* зрозумілий та простий варіант інтерфейсу;
* можливість залишити мітку анонімно.

Недоліки:

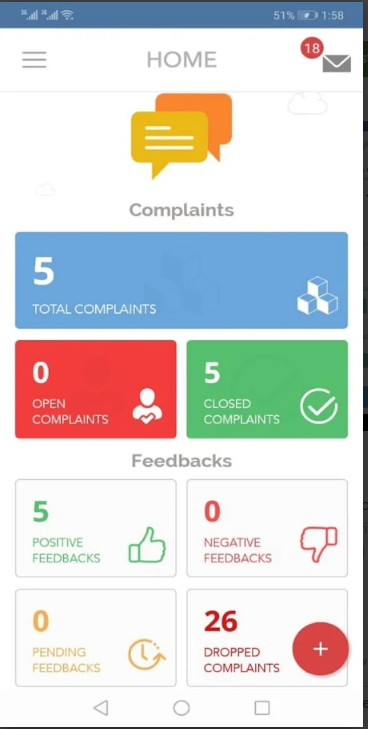
* застарілий інтерфейс, що не відповідає новітнім вимогам для Android та IOS користувачів;
* доступ до встановлення додатку можна отримати лише з території Сполучених Штатів Америки або потрібно використовувати VPN сервіси.

### Система Pakistan Citizen Portal

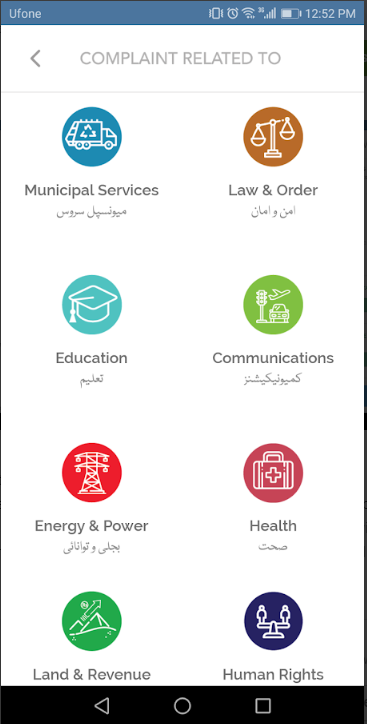
Портал громадянина Пакистану – це інтегрована система розміщення скарги громадян, яка з’єднує всі урядові організації як на федеральному, так і на провінційному рівнях. (Рис. 2.2.1) Система служить носієм скарг у відповідні офіси по всьому Пакистану. Додаток слугує додатковим каналом між громадянином та владою. [2]

  
Рисунок 2.2.1 – Екран привітання Pakistan Citizen Portal

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), або більш детально в пунктах меню «Мої проблеми».

  
Рисунок 2.2.2. – Головний екран

Кожну скаргу можна віднести до різних категорій (Рис. 2.2.3).

  
Рисунок 2.2.3 – Категорії скарг

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

Переваги:

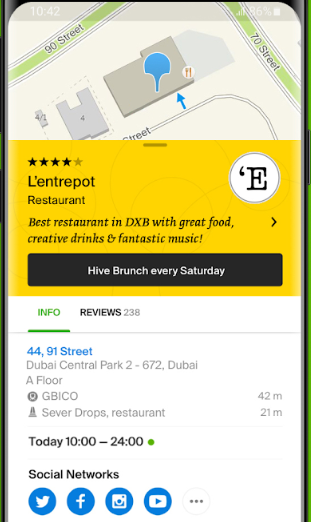
* доступність додатку по всьому світу;
* закріплення системи на рівні країни, що дозволяє вдало вирішувати проблеми та впроваджувати будь-які інші сервіси.

Недоліки:

* система покриває лише одну потребу – скарги;
* не повністю інтернаціоналізований мобільний 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. 2.3.1)

  
  
Рисунок 2.3.1 Приклад інтерфейсу

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

Переваги:

* працює на території України;
* неперевершена карта з великою кількістю інформації;
* додаток відповідає сучасному інтерфейсу Android та IOS системам.

Недоліком крім інформаційних сервісів та побудування маршруту є те, що немає нічого спільного із допомогою виправлення ситуації навколо.

## Висновок до розділу

В цьому розділі було розглянуто три популярних рішення на базі GIS систем, та виділено їх недоліки на переваги. Їх порівняння відображено у таблиці 2.1.

Таблиця 2.1 – Порівняння альтернативних рішень

|  |  |  |  |
| --- | --- | --- | --- |
| Назва  Критерії | LA311 | Pakestan Citizen Portal | 2GIS |
| Сучасний інтерфейс | - | - | + |
| Підтримка державними органами влади | + | + | - |
| Різноманітний функціонал використання | + | - | + |
| Використання на території України | - | - | + |
| Зручна робота з картою | - | - | + |

Потрібна зауважити стосовно останнього пункту, дійсно зручна робота з картою зроблена лише в останньому рішенню на GIS технологіях – 2GIS, тому що тільки для цього рішення це є основним функціоналом системи, з чим стикається користувач кожну секунду проведену у додатку. На відміну від двох інших рішень, де карта є додатковим інструментом до основних робочих процесів систем.

# 3. ЗАСОБИ РОЗРОБКИ

### 3.1 Технології для реалізації мобільного додатку

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

### Flutter Framework – крос-платформний фреймворк

Flutter - це портативний інструментарій для користувальницького інтерфейсу Google для створення красивих програм, створеня мобільних пристроїв, Інтернет застосунків та настільних ПК з однієї бази коду. Flutter працює з існуючим кодом, використовується розробниками та організаціями по всьому світу і є безкоштовним та відкритим кодом.

Для користувачів Flutter реалізує прекрасні інтерфейси додатків.

Для розробників Flutter знижує смугу вступу для створення програм. Це прискорює розробку додатків і зменшує витрати та складність виробництва додатків на різних платформах.

Для дизайнерів Flutter допомагає досягти оригінального дизайнерського бачення, не втрачаючи якості чи компромісів. Він також виступає як продуктивний інструмент прототипування.

Flutter призначений для розробників, які шукають швидкий спосіб створення красивих додатків або спосіб залучення більшої кількості користувачів за допомогою однієї інвестиції.

Flutter також призначений для інженерних менеджерів, які керують розробниками. Flutter дозволяє менеджерам створювати єдину команду розробників додатків для мобільних пристроїв, Інтернету та настільних комп'ютерів, об'єднуючи свої інвестиції в розвиток, щоб швидше надсилати більше функцій, перевозити одну і ту ж функцію на декілька платформ одночасно і знижувати витрати на обслуговування.

Flutter також призначений для дизайнерів, які хочуть, щоб їх оригінальні дизайнерські бачення постачалися послідовно, з високою якістю, всім користувачам. Насправді CodePen тепер підтримує Flutter. [5]

По суті, Flutter призначений для користувачів, які хочуть красивих додатків із чудовим рухом та анімацією, а також користувацькі інтерфейси з персонажем.

Flutter доступний для програмістів, знайомих з об'єктно-орієнтованими поняттями (класи, методи, змінні тощо) та імперативними концепціями програмування (циклі, умовні умови тощо).

Для вивчення та використання Flutter не потрібен попередній досвід.

Ми бачили, як люди з дуже невеликим досвідом програмування навчаються та використовують Flutter для створення прототипів та розробки додатків.

### 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. [6]

### Visual Studio Code – редактор вихідного коду

Visual Studio Code - це спрощений редактор коду з підтримкою таких операцій розвитку, як налагодження, виконання завдань та контроль версій. Він спрямований на надання лише інструментів, необхідних розробнику для швидкого циклу збирання-налагодження коду, а також залишає більш складні робочі процеси для більш повних представлених IDE, таких як ID Visual Studio. Код VS працює на macOS, Linux та Windows. [7]

## Технології для реалізації серверної частини

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

### ASP.NET Core Framework – Web фреймворк від Microfost

ASP.NET Core - це web framework з відкритим кодом, створений Microsoft, для створення сучасних веб-додатків та служб за допомогою .NET.

ASP.NET є кросплатформеним і працює на Windows, Linux, macOS та Docker. Під час використання ASP.NET серверний код, такий як бізнес-логіка та доступ до даних, записується за допомогою C#, F# або Visual Basic.

Оскільки ASP.NET розширює .NET, можна використовувати велику екосистему пакетів і бібліотек, доступних усім розробникам .NET. Також можливо створити власні бібліотеки, якими є змога поділитися між будь-якими програмами, написаними на платформі .NET. [8]

### 3.2.2 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 # (вимовляється "Сі Шарп") - сучасна, об'єктно-орієнтована і безпечна мова програмування. C# має своє коріння в сімействі мов C і буде одразу знайома програмістам C, C ++, Java та JavaScript. C# - об'єктно-орієнтована мова, але C# додатково включає підтримку програмованого орієнтування на компоненти. Сучасний дизайн програмного забезпечення все більше покладається на програмні компоненти у вигляді автономних та самоописуючих пакетів функціональності. Ключовим для таких компонентів є те, що вони представляють модель програмування із властивостями, методами та подіями. Вони мають атрибути, які надають декларативну інформацію про компонент. Вони містять власну документацію. C# надає мовні конструкції для прямої підтримки цих концепцій, роблячи C# природною мовою, для створення та використання програмних компонентів.

Кілька функцій C# допомагають створювати надійні та довговічні програми. Збір сміття автоматично відновлює пам'ять, зайняту недоступними невикористаними об'єктами. Обробка винятків забезпечує структурований та розширюваний підхід до виявлення та відновлення помилок. Безпечно типова конструкція мови не дає змоги читати з неініціалізованих змінних, індексувати масиви за їх межами або виконувати некоректні приведення типів.[10]

## 3.3 Технічні вимоги до програмного забезпечення сервера

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

Процесор: x86 or x64

RAM : 512 MB (мінімум), 1 GB (рекомендовано)

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

## Висновок до розділу

# 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.

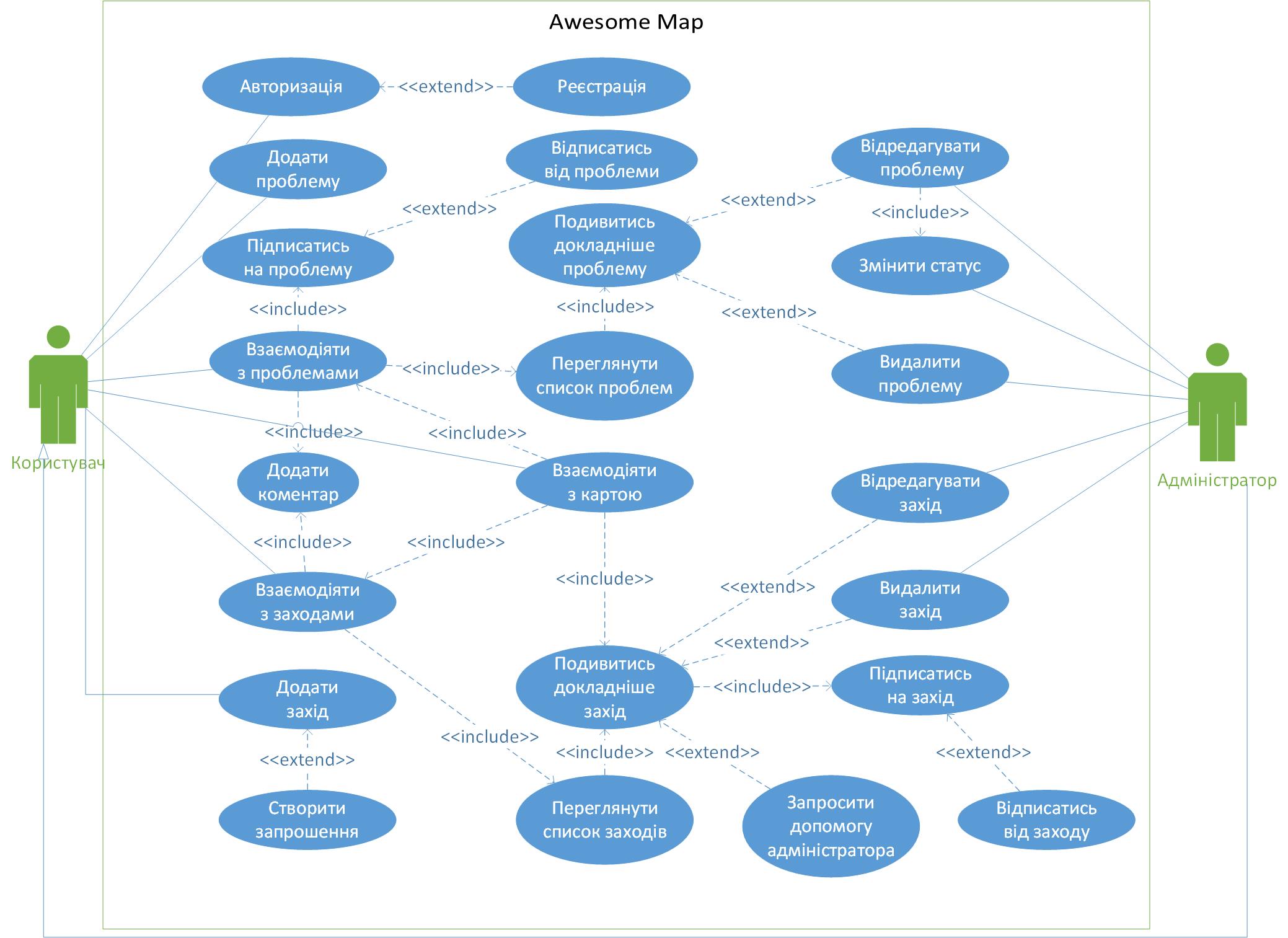
Прецедент - це опис набору послідовних подій (включаючи можливі варіанти), що виконуються системою, які призводять до результату спостережуваного Актором. Прецеденти описують сервіси, що надаються системою Актор, з якими вона взаємодіє. Причому прецедент ніколи не пояснює, "як" працює сервіс, а тільки 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 зареєструватись, якщо немає ще облікового запису або ж авторизуватись. 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 карти або списку.

У кожному з цих варіантів користувач має змогу подивитись докладніше.

Користувач може відредагувати свою проблему, якщо її адміністрація не почала читати, що автоматично переводить проблему в статус «В опрацюванні» і анулює дозвіл на редагування.

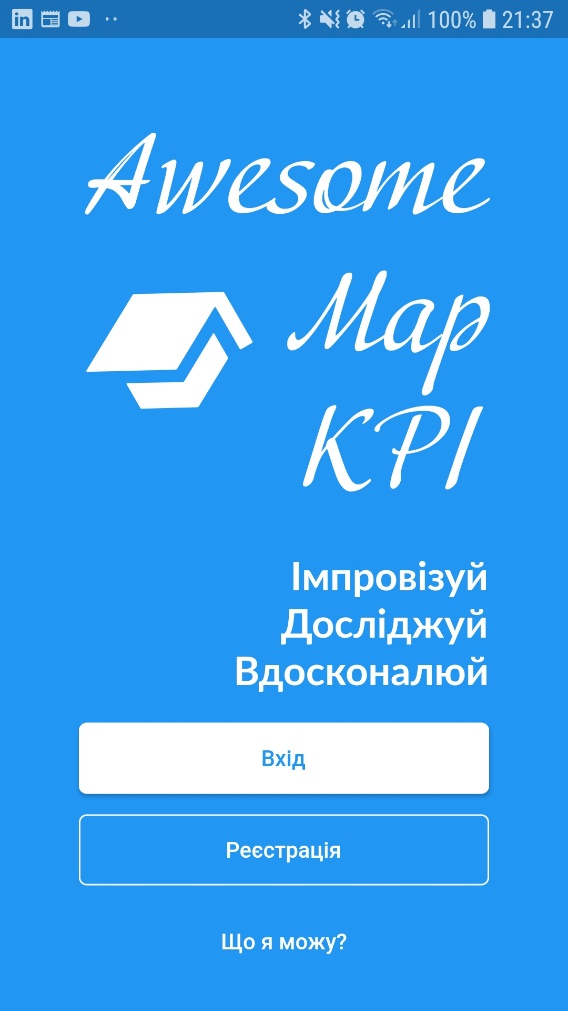
На відміну від проблем, у користувача є змога відредагувати свій захід завжди.

Адміністрація має всі привілеї, 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

В розділі «Що я можу» гість системи може продивитись зображення в GIF-форматі, де проілюстровано, що може робити система, наприклад:

* додати проблему;
* перегляд інших проблем;
* коментування проблем;
* додати захід;
* перегляд заходів;
* коментування заходів;
* фільтрування заходів або проблем.

Більш детально в репозиторії «Awesome Map» [11].

Після авторизації користувач потрапляє на головний екран додатку. (Рис. 4.2.2), де має доступ до головного функціоналу.

Навігаційна панель знизу, що виділена червоним на рисунку 4.2.2 відображає з якими сутностями зараз користувач працює.

  
Рисунок 4.2.2 – Головний екран додатку

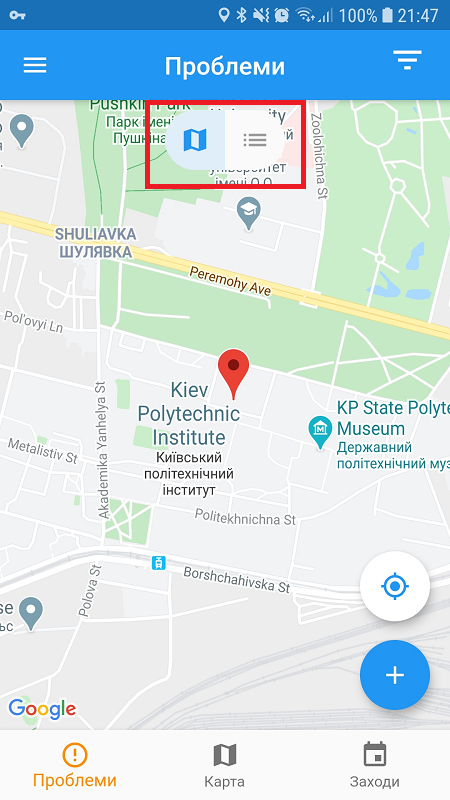
Якщо виділений пункт «Проблеми», як на рисунку 4.2.2, то на карті відображаються лише проблеми, а компонент фільтрації працює лише з ними.

На вкладці «Карта» відображаються і проблеми, і заходи одночасно, що дає змогу користувачу зосередитись на карті та виокремити територію, яка йому цікава.

На вкладці «Заходи» відповідно можна побачити лише заходи.

### 4.2.3 Перемикач «Карта / Список»

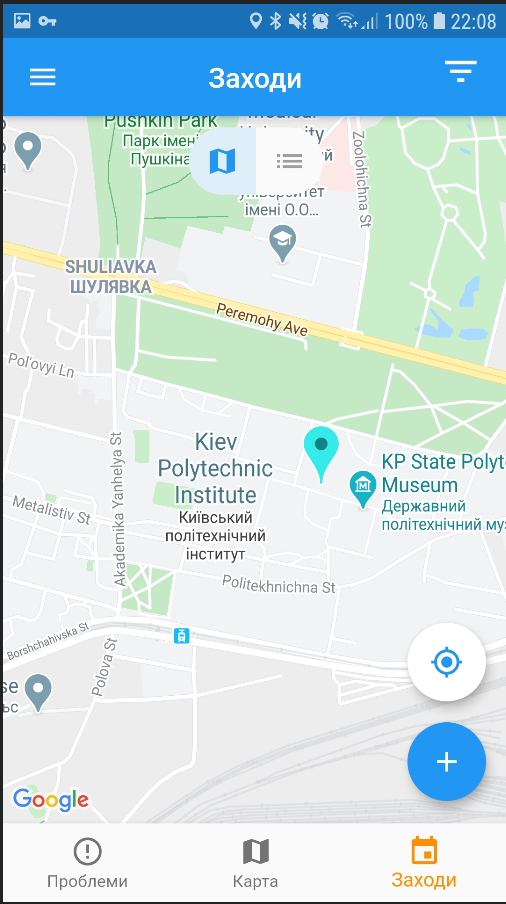
На рисунку 4.2.3 компонент перемикач «Карта / Список» виділений червоним.

  
Рисунок 4.2.3 - Перемикач «Карта / Список»

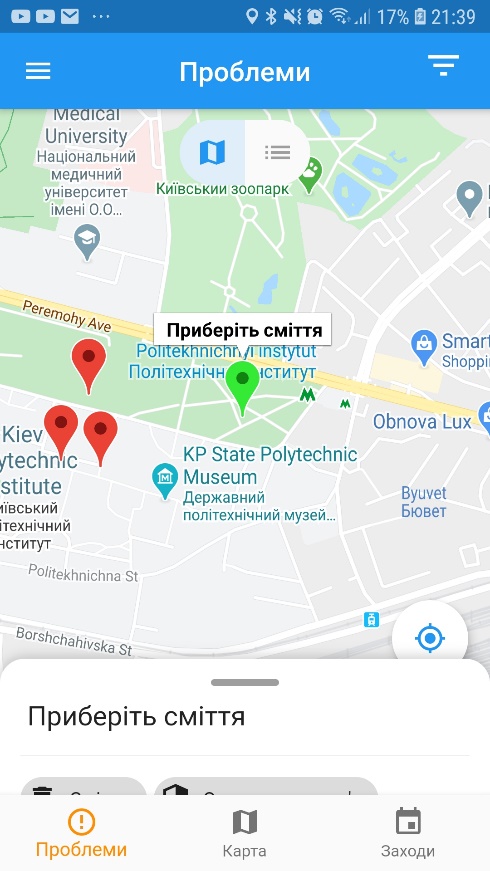
Компонент дозволяє переключитись до списку елементів, що відображаються на карті, або повернутись знову зі списку до карти.

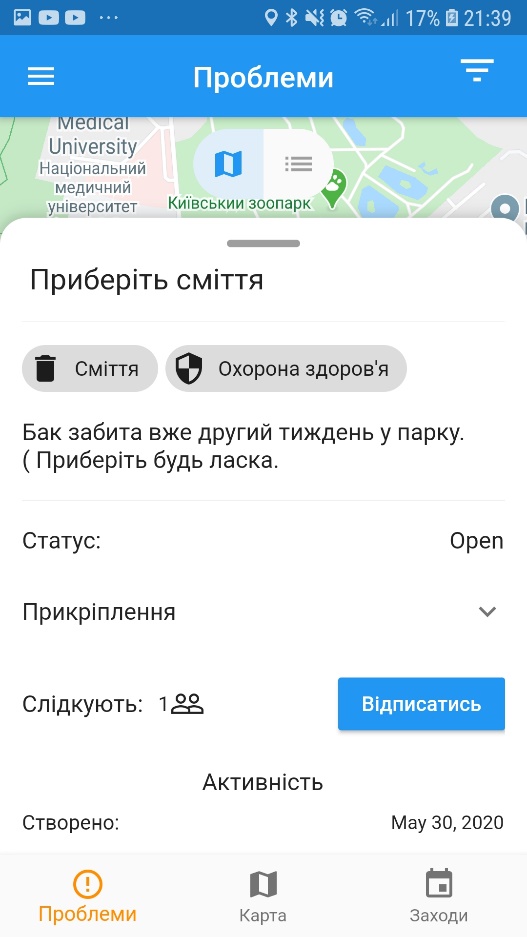
### 4.2.4 Компонент карта

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 перемикач «Карта / Список». Компонент дозволяє натиснути на мітку та отримати більш детальну інформацію про мітку. (Рис. 4.2.5 – Рис 4.2.6)

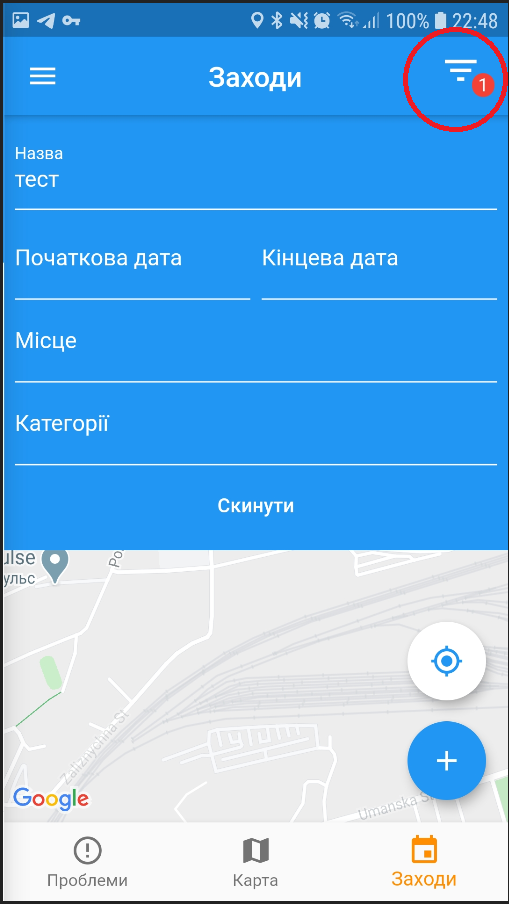
  
Рисунок 4.2.5 – Обрана мітка проблеми з детальним компонентом

  
Рисунок 4.2.6 – Детальний компонент в розгорнутому стані

### 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.

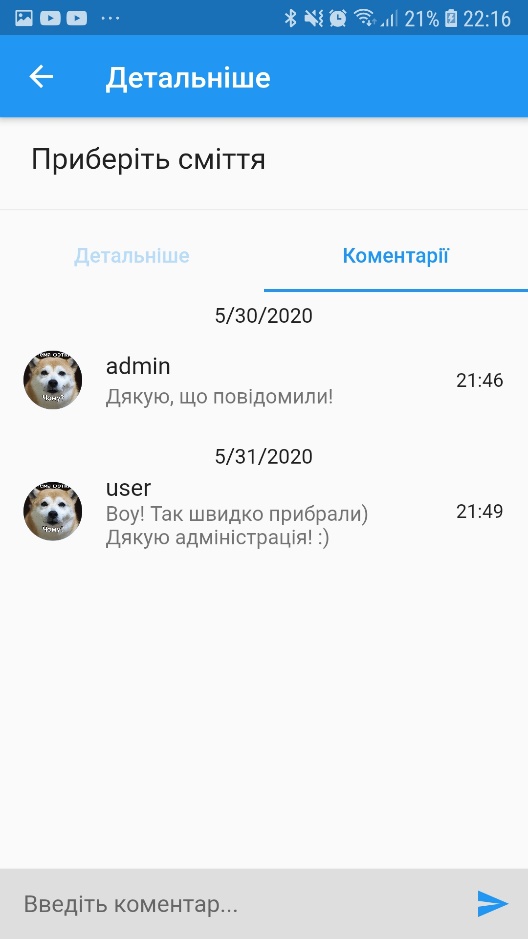
Роздивимось його детальніше на рисунку 4.2.5.

  
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».

### 4.2.6 Компонент коментарів

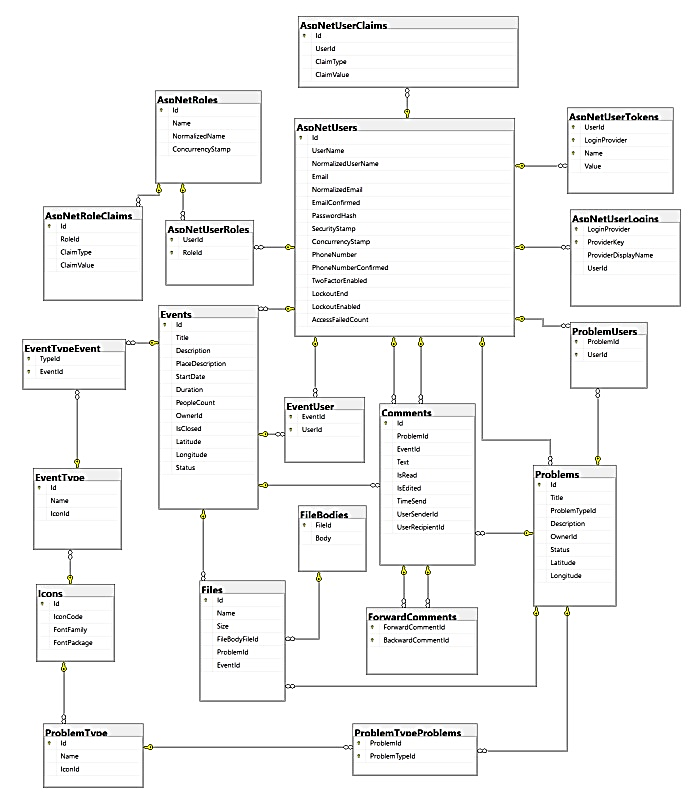
Компонент коментарів інтегровано в кожен компонент деталізації проблеми або захода для того, щоб дізнатися більше інформації про цю сутність. (Рис 4.2.6)

  
Рисунок 4.2.6 – Компонент коментарів

Насамперед, компонент можна побачити лише в деталізаціях сутностей, це зроблено щоб не розсіювати увагу користувача в інших компонентах, а керувати нею тільки, коли користувач дійсно зацікавився проблемою або заходом.

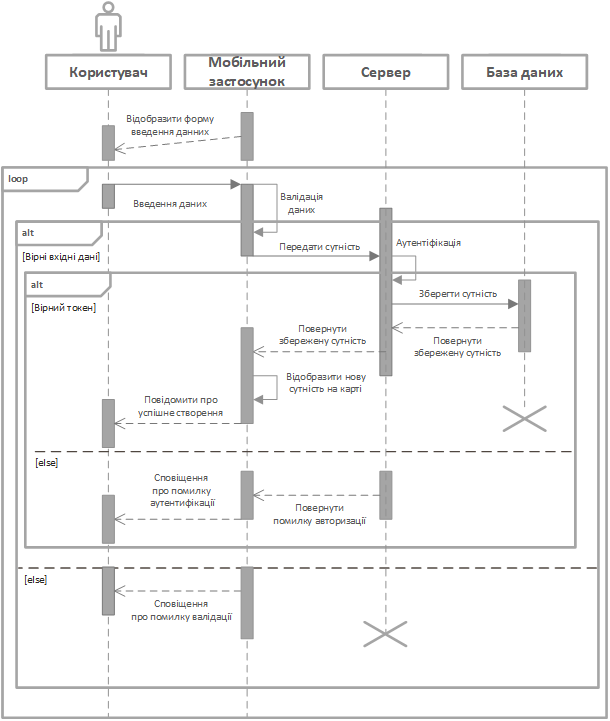
## 4.3 Схема бази даних системи

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 від мобільного додатку в єдиному незалежному запиті.



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 [Електронний 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 [Електронний ресурс] - Режим доступу: https://play.google.com/store/apps/details?id=com.govpk.citizensportal
3. Мобільний додаток «2GIS» від DoubleGIS, LLC [Електронний 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 FAQ [Електронний ресурс] – Режим доступу:  
   https://flutter.dev/docs/resources/faq
3. Dart (programming language) [Електронний ресурс] – Режим доступу:  
   https://dart.dev/
4. Visual Studio Code [Електронний ресурс] – Режим доступу:  
   https://code.visualstudio.com/docs/supporting/faq
5. ASP.NET Core [Електронний ресурс] – Режим доступу:  
   https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet
6. MS SQL [Електронний ресурс] – Режим доступу:  
   https://www.microsoft.com/en-us/sql-server
7. A tour of the C# language [Електронний ресурс] – Режим доступу:  
   https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/
8. Гавриляк О. В. Awesome Map [Електронний ресурс] / Олександр Володимирович Гавриляк. – 2020. – Режим доступу до ресурсу: https://github.com/defa808/awesome\_map.