Міністерство освіти і науки України

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмування та обчислювальної техніки

Курсова робота

з дисципліни «Бази даних»

на тему:

**«Розроблення комп’ютерної системи підвищення ефективності надання первинної медичної допомоги при пораненні військовослужбовця (“Кнопка життя”). Розроблення мобільних додатків»**

**Науковий керівник:**

Бичков О. С.

**Виконавець:**

студент 3 курсу

групи ПІ-31

Савчин Дмитро

**Спеціальність** 6.050103

**Реферат**

Пояснювальна записка до курсової роботи на тему «Розроблення комп’ютерної системи підвищення ефективності надання первинної медичної допомоги при пораненні військовослужбовця (“Кнопка життя”)»: 54 с., 5 рис., 9 інформаційних джерел, 6 додатків.

Ключові слова: VISUAL STUDIO, ANDROID, XAMARIN, БАЗИ ДАНИХ, MYSQL, МОБІЛЬНА РОЗРОБКА, ПЕРВИННА МЕДИЧНА ДОПОМОГА.

**Об’єкт розробки** – 3 мобільні застосунки під ОС Android: «сигнальна кнопка бійця», «сигнальна кнопка товариша за зброєю», «сигнальна кнопка фельдшера».

**Мета роботи** – зменшити кількість смертних випадків, що трапляються до надання медичної допомоги у стаціонарному шпиталі.

**Основний результат** – завдяки використанню комп’ютерної системи буде суттєво зменшено кількість смертних випадків, що трапляються до надання медичної допомоги у стаціонарному шпиталі.

**Результати роботи** можуть бути використані для впровадження у збройних силах чи інших структурах, які можуть і яким потрібно вчасно реагувати на певні події у людей.

**Розробка та дослідження** проводилися під управлінням ОС Windows 10. Розробка програми проводилася у середовищі Visual Studio 2015, на мові програмування C#, версія 6.0; з використанням Xamarin Framework v15.3, Xamarin.Android v7.4.0.2. Тестування проводилося на ОС Android усіх актуальних версій (від 4.1 (API 16) до 6.0.1 (API 23)).

Фреймворк Xamarin був обраний для легшої взаємодії з серверною частиною шляхом .Net та можливістю переносимості між мобільними операційними системами.

**Завдання**

Необхідно написати три мобільних програми, які будуть встановлені на мобільному пристрої бійця.

Перша програма - сигнальна кнопка бійця. При поранені боєць подає сигнал про поранення, час та місцезнаходження бійця, а також його особисті дані. Тому програма при інсталяції вводить усі особисті дані бійця. При натисканні на сигнальну кнопку програма зчитує координати з GPS, час і передає цю інформацію (звіт) на сервер польового шпиталю.

Друга програма - сигнальна кнопка товариша за зброєю. При запуску цієї програми вона фотографує поранення, зчитує координати з GPS, час і передає цю (включаючи особисту інформацію про бійця) інформацію (звіт) на сервер польового шпиталю.

Третя програма - сигнальна кнопка фельдшера. При запуску програми вона фотографує картку бійця, зчитує координати з GPS, час і передає цю (включаючи особисту інформацію про бійця) інформацію (звіт) на сервер стаціонарного шпиталю.

Кожен звіт, що відправляється на сервер, має заноситись до бази даних, що пов’язана з базою даних військових по номеру значка бійця.

Викладач Супрун О. М. Підпис

Завідувач кафедри Бичков О. С. Підпис

**Зміст**

**Вступ5**

**1 Проектування програми «Сигнальна кнопка бійця»7**

1.1 Специфікація програми «Сигнальна кнопка бійця» 7

1.2 Спільні дані програм15

1.3 Спільні методи мобільних програм16

1.3.1 Об’єднаний постачальник місцезнаходження16

1.4 Робота мобільної програми «Сигнальна кнопка бійця» 17

**2** **Проектування програм «Сигнальна кнопка товариша за зброєю» та «Сигнальна кнопка фельдшера» 19**

2.1 Специфікація програми «Сигнальна кнопка товариша»19

2.2 Специфікація програми «Сигнальна кнопка фельдшера»23

2.3 Робота з камерою27

2.3.1 Абстракція від апаратних засобів27

2.3.2 Перевірка фактичної наявності інтерфейсу камери у апарату28

2.3.3 Створення директорії для зберігання фото28

2.3.4 Відкриття інтерфейсу камери28

2.3.5 Обробка результату камери28

2.3.6 Завантаження знімку на сервер28

2.4 Робота мобільних програм «Сигнальна кнопка товариша за зброєю» та «Сигнальна кнопка фельдшера»29

**3 Проектування бази даних31**

3.1 Схема бази даних31

3.2 Таблиця Combatants31

3.3 Таблиця LocationInfo31

**Висновок32**

**Бібліографічний список33**

Додаток 134

Додаток 236

Додаток 337

Додаток 441

Додаток 545

Додаток 650

**Вступ**

Актуальність проблеми впровадження інформаційних систем в охорону здоров’я визначається, перш за все, необхідністю підвищення ефективності процесів управління охороною здоров'я, якості наданої населенню медичної допомоги. Особливо це стосується таких місць невідкладної допомоги, як території бойових дій, де швидкість реакції та прийняття рішень відіграє першочергову роль. Більше 80% поранених не встигають отримати професійну допомогу через затримки під час процесів транспортування та підготовки до отримання пораненого, неготовність до певних видів ушкоджень при наданні допомоги.

Лікувально-евакуаційні заходи є найважливішою складовою частиною медичного забезпечення бойових дій військ. Вони включають своєчасний розшук і збір поранених і хворих на полі бою, послідовне і спадкоємне надання їм медичної допомоги в поєднанні з евакуацією на ті етапи, де буде забезпечено ефективне їх лікування та якнайшвидше відновлення.

Принципи функціонування комп’ютерної системи “Кнопка життя”:

1. Боєць при поранені (якщо має змогу) натискає “кнопку” на мобільному пристрої і подає сигнал на сервер про своє поранення. При фізичної неможливості зробити це йому допомагає товариш за зброєю.

2. Товариш за зброєю надає першу медичну допомогу (наприклад, накладає джгут). Після цього фотографує місце поранення і відправляє повідомлення на сервер із вказівкою: часу накладання джгуту, координати, ПІБ пораненого бійця, номер військової частини та інше. Ця інформація формується автоматично після натискання однієї “кнопки” на мобільному пристрої.

3. Далі бійцю медичну допомогу надає фельдшер або санітар. Він фіксує відповідні дані у особистій картці бійця.

4. Фельдшер (санітар) фотографує картку і відправляє її іншу інформацію (часу надання послуг, координати, ПІБ пораненого бійця, номер військової частини та інше) на сервер для того щоб у польовому шпиталі було відомо про характер поранення та яку першу медичну допомогу було надано й коли. Після цього боєць відправляється до польового шпиталю.

5. Після доставки бійця до польового шпиталю черговий лікар приймає рішення: надавати медичну допомогу чи (за відсутністю необхідного обладнання або необхідних фахівців) доставити пораненого бійця у спеціалізований шпиталь. У другому випадку черговий лікар (або допоміжний персонал) заповнює медичну картку бійця (частина карти вже автоматично заповнена після комп’ютерного розпізнавання картки бійця, що була отримана раніше від фельдшера) і відправляють її до спеціалізованого шпиталю.

6. Спеціалізований шпиталь, після отримання картки, має змогу заздалегідь підготуватися до проведення хірургічного втручання або надання іншої медичної допомоги.

**1 Проектування програми «Сигнальна кнопка бійця»**

* 1. **Специфікація програми «Сигнальна кнопка бійця»**

**1 Вступ**

* 1. *Призначення*

Назва програми: Сигнальна кнопка бійця

Скороченна назва: Кнопка бійця

Версія програмного продукту: 1.0.0

* 1. *Угоди, прийняті в документах*

Зауваження в тексті виділяються курсивом.

Терміни та ключові слова виділяються жирним шрифтом.

* 1. *Передбачуванна аудиторія і рекомендації з читання*

Специфікація призначенна для розробників ПЗ та замовників.

* 1. *Границі проекту*

**Мета роботи** – полегшити слідкування за місцеположенням бійців у зоні бойових дій.

**Основний результат** – завдяки використанню мобільної програми буде суттєво зменшено кількість смертних випадків, що трапляються через невідомість місцезнаходження постраждалого у зоні бойових дій.

Результати роботи можуть бути використані для впровадження у збройних силах чи інших структурах, які можуть і яким потрібно вчасно реагувати на певні події у людей.

* 1. *Посилання*

1. Посилання на стандарти інтерфейсу:

<https://developer.android.com/design/material/index.html>

1. Посилання на стандарти якості:

<https://developer.android.com/develop/quality-guidelines/core-app-quality.html>

1. **Загальний опис**
   1. *Загальний погляд на продукт*

При поранені боєць подає сигнал про поранення, час та місцезнаходження бійця, а також його особисті дані. Тому програма при інсталяції вводить усі особисті дані бійця. При натисканні на сигнальну кнопку програма зчитує координати з GPS, час і передає цю інформацію на сервер польового шпиталю.

Програма встановлюється на мобільному пристрої бійця.

* 1. *Особливості (функціональність) продукту*

Програмамний продукт володіє наступними функціями:

* Збереження інформації бійця
* Збір інформації про місцезнаходження бійця та час
* Відправлення **звіту** *(інформації про бійця)* на сервер

Вербальний опис діаграми прецедентів:

Специфікація прецеденту «*Внесення особистої інформації*»:

1. **Короткий опис:** Боєць вносить особисту інформацію в програму.
2. **Суб’єкти:** Боєць.
3. **Передумови:** відсутні.
4. **Основний потік:** Особиста інформація бійця (ПІБ, військова частина та номер значка) зберігається у програмі та доступна для читання з неї та пов’язаних програм, та для відправлення як частина **звіту**.
5. **Альтернативні потоки:** відсутні.
6. **Постумови:** Особиста інформація бійця збережена у програмі.

Специфікація прецеденту «*Відправлення звіту*»:

1. **Короткий опис:** Боєць заходить у програму та автоматично відправляється **звіт**.
2. **Суб’єкти:** Боєць.
3. **Передумови:** Наявність інтернету, GPS, введена особиста інформація бійця у програмі.
4. **Основний потік:** Особиста інформація бійця завантажується з пам’яті, місцезнаходження та час беруться з даних телефону, відправляються на сервер у вигляді **звіту**.
5. **Альтернативні потоки:**

А1)Особиста інформація не введена. **Звіт** відправляється без неї.

А2) Мережа інтернет відсутня. Бійцю пропонується зайти у налаштування, щоб виправити це.

А3) Сигнал GPS відсутній. Бійцю пропонується зайти у налаштування, щоб виправити це.

1. **Постумови:** **Звіт** відправлений на сервер.

Специфікація прецеденту «*Отримання особистої інформації*»:

1. **Короткий опис:** Завантаження особистої інформації бійця.
2. **Суб’єкти:** Програма.
3. **Передумови:** Введена особиста інформація бійця.
4. **Основний потік:** Програма завантажує з пам’яті особисту інформацію бійця.
5. **Альтернативні потоки:**

А1)Особиста інформація не введена. **Звіт** відправляється без неї.

1. **Постумови:** Особиста інформація готова для додавання в **звіт**.

Специфікація прецеденту «*Отримання часу і дати*»:

1. **Короткий опис:** Одержання часу і дати.
2. **Суб’єкти:** Програма.
3. **Передумови:** Відсутні.
4. **Основний потік:** Програма одержує час і дати з даних ОС телефону.
5. **Альтернативні потоки:** відсутні.
6. **Постумови:** Час і дата готові для додавання в **звіт**.

Специфікація прецеденту «*Отримання місцезнаходження*»:

1. **Короткий опис:** Одержання місцезнаходження.
2. **Суб’єкти:** Програма.
3. **Передумови:** Наявність GPS.
4. **Основний потік:** Програма одержує місцезнаходження з геолокації телефону.
5. **Альтернативні потоки:**

А1)Сигнал GPS відсутній. Бійцю пропонується зайти у налаштування, щоб виправити це.

1. **Постумови:** Місцезнаходження бійця готове для додавання в **звіт**.

*2.3 Класи і характеристики користувачів*

|  |  |
| --- | --- |
| Клас користувачів | Характеристика |
| Боєць | Вводить особисту інформацію у програму, відправляє **звіти** з нею, власним місцезнаходженням та часом і датою |

* 1. *. Середовище функціонування продукту*

Мобільна програма буде створена для використання на телефонах з ОС Android.

Вимоги до апаратного забезпечення:

* ОС актуальних версій (починаючи з 4.1 (API 16))
* розміру екрану від 320dp
* RAM від 512мб
* двух’ядерний процесор з тактовою частотою від 1.2 ГГц
* модулі GPS та інтернет (WiFi+Cellular)
  1. *Обмеження, правила та стандарти*

Розробка програми проводилася у середовищі Visual Studio 2015, на мові програмування C#, версія 6.0; з використанням Xamarin Framework v15.3, Xamarin.Android v7.4.0.2. Апаратні обмеження наведені вище.

* 1. *Документація для користувачів*

При презентації програми буде продемонстроване покрокове використання програми, на електронну пошту буде надіслана презентація функціоналу з ілюстраціями, буде проведений усний інструктаж з кожним користувачем.

* 1. *Припущення і залежності*

Наступна версія програми буде випущена після тривалого користування значною кількістю людей на основі їх вподобань, побажань та порад.

1. **Функціональність системи**

*3.1 Функціональний блок «Внесення особистої інформації»:*

3.1.1 Опис і пріоритет

Функція системи «Внесення особистої інформації» відповідає за наявність особистої інформації бійця у програмі. Середній пріорітет.

3.1.2 Причинно-наслідкові зв’язки, алгоритми

Вхідними даними до функціонального блоку внесення особистої інформації є тип ПІБ бійця, військову частина та номер значка. Відбувається внесення вказаних даних до пам’яті мобільного застосунку та пов’язаних з ним програм.

3.1.3 Функціональні вимоги

Текстові поля для прізвища, імені та по батькові бійця, числово-текстові для військової частини та номеру значка; збереження чи оновлення даних при натисканні відповідної кнопки.

*3.2 Функціональний блок «Відправлення* ***звіту****»:*

3.2.1 Опис і пріоритет

Функція системи «Відправлення **звіту**» відповідає за збір інформації до **звіту** та відправлення його на сервер. Високий пріорітет.

3.2.2 Причинно-наслідкові зв’язки, алгоритми

Вхідними даними до функціонального блоку відправлення **звіту** є оброблена особиста інформація. Вона включається до **звіту**, разом з часом, датою, що отримуються з операційної системи телефону, та місцезнаходження апарату, що отримується з геолокації телефону.

3.2.3 Функціональні вимоги

Об’єднання обробленої особистої інформації бійця, часу, дати, місцезнаходження до єдиного звіту та відправка його у форматі JSON на сервер.

**4 Вимоги до зовнішніх інтерфейсів**

*4.1 Інтерфейси користувачів*

- Фон програми повинен бути темно-зеленого кольору.

- Текстові поля повинні бути на всю ширину екрану.

- Текст введення повинен бути розміру 18sp та білого кольору.

- Кнопка збереження має бути на всю ширину екрану, з текстом розміру 18sp і білого кольору.

- Підтримуванні конфігурації екранів: від 320dр, співвідношення сторін – 16:9 і близькі до нього.

- Усі текстові поля мають містити підказки на фоні на рахунок їх значення.

- При натисканні на кнопку збереження введена інформація повинна зберігатися, за наявності минуло введеної – переписувати її.

- Зовнішній вигляд програми повинен відповідати стандартам інтерфейсу Android-застосунків – Material design.

*4.2 Апаратні інтерфейси*

|  |  |  |
| --- | --- | --- |
|  | **Мінімальні** | **Рекомендовані** |
| **Операційна система** | Android OS актуальних версій (починаючи з 4.1 (API 16)) | Android OS 4.4 і вище |
| [**Процесор**](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) | Двух’ядерний процесор з тактовою частотою від 1.2 ГГц | Двух’ядерний процесор з тактовою частотою від 1.5 ГГц |
| [**Оперативна пам'ять**](https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0_%D0%BF%D0%B0%D0%BC%27%D1%8F%D1%82%D1%8C) | 512 [Мб](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B3%D0%B0%D0%B1%D0%B0%D0%B9%D1%82) [RAM](https://uk.wikipedia.org/wiki/RAM) | 2 Гб RAM |
| **Розмір екрану** | 320dp | 640dp |
| **Інші пристрої** | Модулі GPS та інтернет (WiFi+Cellular) | Модулі GPS та інтернет (WiFi+Cellular) |

*4.3 Програмні інтерфейси*

Операційна система Android.

*4.4 Інтерфейси передачі інформації*

Зв’язок здійснюється через канал HTTP, геолокація шляхом GPS та FusedLocationProvider.

**5 Інші нефункціональні вимоги**

*5.1 Вимоги до продуктивності*

Програма призначена для використання одним користувачем, повинна стабільно і швидко працювати на апаратному забезпеченні, яке відповідає мінімальним вимогам.

*5.2 Проектні обмеження*

Див. п.4.2 «Апаратні інтерфейси».

*5.3 Атрибути системи*

a. Надійність  
• Програма має зберігати у часі в установлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання та транспортування.

b. Безпека  
• Програма не має ніяким чином впливати на роботу інших програм чи операційної системи загалом, не має несанкціоновано зберігати, використовувати чи відправляти будь-які дані користувача.

с. Супроводжуваність  
• Програмний код має бути побудований згідно з принципами та парадигмами об’єктно-орієнтованого програмування, SOLID та з використанням шаблонів проектування.

d. Перенесення  
• Код програми повинен бути побудований та скомпільований таким чином, щоб програма працювала на усьому сімействі апаратів Android OS відповідних версій.

e. Доступність

• Програма має передбачати збереження даних при закритті програми, недоступності зв’язку чи аварійному завершенні.

* 1. **Спільні дані програм**

Оскільки система «Кнопка життя» складається з багатьох програм, які постійно взаємодіють між собою та оперують одними даними, то було б неправильно розробляти їх незалежно одна від одної, тому перш за все важливо спроектувати систему і взаємодії між програмами.

Усі програми використовуватимуть інформацію про бійця – ПІБ, військову частину та номер значка. Для управління цими даними була створена модель інформації бійця (див. додаток 1).

Вона зберігає дані, введені бійцем, та вигружає їх з пам’яті для створення звіту. У пам’яті телефону ж вони також усі містяться у одному місці для доступу усіх мобільних застосунків – завдяки спеціально створеним спільним SharedPreferences: у маніфесті проекту зазначається загальний неймспейс (android:sharedUserId="com.ato.info"), і отримання даних відбувається за допомогою звертання до нього з вказанням унікальної збірки першого застосунку – сигнальної кнопки бійця – що і зберігає ці дані (див. додаток 2).

* 1. **Спільні методи мобільних програм**

Задля кращої читаємості коду, легшого його виправлення, рефакторингу та внесення змін, було прийнято рішення зробити частину методів спільними для усіх трьох мобільних програм. Це стосується неспеціалізованого функціоналу, який потрібний кожній з них: зчитування місцезнаходження пристрою, встановленого часу і дати, роботи з GPS та інтернет-підключенням. Більшість з цих задач є тривіальними, тому зупинятися на них немає сенсу, важливим є тільки вибір методу роботи з геолокацією.

**1.3.1 Об’єднаний постачальник місцезнаходження**

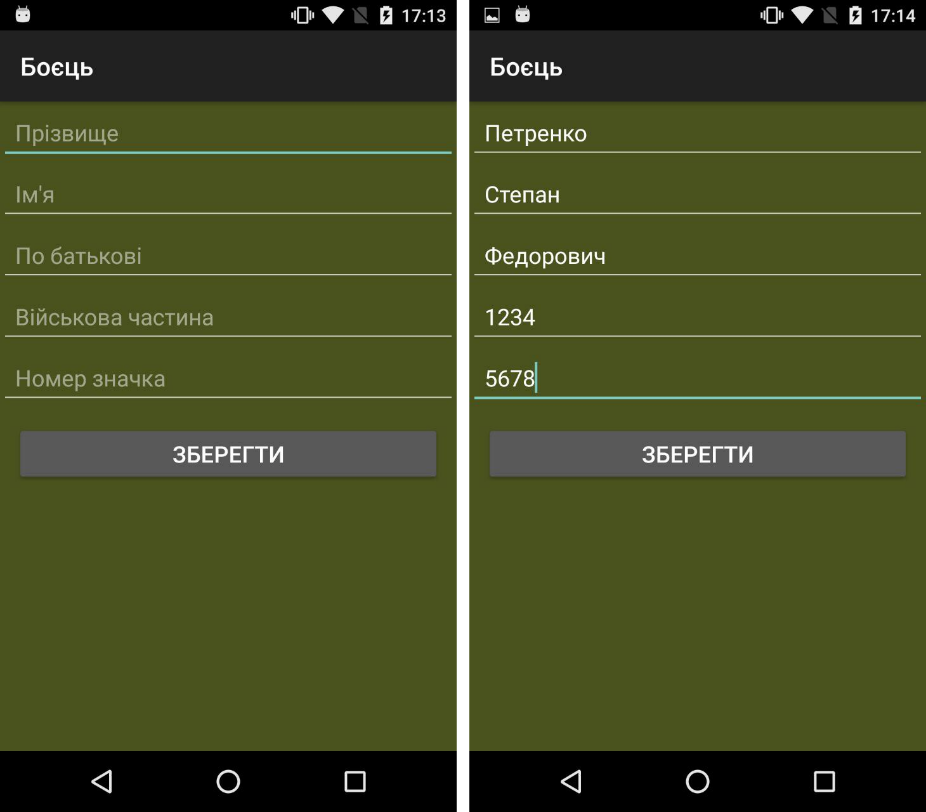
Об’єднаний постачальник місцезнаходження (Fused Location Provider) є додатковою альтернативою стандартної служби Android Location, яка автоматично обробляє зміни стану постачальника всередині програми. Він є частиною Google Play Services, яка включає в себе нові інтерфейси API Google Location Services. Google Play Services повинен бути встановлений і правильно налаштований в додатку для роботи об’єднаного постачальника місцезнаходження.

Додаток за допомогою об’єднаного постачальника місцезнаходження переключається динамічно між постачальниками, щоб вибрати кращий постачальник доступних в будь-який момент часу. Наприклад, коли користувач ходить по вулиці, отримує найкраще місцезнаходження з GPS. Якщо користувач заходить в приміщення, де GPS працює погано, постачальник автоматично перемикається на Wi-Fi, який працює краще в приміщенні.

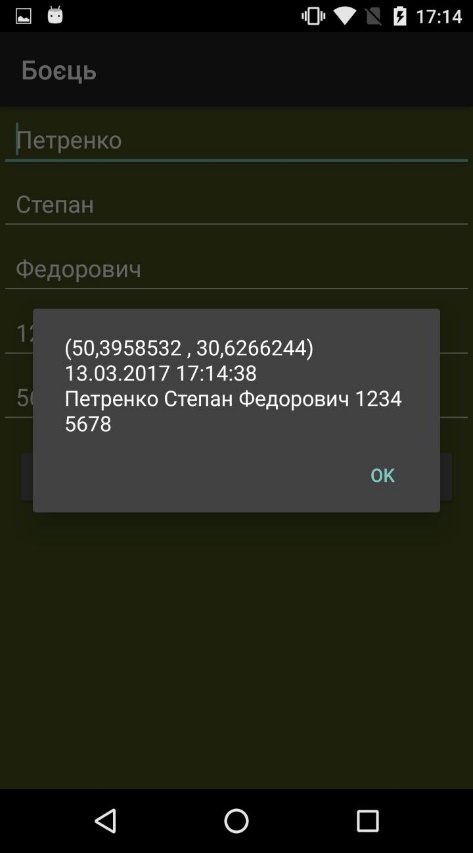
Реалізація відбувається за допомогою інтерфейсів GoogleApiClient IConnectionCallbacks (void OnConnected (Bundle connectionHint), OnConnectionSuspended (int cause)) та GoogleApiClient IOnConnectionFailedListener (void OnConnectionFailed (ConnectionResult result)), отримання останнього відомого місцезнаходження – за допомогою методу LocationServices.FusedLocationApi.GetLastLocation(\_apiClient).

**1.4 Робота мобільної програми «Сигнальна кнопка бійця»**

1. Боєць вводить необхідну особисту інформацію



2. При повторних запусках особиста інформація разом з місцеположенням бійця і датою та часом відправляються на сервер



**2 Проектування програм «Сигнальна кнопка товариша за зброєю» та «Сигнальна кнопка фельдшера»**

* 1. **Специфікація програми «Сигнальна кнопка товариша»**

**1 Вступ**

*1.1 Призначення*

Назва програми: Сигнальна кнопка товариша за зброєю

Скороченна назва: Кнопка товариша

Версія програмного продукту: 1.0.0

*1.2 Угоди, прийняті в документах*

Зауваження в тексті виділяються курсивом.

Терміни та ключові слова виділяються жирним шрифтом.

*1.3 Передбачуванна аудиторія і рекомендації з читання*

Специфікація призначенна для розробників ПЗ та замовників.

*1.4 Границі проекту*

**Мета роботи** – зменшити кількість смертних випадків, що трапляються через неготовність до надання медичної допомоги у польовому шпиталі.

**Основний результат** – завдяки використанню мобільної програми буде суттєво зменшено кількість смертних випадків, що трапляються через неготовність до надання медичної допомоги у польовому шпиталі.

Результати роботи можуть бути використані для впровадження у збройних силах чи інших структурах, які можуть і яким потрібно вчасно реагувати на певні події у людей.

*1.5* *Посилання*

1. Посилання на стандарти інтерфейсу:

<https://developer.android.com/design/material/index.html>

1. Посилання на стандарти якості:

<https://developer.android.com/develop/quality-guidelines/core-app-quality.html>

**2 Загальний опис**

*2.1 Загальний погляд на продукт*

При запуску цієї програми вона фотографує поранення, зчитує координати з GPS, час і передає цю (включаючи особисту інформацію про бійця) інформацію на сервер польового шпиталю.

Програма встановлюється на мобільному пристрої бійця.

* 1. *Особливості (функціональність) продукту*

Програмамний продукт володіє наступними функціями:

* Фотографування поранення бійця
* Збір інформації про місцезнаходження бійця та час
* Відправлення **звіту** *(інформації про бійця)* на сервер

Вербальний опис діаграми прецедентів:

Специфікація прецеденту «*Зробити знімок*»:

1. **Короткий опис:** Товариш фотографує поранення бійця.
2. **Суб’єкти:** Товариш за зброєю.
3. **Передумови:** відсутні.
4. **Основний потік:** Фотографія поранення зберігається на телефоні, та доступна для відправлення як частина **звіту**.
5. **Альтернативні потоки:** відсутні.
6. **Постумови:** Фотографія поранення збережена на телефоні і доступна для відправки як частина **звіту**.

*Специфікації прецедентів «Відправлення звіту», «Отримання особистої інформації», «Отримання часу і дати», «Отримання місцезнаходження» аналогічні до відповідних специфікацій прецедентів програми «Кнопка бійця»*

* 1. *Класи і характеристики користувачів*
  2. *Середовище функціонування продукту*
  3. *Обмеження, правила та стандарти*
  4. *Документація для користувачів*

*2.7 Припущення і залежності*

*Аналогічні до відповідних пунктів специфікації програми «Кнопка бійця».*

1. **Функціональність системи**

*3.1 Функціональний блок «Зробити знімок»:*

3.1.1 Опис і пріоритет

Функція системи «Зробити знімок» відповідає за фотографування знімку поранення бійця товаришем за зброєю. Середній пріорітет.

3.1.2 Причинно-наслідкові зв’язки, алгоритми

Вхідними даними до функціонального блоку фотографування поранення є фотографія поранення. Відбувається збереження знімку до пам’яті мобільного пристрою.

3.1.3 Функціональні вимоги

Інтерфейс камери має бути рідним інтерфейсом Android-апарату.

*3.2 Функціональний блок «Відправлення* ***звіту****»:*

3.2.1 Опис і пріоритет

Функція системи «Відправлення **звіту**» відповідає за збір інформації до **звіту** та відправлення його на сервер. Високий пріорітет.

3.2.2 Причинно-наслідкові зв’язки, алгоритми

Вхідними даними до функціонального блоку відправлення **звіту** є зроблене фото поранення бійця. Воно включається до **звіту**, разом з часом, датою, що отримуються з операційної системи телефону, та місцезнаходження апарату, що отримується з геолокації телефону.

3.2.3 Функціональні вимоги

Об’єднання знімку поранення, особистої інформації бійця, часу, дати, місцезнаходження до єдиного звіту та відправка його POST-запитом на сервер.

**4 Вимоги до зовнішніх інтерфейсів**

*4.1 Інтерфейси користувачів*

*4.2 Апаратні інтерфейси*

*4.3 Програмні інтерфейси*

*4.4 Інтерфейси передачі інформації*

*Аналогічні до відповідних пунктів специфікації програми «Кнопка бійця».*

**5 Інші нефункціональні вимоги**

*5.1 Вимоги до продуктивності*

*5.2 Проектні обмеження*

*5.3 Атрибути системи*

*Аналогічні до відповідних пунктів специфікації програми «Кнопка бійця».*

**2.2 Специфікація програми «Сигнальна кнопка фельдшера»**

**1 Вступ**

*1.1 Призначення*

Назва програми: Сигнальна кнопка фельдшера

Скороченна назва: Кнопка фельдшера

Версія програмного продукту: 1.0.0

*1.2 Угоди, прийняті в документах*

Зауваження в тексті виділяються курсивом.

Терміни та ключові слова виділяються жирним шрифтом.

*1.3 Передбачуванна аудиторія і рекомендації з читання*

Специфікація призначенна для розробників ПЗ та замовників.

*1.4 Границі проекту*

**Мета роботи** – зменшити кількість смертних випадків, що трапляються через неготовність до надання медичної допомоги у стаціонарному шпиталі.

**Основний результат** – завдяки використанню мобільної програми буде суттєво зменшено кількість смертних випадків, що трапляються через неготовність до надання медичної допомоги у стаціонарному шпиталі.

Результати роботи можуть бути використані для впровадження у збройних силах чи інших структурах, які можуть і яким потрібно вчасно реагувати на певні події у людей.

*1.5* *Посилання*

1. Посилання на стандарти інтерфейсу:

<https://developer.android.com/design/material/index.html>

1. Посилання на стандарти якості:

<https://developer.android.com/develop/quality-guidelines/core-app-quality.html>

**2 Загальний опис**

*2.1 Загальний погляд на продукт*

При запуску програми вона фотографує картку бійця, зчитує координати з GPS, час і передає цю (включаючи особисту інформацію про бійця) інформацію на сервер стаціонарного шпиталю.

Програма встановлюється на мобільному пристрої бійця.

*2.2* *Особливості (функціональність) продукту*

Програмамний продукт володіє наступними функціями:

* Фотографування картки бійця
* Збір інформації про місцезнаходження бійця та час
* Відправлення **звіту** *(інформації про бійця)* на сервер

Вербальний опис діаграми прецедентів:

Специфікація прецеденту «*Зробити знімок*»:

1. **Короткий опис:** Фельдшер фотографує картку бійця.
2. **Суб’єкти:** Фельдшер.
3. **Передумови:** відсутні.
4. **Основний потік:** Фотографія картки зберігається на телефоні, та доступна для відправлення як частина **звіту**.
5. **Альтернативні потоки:** відсутні.
6. **Постумови:** Фотографія картки збережена на телефоні і доступна для відправки як частина **звіту**.

*Специфікації прецедентів «Відправлення звіту», «Отримання особистої інформації», «Отримання часу і дати», «Отримання місцезнаходження» аналогічні до відповідних специфікацій прецедентів програми «Кнопка бійця»*

* 1. *Класи і характеристики користувачів*
  2. *Середовище функціонування продукту*
  3. *Обмеження, правила та стандарти*
  4. *Документація для користувачів*
  5. *Припущення і залежності*

*Аналогічні до відповідних пунктів специфікації програми «Кнопка бійця».*

1. **Функціональність системи**

*3.1 Функціональний блок «Зробити знімок»:*

3.1.1 Опис і пріоритет

Функція системи «Зробити знімок» відповідає за фотографування знімку картки бійця фельдшером. Середній пріорітет.

3.1.2 Причинно-наслідкові зв’язки, алгоритми

Вхідними даними до функціонального блоку фотографування картки є фотографія картки. Відбувається збереження знімку до пам’яті мобільного пристрою.

3.1.3 Функціональні вимоги

Інтерфейс камери має бути рідним інтерфейсом Android-апарату.

*3.2 Функціональний блок «Відправлення* ***звіту****»:*

3.2.1 Опис і пріоритет

Функція системи «Відправлення **звіту**» відповідає за збір інформації до **звіту** та відправлення його на сервер. Високий пріорітет.

3.2.2 Причинно-наслідкові зв’язки, алгоритми

Вхідними даними до функціонального блоку відправлення **звіту** є зроблене фото картки бійця. Воно включається до **звіту**, разом з часом, датою, що отримуються з операційної системи телефону, та місцезнаходження апарату, що отримується з геолокації телефону.

3.2.3 Функціональні вимоги

Об’єднання знімку картки бійця, особистої інформації бійця, часу, дати, місцезнаходження до єдиного звіту та відправка його POST-запитом на сервер.

**4 Вимоги до зовнішніх інтерфейсів**

*4.1 Інтерфейси користувачів*

*4.2 Апаратні інтерфейси*

*4.3 Програмні інтерфейси*

*4.4 Інтерфейси передачі інформації*

*Аналогічні до відповідних пунктів специфікації програми «Кнопка бійця».*

**5 Інші нефункціональні вимоги**

*5.1 Вимоги до продуктивності*

*5.2 Проектні обмеження*

*5.3 Атрибути системи*

*Аналогічні до відповідних пунктів специфікації програми «Кнопка бійця».*

*Діаграми «Кнопки фельдшера» є по суті діаграмами «Кнопки товариша», тільки дієвою особою у них є фельдшер замість товариша за зброєю бійця, та фото слугує фото медичної картки бійця замість фото поранення. Інші відмінності є суто технічними (сервер відправки* ***звіту****) та до проектування не відносяться.*

* 1. **Робота з камерою**

Для реалізації функціоналу відправлення фото поранення (у другому мобільному застосунку «сигнальна кнопка товариша за зброєю») та медичної картки бійця (у третьому мобільному застосунку «сигнальна кнопка фельдшера») була виконана робота з камерою мобільного пристрою на базі ОС Android.

**2.3.1 Абстракція від апаратних засобів**

Через велику різноманітність апаратних представлень важливим є правильне перемикання контекстів між власним застосунком і застосунком камери, мінімальне з’єднання з контекстом камери, та слабкий зв’язок з і залежність від конкретних апаратних властивостей камери чи пристрою.

Для роботи з апаратним інтерфейсом камери було створено статичний клас – модель фото (див. додаток 1).

Нас не цікавлять характеристики камери чи деталі її роботи, ми абстрагуємося до рівня того, що вона фотографує, і ведемо роботу з результатом роботи, бо тільки він нас і цікавить.

**2.3.2** **Перевірка фактичної наявності інтерфейсу камери у апарату**

Відбувається пошук будь-якого застосунку, що встановлений на додатку, та задовольняє інтерфейс камери – надає фотознімок (див. додаток 3).

**2.3.3 Створення директорії для зберігання фото**

Такий підхід є кращим за інший існуючий варіант – роботу з фото у життєвому циклі застосунку, оскільки він зберігає зроблені фото у галереї для подальшого доступу до них та повторної відправки при відсутності інтернету (див. додаток 3).

**2.3.4 Відкриття інтерфейсу камери**

Використовуючи максимально можливу абстракцію, ми віддаємо лише директорію для збереження файлу та його назву, та очікуємо результат, який буде нас задовольняти за попередніми перевірками існування інтерфейсу камери. Далі і до наступного пункту усі роботи ведуться в камери і не залежать від нашого застосунку (див. додаток 3).

**2.3.5 Обробка результату камери**

При переключенні контексту з камери до застосунку, виконується передпоказ фото та запускається механізм відправки усіх необхідних даних на сервер (див. додаток 3).

**2.3.6 Завантаження знімку на сервер**

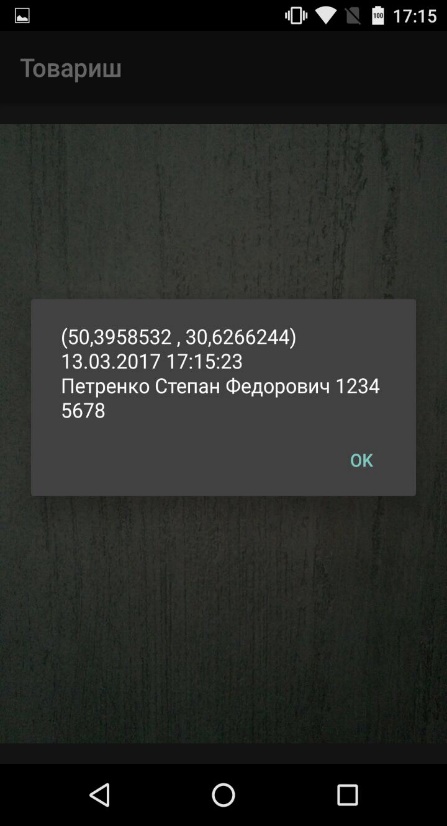
Відбувається як частина відправки всього звіту, долучається до інформації про військового, місцезнаходження і часу та дати. Використовується переведення медіафайлу через потік пам’яті у байтовий масив (див. додаток 3).

**2.4** **Робота мобільних програм «Сигнальна кнопка товариша за зброєю» та «Сигнальна кнопка фельдшера»**

1. Запускається інтерфейс камери. Товариш бійця фотографує поранення / Фельдшер фотографує медичну картку бійця

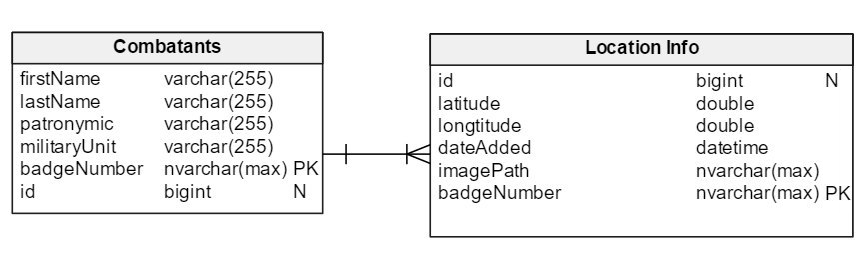


2. Фотографія поранення разом з особистою інформацією, місцеположенням бійця і датою та часом відправляються на сервер польового шпиталю



**3 Проектування бази даних**

**3.1 Схема бази даних**



База даних складається з двох таблиць: таблиці бійців Combatants та таблиці інформації про них LocationInfo. Серверна частина прийняття та обробки інформації, що надсилається мобільними застосунками - див. додаток 4.

**3.2 Таблиця Combatants**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назва поля** | **Опис** | **Тип** | **Первинний ключ** | **Зовнішній ключ** |
| id | Ідентифікатор запису | Bigint | Так | Ні |
| firstName | Ім’я бійця | Varchar(255) | Ні | Ні |
| lastName | Прізвище бійця | Varchar(255) | Ні | Ні |
| patronymic | По-батькові бійця | Varchar(255) | Ні | Ні |
| militaryUnit | Військова частина | Varchar(255) | Ні | Ні |
| badgeNumber | Номер значка | Nvarchar(max) | Ні | LocationInfo.badgeNumber |

**3.3 Таблиця LocationInfo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назва поля** | **Опис** | **Тип** | **Первинний ключ** | **Зовнішній ключ** |
| id | Ідентифікатор запису | Bigint | Так | Ні |
| latitude | Широта | Double | Ні | Ні |
| longtitude | Довгота | Double | Ні | Ні |
| dateAdded | Дата додання | Datetime | Ні | Ні |
| imagePath | Путь до фото | Nvarchar(max) | Ні | Ні |
| badgeNumber | Номер значка | Nvarchar(max) | Ні | Сombatants.badgeNumber |

**Висновок**

Під час виконання даної курсової роботи було розроблено три мобільних програми для підвищення ефективності надання первинної медичної допомоги при пораненні військовослужбовця та спроектована база даних для зберігання інформації про бійців.

Курсова робота містить змістовну інформацію щодо проектування комп’ютерних кросплатформених систем і баз даних, ілюстрації роботи застосунків та детальний розгляд роботи з геолокацією та камерою на пристроях на базі ОС Android.

В першому розділі курсової роботи було розглянуто проектування мобільної програми «Сигнальна кнопка бійця»; особливості роботи з геолокацією у ОС Android.

У другому розділі було розглянуто проектування мобільних програм «Сигнальна кнопка товариша за зброєю» і «Сигнальна кнопка фельдшера»; робота з інтерфейсом камери апаратних засобів на базі ОС Android; передача медіафайлів засобами .Net.

У третьому розділі розглянута та описана спроектована база даних для зберігання інформації про бійців; обробка і збереження інформації, що передається мобільними програмами на сервер.

Результат роботи допоможе суттєво зменшити кількість смертних випадків, що трапляються до надання медичної допомоги у стаціонарному шпиталі, і може бути вбудований в велику кількість систем з необхідним наглядом і швидким реагуванням.

**Бібліографічний список**

1. Stephane Faroult «The Art of SQL»
2. <https://developer.android.com>
3. <https://developer.android.com/design/material/index.html>
4. <https://developer.android.com/develop/quality-guidelines/core-app-quality.html>
5. <https://developer.xamarin.com/guides/android/>
6. <http://stacktips.com>
7. Bertrand Meyer «Object-Oriented Software Construction»
8. Steve McConnell «Code Complete (2nd Edition)»
9. Martin Fowler «Refactoring: Improving the Design of Existing Code»

**Додаток 1: Моделі програми**

1. **Модель інформації бійця**

public class CombatantInfo

{

private string \_surname;

private string \_name;

private string \_patronymic;

private string \_militaryUnit;

private string \_badgeNumber;

public CombatantInfo(ISharedPreferences pref)

{

\_surname = pref.GetString("surname", string.Empty);

\_name = pref.GetString("name", string.Empty);

\_patronymic = pref.GetString("patronymic", string.Empty);

\_militaryUnit = pref.GetString("militaryUnit", string.Empty);

\_badgeNumber = pref.GetString("badgeNumber", string.Empty);

}

public override string ToString() => $"{\_surname} {\_name} {\_patronymic} {\_militaryUnit} {\_badgeNumber}";

}

1. **Модель фото**

public static class Photo

{

public static File File;

public static File Directory;

public static Bitmap Bitmap;

}

**Додаток 2: Збереження за збіркою**

var context = CreatePackageContext("com.ato.combatant", PackageContextFlags.IgnoreSecurity);

var pref = context.GetSharedPreferences("info", FileCreationMode.Private);

CombatantInfo combatantInfo = new CombatantInfo(pref);

**Додаток 3: Робота з камерою**

1. **Перевірка існування інтерфейсу камери**

private bool IsThereAnAppToTakePictures()

{

Intent intent = new Intent(MediaStore.ActionImageCapture);

IList<ResolveInfo> availableActivities = PackageManager.QueryIntentActivities(intent, PackageInfoFlags.MatchDefaultOnly);

return availableActivities != null && availableActivities.Count > 0;

}

1. **Створення директорії для зберігання фото**

private void CreateDirectoryForPictures()

{

Photo.Directory = new Java.IO.File(Environment.GetExternalStoragePublicDirectory(Environment.DirectoryPictures), "Товариш");

if (!Photo.Directory.Exists())

{

Photo.Directory.Mkdirs();

}

}

1. **Відкриття інтерфейсу камери**

private void OpenCamera()

{

Photo.File = new Java.IO.File(Photo.Directory, string.Format("Товариш\_{0}.jpg", Guid.NewGuid()));

Intent intent = new Intent(MediaStore.ActionImageCapture);

intent.PutExtra(MediaStore.ExtraOutput, Uri.FromFile(Photo.File));

StartActivityForResult(intent, 0);

}

1. **Обробка результату камери**

protected override void OnActivityResult(int requestCode, Result resultCode, Intent data)

{

base.OnActivityResult(requestCode, resultCode, data);

Intent mediaScanIntent = new Intent(Intent.ActionMediaScannerScanFile);

Uri contentUri = Uri.FromFile(Photo.File);

mediaScanIntent.SetData(contentUri);

SendBroadcast(mediaScanIntent);

\_bitmap = BitmapFactory.DecodeFile(Photo.File.Path);

\_picturePreview.SetImageBitmap(\_bitmap);

Task.Run(async () => await SendReportAsync(GetLastLocation()));

}

1. **Завантаження знімку на сервер**

public async Task<string> UploadBitmapAsync(Bitmap bitmap)

{

byte[] bitmapData;

var stream = new MemoryStream();

bitmap.Compress(Bitmap.CompressFormat.Jpeg, 0, stream);

bitmapData = stream.ToArray();

var fileContent = new ByteArrayContent(bitmapData);

fileContent.Headers.ContentType = MediaTypeHeaderValue.Parse("application/octet-stream");

fileContent.Headers.ContentDisposition = new ContentDispositionHeaderValue("form-data")

{

Name = "file",

FileName = "my\_uploaded\_image.jpg"

};

string boundary = "---8d0f01e6b3b5dafaaadaad";

MultipartFormDataContent multipartContent = new MultipartFormDataContent(boundary);

multipartContent.Add(fileContent);

HttpClient httpClient = new HttpClient();

HttpResponseMessage response = await httpClient.PostAsync(UPLOAD\_IMAGE, multipartContent);

if (response.IsSuccessStatusCode)

{

string content = await response.Content.ReadAsStringAsync();

return content;

}

return null;

}

**Додаток 4: Робота з БД**

public static class CommandantRepository

{

private static MySql.Data.MySqlClient.MySqlConnection con;

private static MySql.Data.MySqlClient.MySqlCommand command;

static CommandantRepository()

{

con = new MySql.Data.MySqlClient.MySqlConnection();

con.ConnectionString = "Server=MYSQL5015.Smarterasp.net;Database=db\_9f9885\_wau;Uid=9f9885\_combatant;Pwd=secretno;";

}

1. **Перевірка бійця в системі**

public static bool CheckIfCombatantIsInSystem(string BadgeNumber)

{

bool output=false;

string query = string.Format("select \* FROM Combatants where Combatants.BadgeNumber={0};",BadgeNumber);

try

{

command = new MySql.Data.MySqlClient.MySqlCommand(query, con);

con.Open();

command.CommandText = query;

MySql.Data.MySqlClient.MySqlDataReader reader;

reader = command.ExecuteReader();

if(reader.FieldCount()>0)

output=true;

}

finally

{

con.Close();

}

return output;

}

1. **Додавання бійця в систему**

public static void AddCombatantToSystem(string JsonString)

{

JToken token = JObject.Parse(JsonString);

string query = string.Format("Insert into Combatants(firstName,lastName,patronymic,militaryUnit,badgeNumber)VALUES({0},{1},{2},{3},{4})",(string)token.SelectToken("name"),

(string)token.SelectToken("surname"),(string)token.SelectToken("patronymic"),(string)token.SelectToken("militaryUnit"),(string)token.SelectToken("badgeNumber"));

try

{

command = new MySql.Data.MySqlClient.MySqlCommand(query, con);

con.Open();

command.CommandText = query;

MySql.Data.MySqlClient.MySqlDataReader reader;

reader = command.ExecuteReader();

}

finally

{

con.Close();

}

}

1. **Додавання даних про бійця в систему**

public static void AddCombatantLocation(string JsonString)

{

JToken token = JObject.Parse(JsonString);

string query = string.Format("Insert into LocationInfo (latitude,longtitude,dateAdded,imagePath,badgeNumber)VALUES({0},{1},{2},{3},{4})",(string)token.SelectToken("latitude"),

(string)token.SelectToken("longtitude"),(string)token.SelectToken("dateAdded"),(string)token.SelectToken("imagePath"),(string)token.SelectToken("badgeNumber"));

try

{

command = new MySql.Data.MySqlClient.MySqlCommand(query, con);

con.Open();

command.CommandText = query;

MySql.Data.MySqlClient.MySqlDataReader reader;

reader = command.ExecuteReader();

}

finally

{

con.Close();

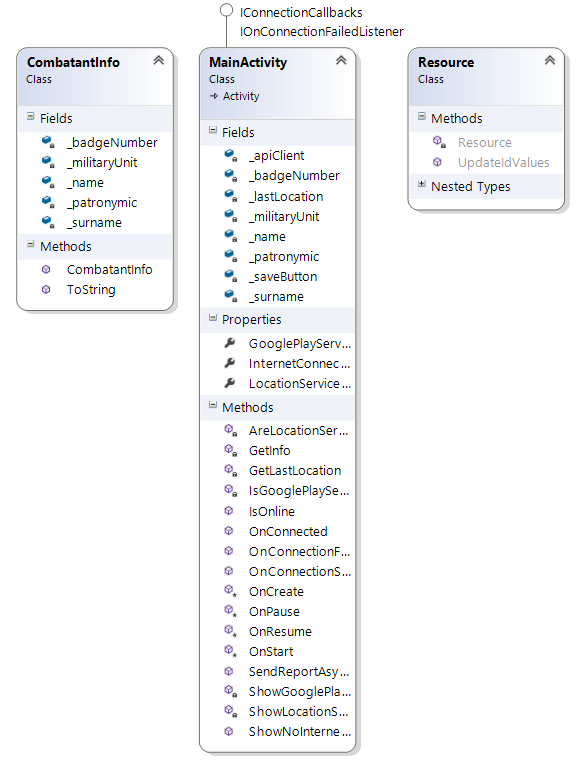
}

}

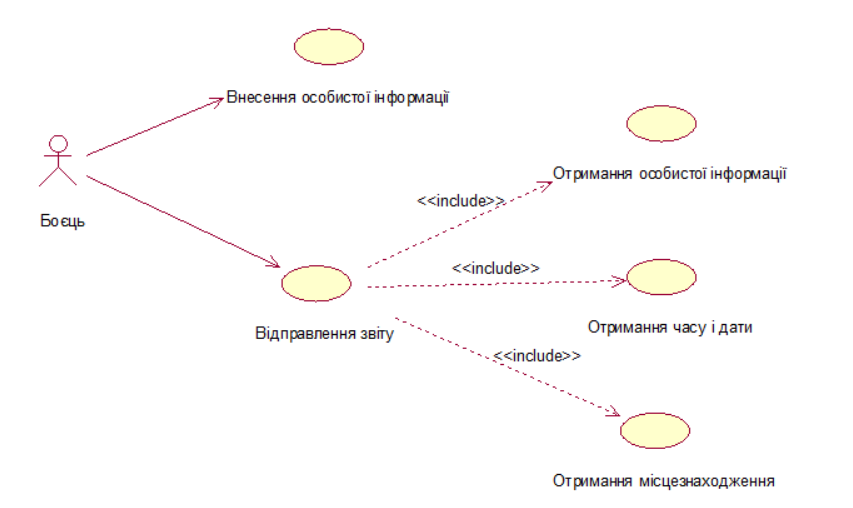
}

**Додаток 5: Моделі аналізу сигнальної кнопки бійця**

1. **Діаграма класів:**

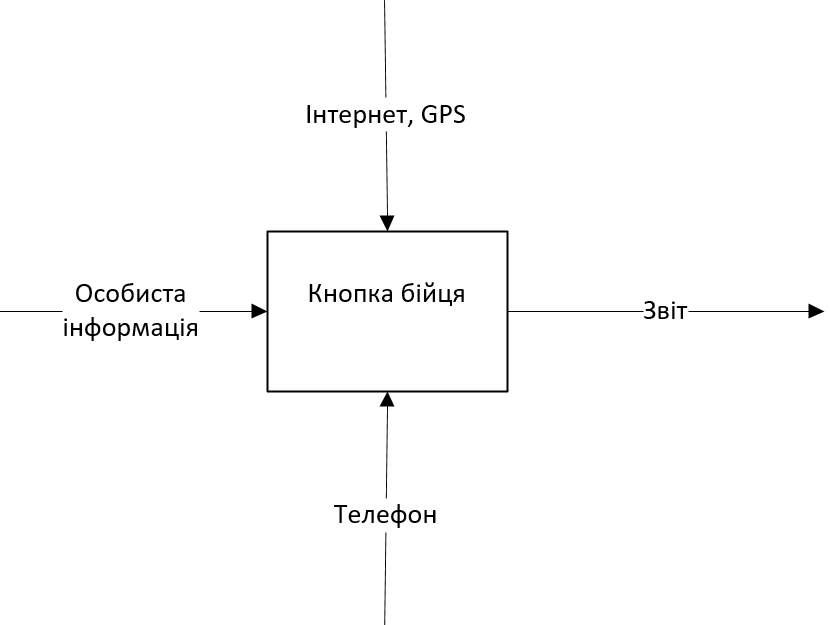


1. **Діаграма прецедентів:**

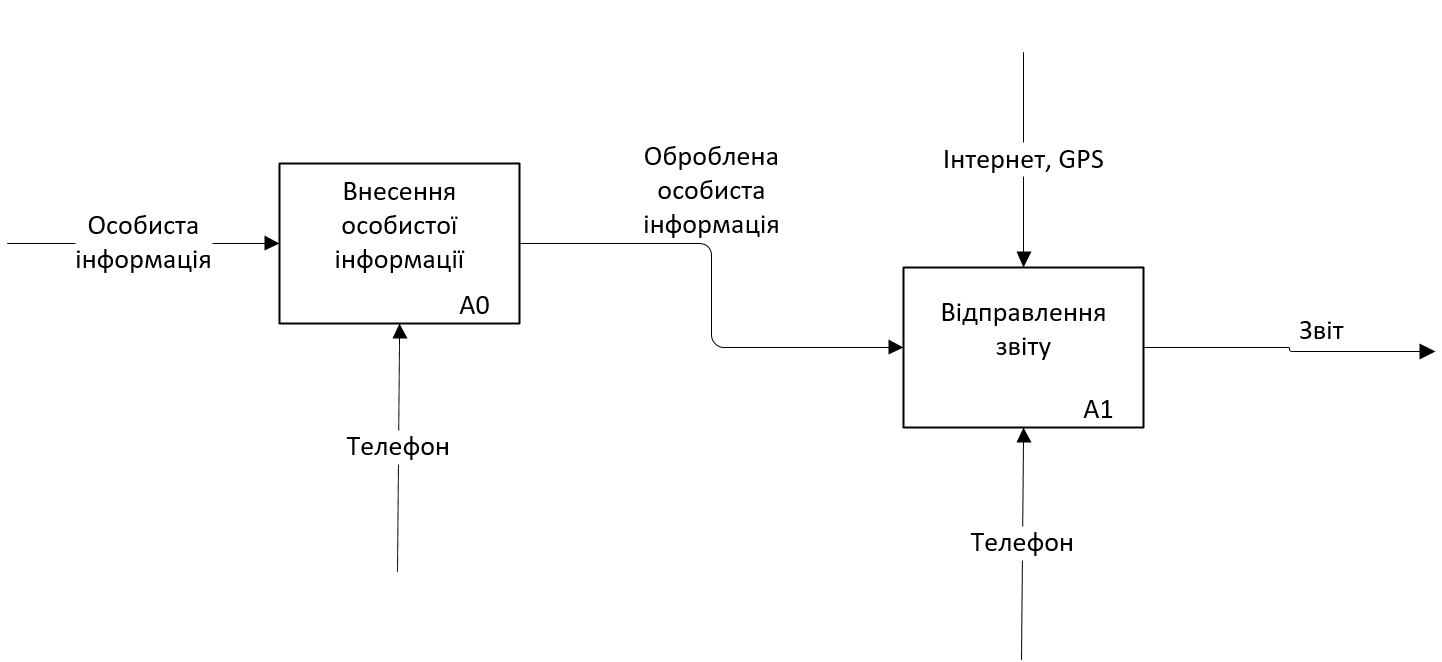


1. **Нотація IDEF0:**

*Контекстний рівень:*



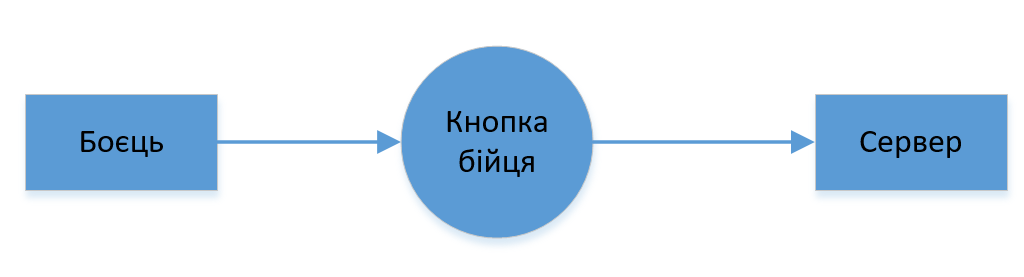
*Нульовий рівень:*



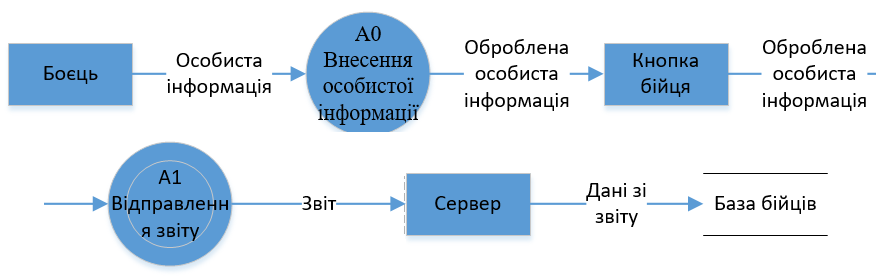
**Нотація IDEF3 та нижчі рівні IDEF0** не є необхідними для даного програмного забезпечення, оскільки вони використовується для декомпозиції нотації IDEF0.

1. **Діаграма потоків:**

**Контекстна діаграма:**

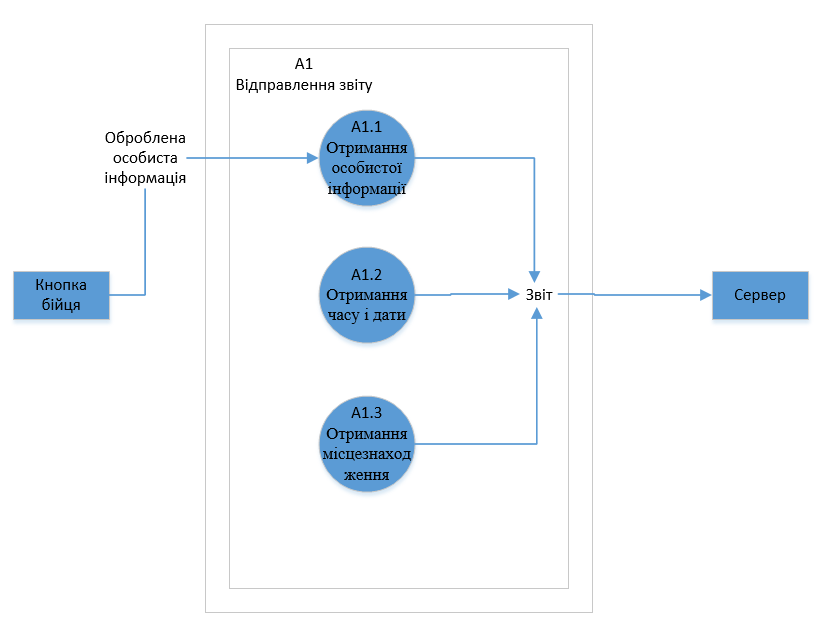


**Діаграма потоків нульового рівня:**

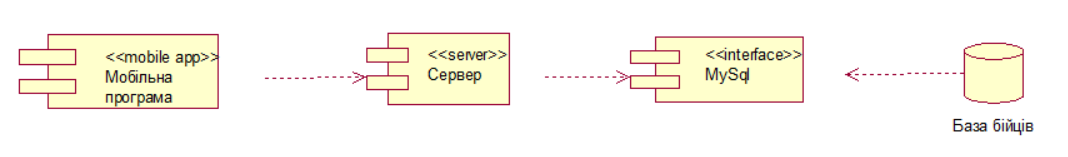
****

**Діаграми потоків першого рівня:**

**А1:**



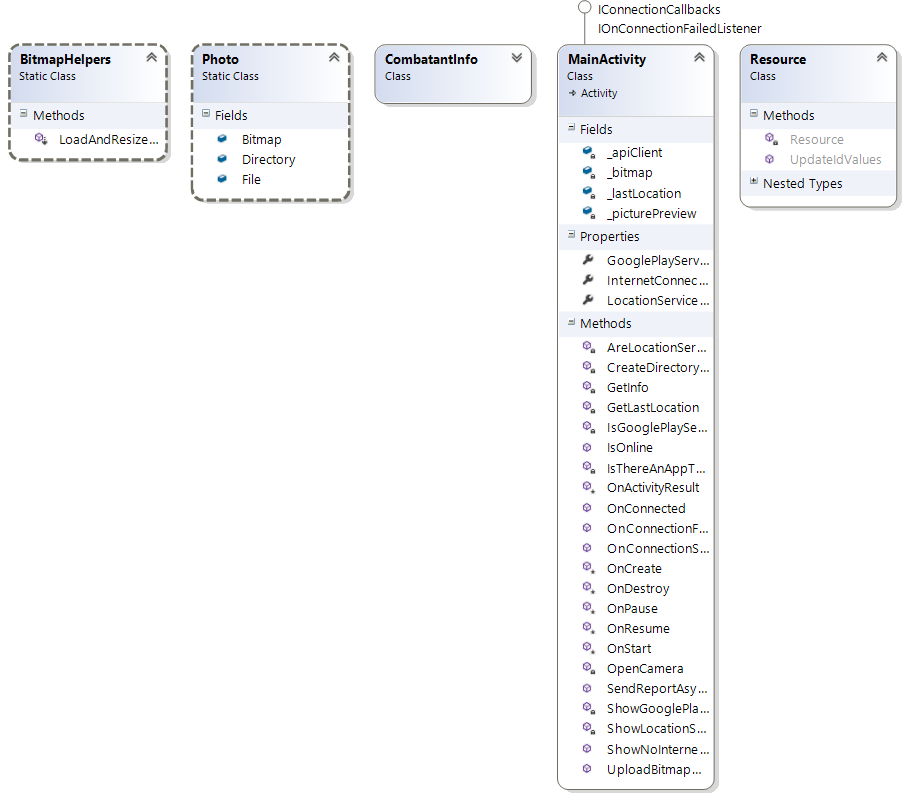
1. **Діаграма компонентів:**



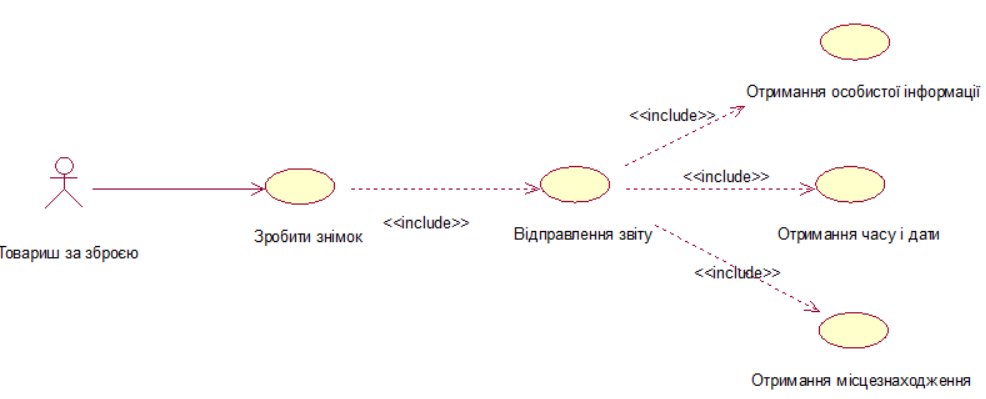
1. **Діаграма розгортання** описує відображає загальну конфігурацію і топологію системи, і не є необхідною для даного програмного забезпечення через відсутність взаємодії інших зовнішніх, відносно програми, апаратних компонентів з програмою як з єдиним цілим.

**Додаток 6: Моделі аналізу сигнальної кнопки товариша за зброєю**

1. **Діаграма класів:**

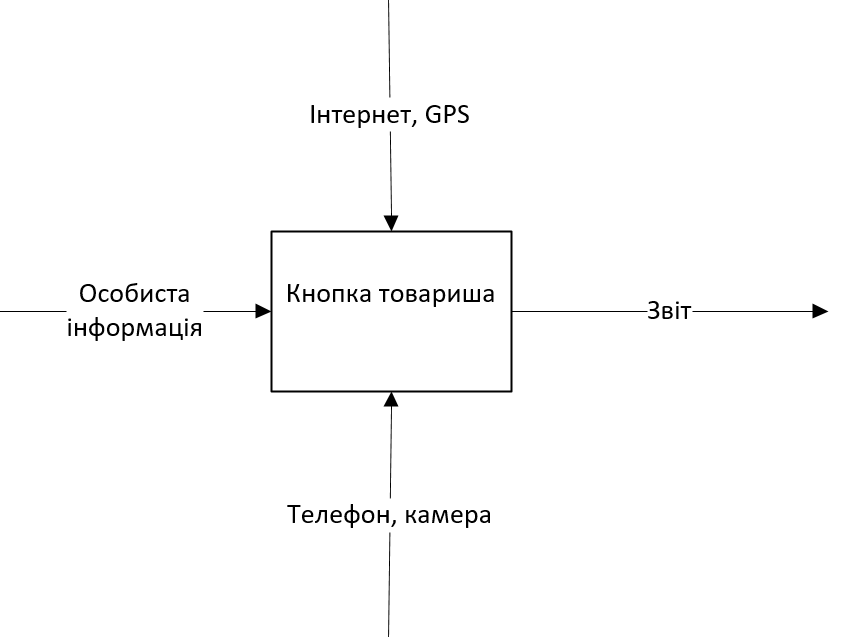


1. **Діаграма прецедентів:**

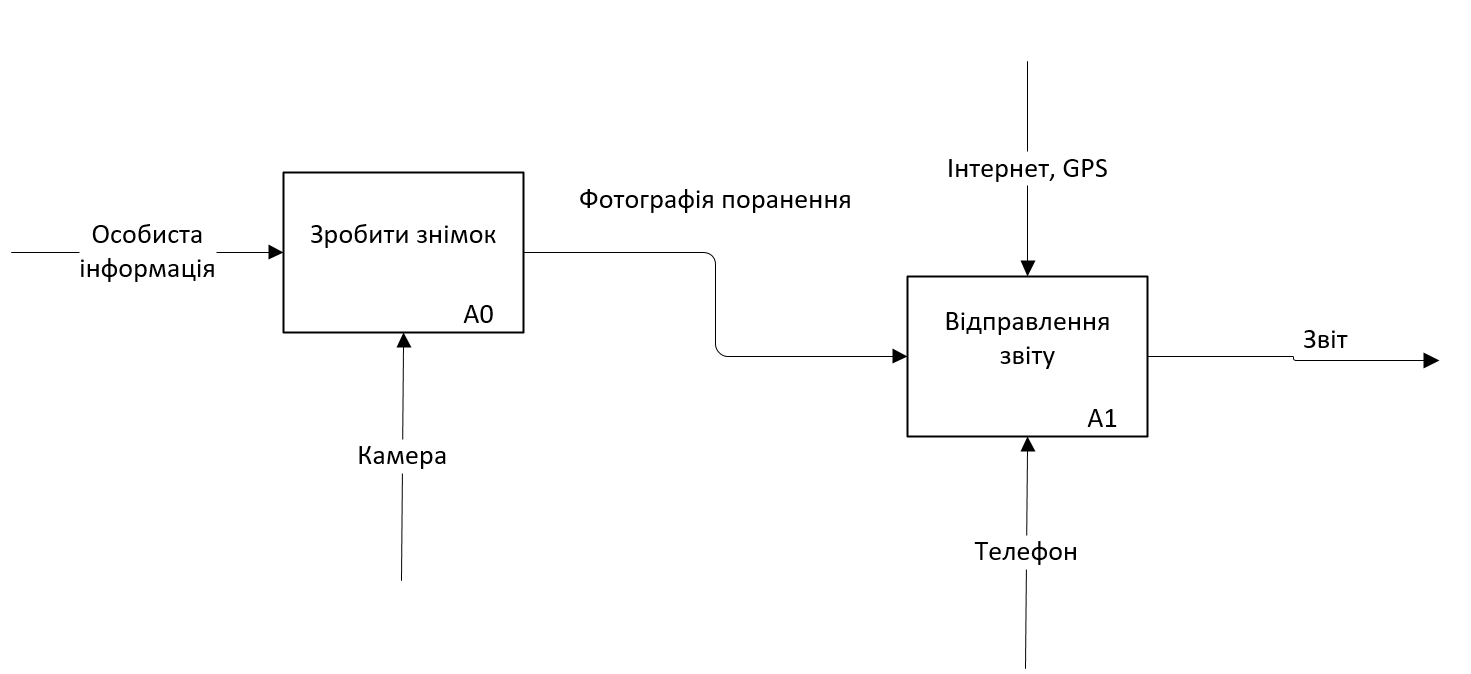


1. **Нотація IDEF0:**

*Контекстний рівень:*



*Нульовий рівень:*



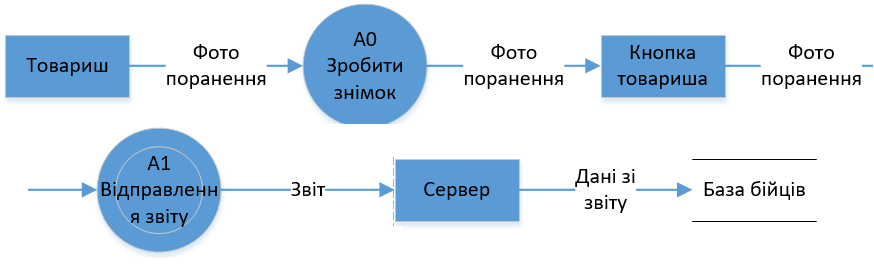
**Нотація IDEF3 та нижчі рівні IDEF0** не є необхідними для даного програмного забезпечення, оскільки вони використовується для декомпозиції нотації IDEF0.

1. **Діаграма потоків:**

**Контекстна діаграма:**

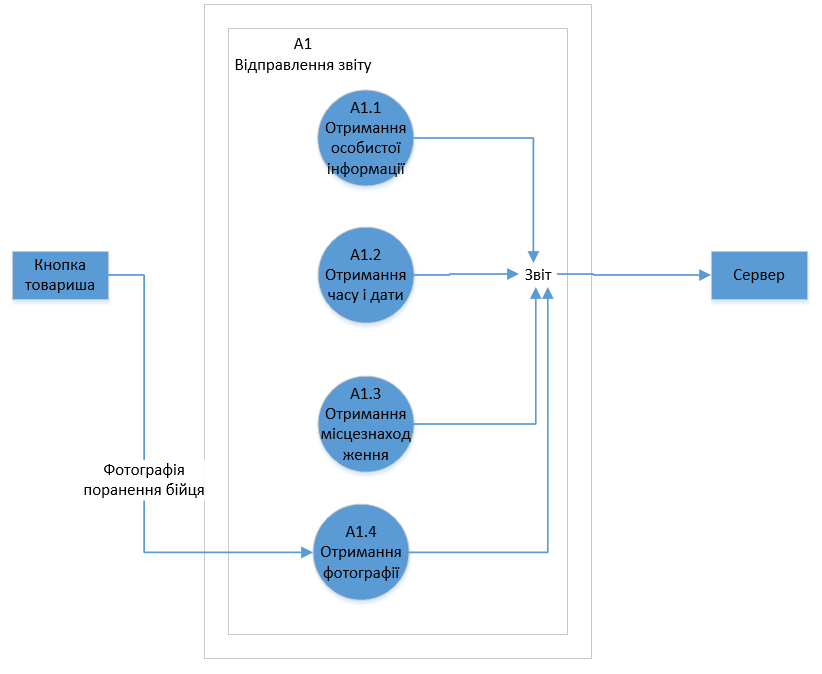


**Діаграма потоків нульового рівня:**

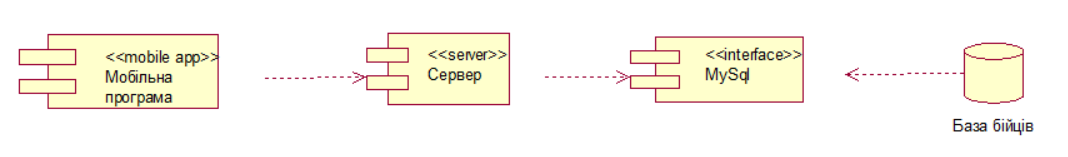
****

**Діаграми потоків першого рівня:**

**А1:**



1. **Діаграма компонентів:**



1. **Діаграма розгортання** описує відображає загальну конфігурацію і топологію системи, і не є необхідною для даного програмного забезпечення через відсутність взаємодії інших зовнішніх, відносно програми, апаратних компонентів з програмою як з єдиним цілим.