

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

**Пояснювальна записка**

до бакалаврської роботи  
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ  
СКЛАДАННЯ ПРОТОКОЛУ ПРО ДОРОЖНЬО-ТРАНСПОРТНУ  
ПРИГОДУ НА МОВІ C#»**

Виконав: студент 4 курсу, групи ПД–41  
спеціальності  
121 Інженерія програмного забезпечення  
(шифр і назва спеціальності/спеціалізації)

\_\_\_\_\_ Маринскас В.М.  
(прізвище та ініціали)

Керівник \_\_\_\_\_ Негоденко О.В.  
(прізвище та ініціали)

Рецензент \_\_\_\_\_  
(прізвище та ініціали)

Київ –2021

# ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

## Навчально–науковий інститут інформаційних технологій

Кафедра Інженерія програмного забезпечення

Ступінь вищої освіти – «Бакалавр»

Спеціальність – 121 Інженерія програмного забезпечення

### ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерія програмного забезпечення

Негоденко О.В.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 року

## ЗАВДАННЯ

### НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

МАРИНСЬКАС ВАДИМ МИКОЛАЙОВИЧ

1. Тема роботи: «Розробка програмного забезпечення для складання протоколу про дорожньо-транспортну пригоду на мові С#»

Керівник роботи: Негоденко О.В кандидат технічних наук, доцент, затверджені наказом вищого навчального закладу від «12» березня 2021 року № 65.

2. Строк подання студентом роботи «01» червня 2021 року

3. Вхідні дані для роботи:

Microsoft Visual Studio , Windows Form, мова програмування С#, науково технічна література з питань тестування програмного забезпечення та розробки десктопного додатку для Windows

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1. Аналіз та характеристика програмного забезпечення

4.2. Концепція

4.3. Аналіз існуючих розробок

- 4.4. Аналіз отриманої інформації
5. Перелік демонстраційного матеріалу
- 5.1. Мета, об'єкт, предмет дослідження
- 5.2. Актуальність
- 5.3. Вибір мови програмування
- 5.4. Середовище розробки
- 5.5. Аналоги
- 5.6. Моделювання об'єктів проектування
- 5.7. Програмні засоби реалізації
- 5.8 Апоботація результатів дослідження
- 5.9. Висновок

Дата видачі завдання: «19» квітня 2021

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір джерел інформації	19.04.2021	Виконано
2	Вимоги до встановленого додатку	23.04.2021	Виконано
3	Оцінка якості тестування до систем	29.04.2021	Виконано
4	Концепція та архітектура додатку	02.05.2021	Виконано
5	Вступ, висновки, реферат	05.05.2021	Виконано
6	Розробка презентації	06.05.2021	Виконано
7	Перед захист диплому	11.05.2021	
8	Задача роботи	01.06.2021	

Студент

Керівник роботи

\_\_\_\_\_

( підпис )

\_\_\_\_\_

( підпис )

Маринскас В.М

(прізвище та ініціали)

Негоденко О.В

(прізвище та ініціали)





## РЕФЕРАТ

Текстова частина бакалаврської роботи: с.49, табл.5, рис.18, джерел.24.

Ключові слова: VISUAL STUDIO, C#,WINDOWS

*Об'єкт дослідження* – підвищення ефективності заповнення електронного протоколу при дорожньо-транспортній пригоді.

*Предмет дослідження* – основні принципи функціонування електронного протоколу про дорожньо-транспортну пригоду.

*Мета роботи* – засоби розробки додатку для складання електронного протоколу про дорожньо-транспортну пригоду.

*Методи дослідження* – метод теорії інформації, метод оптимального управління, обробка та аналіз інформації, методи проектування та розробки програмного забезпечення, валідації та верифікації акаунта.

Для досягнення поставленої мети в даній роботі:

- розроблено алгоритм для покращення системи складання протоколу про дорожньо-транспортну пригоду;
- встановлено використання мови програмування C# та середовища розробки Visual Studio
- розроблено систему для складання протоколів про Дорожньо-транспортну пригоду для поліцейських при виконанні своїх обов'язків.

Упровадження розробленого застосунку допоможе користувачам ознайомитись з основними функціями додатку.

В роботі виконано аналіз існуючих застосунків для операційної системи Windows.

Проаналізовано можливість середовища розробки Visual Studio.

Розроблено логіку додатку та зручність користування для користувачів.

*Галузь використання* - завдяки вільному доступу, додатком може користуватись будь-який працівник органів, який приїхав на виклик по причині Дорожньо-транспортної пригоди.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>9</b>
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....</b>	<b>11</b>
1.1 Аналіз технічних чинників виникнення аварій транспортних засобів .....	12
1.2 Причини виникнення ДТП .....	15
1.3 Дослідження аналогів та прототипів .....	19
<b>2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ.....</b>	<b>22</b>
2.1 Вибір мови програмування для додатку.....	22
2.2 Вибір середовища розробки .....	26
2.3 Microsoft . NET Framework .....	27
2.3.1 .NET Framework для користувачів.....	28
2.3.2 .NET Framework для розробників .....	29
2.4 Windows Form .....	30
2.5 База даних phpMyadmin.....	33
<b>3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ.....</b>	<b>35</b>
3.1 Завдання та можливість програмного забезпечення .....	35
3.2 Моделювання об'єктів проектування .....	36
3.2.1 Діаграма класів .....	36
3.2.2 Діаграма використання.....	37
3.3 Використання бази даних phpMyAdmin .....	38
3.4 Запуск застосунку .....	41
3.5 Тестування застосунку .....	42
<b>4 ІНСТРУКЦІЯ КОРИСТУВАЧА .....</b>	<b>46</b>
<b>ВИСНОВКИ.....</b>	<b>49</b>
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....</b>	<b>50</b>

## УМОВНІ ПОЗНАЧЕННЯ

ООР– об'єктно-орієнтоване програмування

DSP– цифрова обробка сигналів

CLR– загальномовне виконуючого середовища

ASP– сторінки активних серверів

CIL– високорівневий асемблер

MFC– бібліотека для розробки застосунків

CLI– інтерфейс командного рядка

WPF– система для побудови клієнтських додатків Windows

IDE – інтегроване середовище розробки

API – прикладний програмний інтерфейс

XML – розширювана мова розмітки

UML – уніфікована мова моделювання



## ВСТУП

*Актуальність моєї дослідницької роботи* полягає у тому, що в поліцейських виникає з кожним днем все більше роботи, для зменшення часу складання протоколу, тому було прийнято рішення розробити додаток для допомоги працівникам правоохоронних органів складати електронний протокол про дорожньо-транспортну пригоду.

*Мета роботи.* Виходячи з актуальності засобів для складання протоколу про дорожньо-транспортну пригоду метою даної роботи є створення додатку для оформлення електронного протоколу.

Для виконання поставленої мети слід виконати наступні завдання:

1. Вивчити та проаналізувати предметну область.
2. Проаналізувати переваги та недоліки подібних на застосунок який розробляється.
3. Розробити вимоги до нового застосунку.
4. Проаналізувати технічні засоби застосунку
5. Спроекувати та розробити новий додаток на основі аналізу користувачів.

*Об'єкт дослідження* – підвищення ефективності заповнення електронного протоколу при дорожньо-транспортній пригоді.

*Предмет дослідження* – основні принципи функціонування електронного протоколу про дорожньо-транспортну пригоду.

*Мета роботи* – засоби розробки додатку для складання електронного протоколу про дорожньо-транспортну пригоду.

*Методи дослідження* – метод теорії інформації, метод оптимального управління, обробка та аналіз інформації, методи проектування та розробки програмного забезпечення, валідації та верифікації акаунта.

*Практичне значення одержаних результатів.* Додаток який допоможе органам правопорядку складати швидше та без ніяких перешкод оформити електронний протокол про дорожньо-транспортну пригоду

*Результати роботи.* Матеріали дипломного проекту можуть сприяти вдосконаленню розробки електронного протоколу.

Розроблено алгоритм складання протоколу про дорожньо-транспортну пригоду. Для ергономічного використання засобів застосування. На основі результатів виконаних досліджень розроблено систему для складання протоколів про дорожньо-транспортну пригоду для органів правопорядку при виконанні своїх обов'язків.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

На жаль, автомобільні аварії дуже поширені в Україні, і більшість цих дорожньо-транспортних пригод спричинені людською помилкою. Хоча деякі з них відносно незначні, щороку ці жахливі автокатастрофи забирають тисячі життів. Оскільки ваше життя може опинитися під загрозою, якщо ви їдете небезпечно, так важливо їздити акуратно та дотримуватися всіх правил дорожнього руху.

Однак те, що ви обережні, не означає, що ви можете запевнити, що всі інші водії на дорозі будуть робити те саме. Якщо ви потрапили в автокатастрофу, можливо, це не ваша вина, і ви не повинні нести відповідальність за збитки, завдані незнанням або помилками інших водіїв. У таких випадках вам слід задуматися про захист, подавши позов про дорожньо-транспортну пригоду. Внаслідок дорожньо-транспортної пригоди може виникнути стільки збитків, болю та розчарувань, і найкраще охороняти своє життя перш за все.

До найпоширеніших рухомих порушень належать:

- Переїзд через встановлене обмеження швидкості;
- Перебігати проїжджу частину на забороняючі знаки;
- Здійснення незаконного розвороту;
- Пересуватись по смузї громадського руху;

Залежно від серйозності та частоти рухомого порушення, водію може бути призупинено або назавжди анульовано посвідчення. Призупинення дії ліцензії може становити від декількох тижнів до декількох років.

До серйозних порушень належать:

- Водіння без посвідчення водія ;
- Безрозсудне або небезпечне водіння;
- Керування автотранспортом у нетверезому стані;
- Гонки на громадській вулиці;
- Автомобільне вбивство;

Водіння за правопорушення, які звинувачуються як проступки або тяжкі злочини, можуть спричинити тюремний ув'язнення та великі штрафи. Це може коштувати людині тисячі доларів додаткових страхових внесків.

## **1.1 Аналіз технічних чинників виникнення аварій транспортних засобів**

Автомобільні аварії, спричинені несправними транспортними засобами. Хоча безпека автомобіля значно покращилася за останні декілька десятиліть, дефекти транспортних засобів все ще спричиняють автомобільні аварії. Автомобільні фактори стали критичною причиною для 4400 з 2.2 мільйонів аварій, розглянутих в Національному огляді причин наслідку аварій на транспортних засобах, згідно з даними Національної адміністрації безпеки дорожнього руху

### **Поширені типи автоматичних дефектів**

Подушки безпеки, які не розгортаються: якщо подушки безпеки автомобілів не спрацьовують, травми жертви можуть бути ще гіршими. Неважливо, чи винили ви чи хтось інший, подушки безпеки, що не розігнали, можна розглядати як дефектну деталь.

Подушки безпеки, які розгортаються, коли вони не повинні: Деякі автомобільні аварії трапляються через те, що подушки безпеки розгортаються тоді, коли їх не було. Це не тільки дратує досвід, але й ризикує, оскільки може спричинити перешкоду, яка призводить до нещасного випадку. Для автомобіля небезпечно мати несправні подушки безпеки, які розмикаються простим краном або не мають удару.

Подушки безпеки розгортаються занадто швидко із занадто великою силою: коли це трапляється, це може призвести до серйозних травм. Подушки безпеки призначені для мінімізації наслідків дорожньо-транспортної пригоди та не погіршення ситуації. Якщо вони розгортаються із занадто великою силою, вони можуть спричинити серйозні травми кісток, а іноді і смерть.

Ремені безпеки, що відстібаються під час аварії: Під час аварії ремінь безпеки повинен залишатися цілим, щоб запобігти викиданню пасажирів автомобіля через лобове скло. Якщо ремінь безпеки розстібається під час аварії, це забезпечує менший захист та спричиняє серйозніші травми.

Забезпечення належного захисту від перекидання: Під час аварії при перекиданні автомобіль розроблений для забезпечення належного захисту. Такі функції, як поворотні бари, можуть захистити людей всередині, якщо машина перекинеться.

Дефекти електричної системи: часто відомо, що вони спричиняють серйозні пожежі на транспорті. Якщо несправна електрична система, машина може мимовільно загорітися. Зазвичай це якась ознака дефекту.

Несподіване прискорення: це трапляється, коли водій втрачає контроль над прискоренням автомобіля. Це небезпечний дефект, який може спричинити смертельні аварії.

Відмова гальм: гальма можуть несподівано вийти з ладу. Коли це трапиться, це може спричинити серйозну автомобільну аварію.

Несправність гідро підсилювача керма: можна керувати автомобілем, навіть коли підсилювач керма виходить з ладу. Однак раптова втрата контролю може стати несподіванкою та призвести до аварії.

Несправність шин: Це може призвести до втрати контролю над транспортним засобом, що призведе до серйозних аварій.

Є багато інших дефектів транспортних засобів, які можуть спричинити серйозні аварії або погіршити травми потерпілого. Оскільки в побудові автомобіля існує так багато елементів, потенційних точок відмови буде незліченна. Багато з цих дефектів можуть створити небезпечні дорожні умови. Щоб дізнатись, чи можна усунути несправність транспортного засобу, зверніться до юриста з питань відповідальності за товар.

Якщо ви потрапили в аварію, спричинену автодефектом, важливо зібрати всі докази, які допоможуть підтвердити вашу претензію. Сюди входять фотографії транспортного засобу, а також будь-які відеозаписи, які можуть бути у вас із місця

події. Не поспішайте ремонтувати машину, поки не поговорите з адвокатом щодо інциденту. Якщо можливо, переконайтеся, що транспортний засіб зберігається у безпечному місці, доки адвокат не зможе додатково розслідувати та оглянути його на наявність дефектів. Претензії до автодефектів складні і вимагають кваліфікованої юридичної допомоги для успішного орієнтування. Виробники автомобілів та автомобільних деталей повинні дотримуватися Федеральних стандартів та правил безпеки автомобілів, виданих NHTSA. МНССА відкликала мільйони транспортних засобів та автозапчастин за невиконання цих стандартів.

Деякі приклади дефектів, пов'язаних із безпекою, включають:

- Прискорювачі, які стирчать або ламаються;
- Подушки безпеки, які розгортаються тоді, коли не повинні;
- Проблеми з електричною системою, які спричиняють пожежу або втрату світла;
- Колеса, які тріскаються або ламаються, спричиняючи втрату контролю над транспортним засобом;
- Компоненти рульового управління, які ламаються і спричиняють втрату контролю над автомобілем;
- Компоненти паливної системи, сприйнятливі до аварійних пошкоджень, які витікають та спричиняють пожежу;
- Вузли склоочисника, які не працюють належним чином;
- Сидіння або спинки сидінь, які несподівано виходять з ладу;

Шини або інші важливі компоненти транспортного засобу, які ламаються, розпадаються або відокремлюються від транспортного засобу, що може призвести до втрати контролю над транспортним засобом та травмування людей у транспортному засобі чи поза ним.

Види травм, спричинених несправними транспортними засобами

Несправні транспортні засоби та автозапчастини можуть спричинити нещасні випадки та травми двома різними способами:

Коли справжньою причиною аварії є дефектний транспортний засіб - наприклад, при руйнуванні рульового управління, несподівано спрацює

подушка безпеки, вибухає шина або відключається запалювання, в результаті чого водій втрачає контроль над автомобілем.

Коли дефекти транспортного засобу спричиняють травми внаслідок аварії більш серйозні, ніж би були в іншому випадку - наприклад, коли виходить з ладу подушка безпеки, виходить з ладу ремінь безпеки, транспортний засіб перевертається при повороті або дах руйнується при перекиданні автомобіля.

#### Відповідальність та компенсація за автодефекти

Виробники автомобілів та автозапчастин зобов'язані забезпечити достатньо безпечну їхню продукцію або адекватно попередити споживачів про наявність небезпек, як діючих, так і виявлених пізніше. У разі невиконання цього обов'язку потерпіла сторона може мати право подати позов про відповідальність за відшкодування збитків, заподіяних пошкодженням товаром.

Залежно від обставин, заява про пошкодження автотранспорту може бути подана проти виробника автомобіля, виробника дефектної деталі, компанії, яка розповсюдила транспортний засіб, автодилера чи інших осіб.

Потерпілі жертви автомобільних аварій, спричинених автомобільними дефектами, можуть мати право подати позов про компенсацію медичних витрат, майнової шкоди, втрати заробітної плати, болю та страждань та інших збитків. У деяких виняткових випадках може бути також можливим стягнення збитків у вигляді покарання, призначених для покарання відповідальної сторони.

## **1.2 Причини виникнення ДТП**

Усім водіям потрібно безпечно та розумно керувати своїми автомобілями. Однак люди не завжди роблять це, і людські фактори, що призводять до небезпечної їзди, є основною причиною дорожньо-транспортних пригод.

Крім того, закон забороняє водіям експлуатувати свої транспортні засоби, якщо вони вживали наркотики або алкоголь понад норму, встановлену законом. Недотримання цих законів є помилкою суджень, яка зазвичай спричиняє нещасні випадки. В інших випадках водіям може просто не вистачати досвіду водіння, або

вони можуть не визнати, що їх вік або проблеми зі здоров'ям заважають їхній безпечній їзді. Навіть грамотні водії можуть часом відволікатися, і якщо не звертати уваги навіть на кілька секунд, це може спричинити ДТП. Різні людські фактори можуть спричинити нещасні випадки. Закони України вимагають від усіх водіїв дотримуватися безпечної швидкості руху, дотримуватися безпечної відстані від будь-якого автомобіля, що знаходиться перед ними, та підкорятися пристроям управління дорожнім рухом для забезпечення безпеки інших водіїв та пасажирів.

Хоча більшість нещасних випадків спричинені перевищенням швидкості, відволіканням від руху та нетверезими водіями, погана якість дороги та умови є важливим фактором багатьох аварій.

Іншими словами, навіть якщо погані дорожні умови не є єдиною причиною аварії, вони, ймовірно, значно погіршили ситуацію. Загальні дорожні умови, які сприяють дорожньо-транспортним пригодам:

Пошкоджені, відсутні або приховані дорожні знаки  
Вибоїни та тріщини на дорожньому покритті  
Відсутність гуркоту смуг на шосе  
Відсутні або неадекватні огороження  
Марки, що вицвіли фарбою, як розмітка на центральній лінії або краю дороги  
Висадка плеча дороги  
Дорожнє сміття  
Погане управління дорожнім рухом навколо будівельних зон  
Не оброблені дороги в зимову погоду.  
Захистить свою претензію з самого початку  
Незалежно від того, потрапили ви в аварію з одним транспортним засобом, спричинену поганою дорожньою обстановкою, або аварію на декількох автомобілях, знання того, що робити з самого початку, допоможе скласти серйозну заяву про пошкодження.

Погані дорожні умови можуть призвести до більш ніж просто нерівної їзди. Дороги можуть погіршитися до рівня, коли вони стають небезпечними. Сюди входять вибоїни, нерівності дорожніх покриттів, битий бетон, відкрита арматура, поглиблення та дорожні тріщини. Якщо водій потрапляє у велику вибоїну, це може розірвати шину, в результаті чого транспортний засіб виїде на іншу смугу, зіткнувшись з іншим транспортним засобом. Нерівні дорожні покриття можуть призвести до того, що водій втратить контроль над своїм транспортним засобом,



що призведе до аварії або аварії з перекиданням, що поранить водія, пасажирів та пішоходів.

Небезпечні дорожньо-транспортні пригоди можуть призвести до серйозних травм, що загрожують життю. Це включає травму голови, черепно-мозкову травму, шию або спину, зламані кістки, травми очей та внутрішні травми. Це може призвести до поїздки швидкої допомоги до лікарні швидкої допомоги, хірургічного втручання, лікування та тривалого перебування в лікарні. Деякі травми можуть вимагати дорогої медичної допомоги протягом усього життя.

Навіть незначна аварія може призвести до травм. Після багатьох аварій на низькій швидкості водій та пасажир можуть спочатку почуватися добре. Однак через пару днів вони можуть зрозуміти, що отримали більші травми, ніж вони спочатку вважали. Вони можуть страждати від болю в шиї, головного болю, поколювання в кінцівках та відсутності рухливості, що часто спричиняється травмами шиї або хребта, такими як хлистові пошкодження. Якщо є шанс, що ви постраждали в результаті нещасного випадку, вам слід звернутися за медичною допомогою.

Якщо ви дбайливий водій, ви, мабуть, не зводите очей з дороги, їдете оборонно і тримаєтеся на відстані від небезпечних автомобілістів.

Однак нещасні випадки можуть трапитися навіть з найобережнішими водіями, і іноді вони трапляються через погану безпеку на дорозі.

Є багато дорожніх умов, які можуть призвести до втрати контролю над вашим автомобілем і аварії, зокрема:

- Предмети на дорозі;
- Лід і сніг, лід;
- Заплутані знаки;
- Відсутність знаків;
- Вибоїни Круті плечі;
- Небезпечні робочі зони Об'єднання води ;
- Вітряні дороги без ліній Колії;

Проїжджаючи в середньому понад 13000 км на рік, українські водії спостерігають багато в поїздках. Рухаючись дорогою зі швидкістю 40, 50 або навіть 60 з лишком км на годину, це також одне з останніх місць, де ви хотіли б зустріти пухнастих друзів. Це серйозна проблема безпеки, яка вимагає життя людей і тварин, але ми рідко зупиняємося, щоб розглянути масштаби проблеми. У певний рік відбувається понад 260 000 аварій, в яких задіяні тварини, що спричиняє 12 000 людських поранень.

Щоб уникнути сумних статистичних даних, подібних до наведених вище, важливо, щоб ми якомога безпечніше вивозили живу природу з проїжджих частин, де вони можуть поранитися, а також загрожувати водіям.

Ймовірні тварини які вибігають на проїжджу частину

Деякі інші популярні тварини - білки, коти, собаки та птахи. На думку дослідників, допомогти забезпечити безпеку цих тварин у дорозі сьогодні та завтра, особливо з появою автомобілів без водіїв. Це пов'язано з реальністю, з якою стикаються водії, коли вони помічають тварину на дорозі або поруч із нею і мусять здогадуватися, що їм робити далі. Уважність - це перший крок до того, щоб допомогти зберегти дороги чистими для всіх тварин, з якими ми стикаємось.

Відповідно до Програми запобігання зіткненню з дикою природою, Ви можете вжити заходів, щоб забезпечити безпеку для себе та дикої природи

- Слідкуйте за швидкістю;
- Будьте обачні;
- Майте на увазі попереджувальні знаки дикої природи;

Подумайте, «а якщо» тварина вибігла на дорогу; якою б була ваша тактика, щоб її уникнути

Спільні дороги для зустрічей тварин

Найчастіше вражають білок. Для білок в будь-якому місці небезпечно, оскільки скрізь, де ви їх очікуєте - мікрорайони, сільські дороги та шосе - були найпопулярнішими місцями, де автомобілісти збивали цих істот.

Околиці та сільські дороги виявились зоною бою майже для кожної істоти. Птахів, оленів, собак, котів, кроликів і скунсів частіше вражали на цих дорогах.

## Збереження життя та здоров'я тваринам

Ніхто не хоче збити тварину. Це не тільки страшно для водія та пасажирів, але й може викликати серце. Більшість водіїв роблять все можливе, щоб уникнути наїзду тварин на дорогу, але в цю частку секунди вони стикаються з важким вибором.

Це означає, що багато хто з нас стикалися з болісним досвідом принаймні один раз, розриваючись між чотирма або п'ятьма різними реакціями, і лише на частку секунди для прийняття рішення. Вирішіть правильно, і тварина в безпеці - і вам, і тим, хто у вашому транспортному засобі, слід відійти від місця події. Однак вибирайте погано, і це може коштувати не просто грошей, а життя тварин і коханих.

### 1.3 Дослідження аналогів та прототипів

Кожного дня, в світі гинуть понад 3 тисяч людей і тисячі потрапляють в лікарню. В основному дорожньо-транспортна пригода стається із за не уважності, алкогольного сп'яніння, перевищення швидкості та багато інших аспектів які можуть призвести до ДТП. Складання протоколу дорожньо-транспортної пригоди займає багато часу водіїв та поліцейських які їх складають. Додатки для автоматизації цього процесу вже існують. Прямих аналогів для порівняння не доступні у використанні, але є аналоги які частково схожі додаток який розробляється. Ось представлені додатки які згадувались вище. Розглянемо основні з них які доступні у використанні.

Електронний Євро протокол - це веб-сайт, українською мовою який був створений у 2016 році. Для автоматизації процесу складання електронного євро протоколу з збоку водіїв. Додаток має багато сторінок для заповнення різної інформації. Такої як інформація про транспортні засоби, інформація про водіїв з можливістю додавання картинок та фото ДТП. З можливістю автоматичного знаходження місця дислокації через функцію місцезнаходження та багато інших зручних функцій які присутні в цьому додатку. Нижче буде показана одна із сторінок сайту рис. 1.1.

Рисунок 1.1 – Одна із сторінок цього web-застосунка

Мій Поліс- мобільний додаток який підтримує платформи Android та IOS. Для зберігання даних про страховий поліс та складання електронного євро протоколу у разі ДТП. Перехід на нову сторінку відбувається відкриттям новим блоком. На даному рисунку буде зображений скріншот на платформі IOS рис. 1.2. Застосунок підтримує поліси усіх компаній які надають послуги страхування.

Рисунок 1.2 – Сторінка додавання полісу

Узагальнене порівняння переваг та недоліків представлено в табл. 1.1.

Таблиця 1.1 – Переваги та недоліки існуючих додатків

Назва застосунку	Переваги	Недоліки
Електронний європротокол	<ul style="list-style-type: none"> <li>- Багатий функціонал</li> <li>- Простота у використанні</li> <li>- Зберігання безпечно інформації в базі даних</li> <li>- Можливість зробити фотозвіт</li> <li>- Геолокація місця ДТП</li> <li>- Прораховані всі варіанти які можуть призвести до аварії</li> <li>- Перевірка введених даних</li> <li>- Можливість відправлення на електронну пошту обом учасникам ДТП</li> </ul>	Потребує підключення до мережі інтернет
Мій поліс	<ul style="list-style-type: none"> <li>- Зручний в користуванні</li> <li>- Можливість сканування QR- кодом</li> <li>- Можливість складання електронного європротоколу</li> <li>- Швидкість роботи додатку</li> </ul>	Спливаюче вікно не несе ніякої потрібної інформації

## 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ

### 2.1 Вибір мови програмування для додатку

Вибір програмних засобів- це саме головне перед тим, як почати розробляти будь-який додаток. Реалізація програмного забезпечення, а саме розробка додатку для допомоги складання протоколу.

На сьогодні існує багато різних мов програмування з основних це Java, C++ та C#. Почнемо з мови програмування Java є однією з найпопулярніших мов програмування, головним чином через те, наскільки вона універсальна і сумісна. Java може використовуватися для великої кількості речей, включаючи розробку програмного забезпечення, мобільних додатків та розробку великих систем Також C та C++ є не відмінною частиною у вивченні програмування. C та C++ - це потужна мова програмування загального призначення.. З його допомогою можна розробляти операційні системи, браузері, ігри тощо. C++ підтримує різні способи програмування, такі як процедурне, об'єктно-орієнтоване, функціональне тощо. Це робить

C++ потужним, а також гнучким. У цьому вступі давайте розглянемо його основні атрибути та чому він залишається таким популярним.

Але на думку багатьох, вам потрібно навчитися програмуванню на C#.

Мова програмування C# - сучасна мова програмування. Він отримує свої характеристики від своїх попередників, таких як C та C++. Він може виконувати широкий спектр завдань та процесів у різних нішах.

Серед інших мов, C# набуває величезної популярності для розробки веб-додатків. Її основні концепції допомагають створити інтерактивне середовище та надають функціональні можливості, необхідні динамічній веб-платформі.

Більшість початківців розробників повних стеків вибирають цю універсальну мову. Він працює на обох платформах: Windows .NET Framework та інших платформах з відкритим кодом. Тому це забезпечує неймовірну ефективність та масштабованість.

Microsoft розробила C # у 2000 році. Він був розроблений для задоволення зростаючого попиту на веб-програми, яким Visual Basic (VB) та C ++ не можуть задовольнити. Його архітектура враховує найкращі можливості Java та C ++. Як результат, розробники, які добре знаються на C та C ++, можуть легко перейти на C #.

- Крім того, його величезна популярність пояснюється:
- Багаторазові компоненти для швидшої розробки програмного забезпечення;
- Синтаксис коду схожий на Java та C ++;
- Типи даних всередині C # є більш гнучкими та без помилок;

Це лише кілька причин, які роблять C # найкращою мовою програмування, коли мова йде про веб-додатки. Він забезпечує широкий спектр компонентів, що охоплюють бізнес, орієнтований на системно-орієнтовані проекти.

C # - це мова програмування загального призначення, що використовується для побудови різних типів програм та додатків. Незважаючи на те, що він надзвичайно універсальний, існує три поля, де він зазвичай застосовується:

Незалежно від платформи, ви все одно можете використовувати мову програмування C #.

Ви можете створювати динамічні веб-сайти та веб-програми, використовуючи платформу .NET або інші платформи з відкритим кодом.

Microsoft створила C # для Microsoft. Отже, немає сумнівів, чому він популярний у створенні додатків для Windows. Це робить ваш процес розробки плавним, а такі функції, як збір сміття на C #, працюють чудово.

Крім того, розробники можуть розраховувати на підтримку спільноти та документацію щодо розробки додатків та програм, характерних для архітектури платформи Microsoft.

У світі ігор розробники віддають перевагу мові програмування C #. Ця мова особливо міцна при створенні улюблених ігор фанатами. Крім того, ігровий движок Unity — один із найпопулярніших ігрових рухів сьогодні — побудований за допомогою C ++ та C #.

Розробники ігор використовують ігрові двигуни, такі як Unity, для створення відеоігор. У ньому понад мільярд активних користувачів. Це також надійна платформа для 1,5 мільйона активних користувачів у всьому світі. Ці активні користувачі продовжують розробляти відзначені нагородами ігри та досвід AR / VR.

C # легко інтегрується з движком Unity. Для розробників мобільних пристроїв його можна використовувати практично на будь-якому сучасному мобільному пристрої чи консолі, використовуючи кроссплатформенні технології, такі як Xamarin. Отже, якщо ви хочете проникнути в індустрію розробки ігор або приєднатися до віртуальної реальності, тоді ви можете використати потенціал мови програмування.

Програмування на C # є найбільш популярним серед набору мов Microsoft. Ось кілька статистичних даних, що підтверджують його популярність:

За результатами опитування StackOverflow у 2019 році 31,9% професійних розробників віддають перевагу мові програмування C #, і вона має третю за величиною спільноту в тому ж опитуванні

Він отримав 67% схвалення як найулюбленіша мова програмування у світі (опитування StackOverflow 2019).

І його переваги на цьому не зупиняються. Він може похвалитися:

Найбільшою перевагою є його здатність економити час на розробку. Ось деякі його атрибути, які постійно допомагають розробникам та власникам проектів:

- Статично друкований і легкий для читання, що зменшує час налагодження;
- Величезна бібліотека, що забезпечує високорівневі функціональні можливості порівняно з іншими мовами, такими як Java та C ++;
- Пропонує простоту та ефективність і допомагає програмістам писати складні стеки коду;
- Великий банк пам'яті для скорочення часу розробки та зменшення часу виходу на ринок;
- Крива низького рівня навчання;



Розробники, особливо розробники з повним стеком, віддають перевагу мові програмування C #. Це не тільки може заощадити час на розробку, але й має низький рівень навчання. Розробники витрачають менше часу на вивчення C # порівняно з іншими мовами програмування.

Його простота та прості у використанні функції допомагають новим розробникам та тим, хто хоче вивчити нову мову.

Для старших розробників, що працюють над старими проектами, написаними на C #, ви із задоволенням побачите, що ваш стек залишився незмінним. З роками процеси, можливо, змінювались, але все ще залишається на місці.

Оскільки технічний гігант Microsoft створив C #, немає сумнівів у тому, чому він має величезну підтримку з боку громади. У світі розробки програмного забезпечення важливо мати спільноту. Тут розробники можуть шукати допомогу та інформацію. Мови програмування не є платформою чи послугою із спеціальною лінією довіри або IT-підтримкою.

Розробники покладаються на інших у тій самій галузі інтересів. Вони знаходять відповіді та рішення у тих, хто зазнав однакових перешкод у розвитку.

Наприклад, однією з найбільших спільнот є StackOverflow. Цей веб-сайт розробника дає змогу розробникам C # запитувати, відповідати, міркувати або навіть висловлювати свої розчарування.

Програмування на C # - одна з найпопулярніших мов програмування у світі розробки програмного забезпечення. Наприклад, давайте подивимося, як компанія будує свій продукт. Найчастіше, коли йдеться про надійні веб-програми (наприклад, SaaS, PaaS), використовується C #.

Таким чином, розробники можуть використовувати потенціал C # та вивчати мову. Для розробників вивчення програмування на C # може покращити ваші навички на роботі. А для новачків ви можете навіть вирішити продовжувати це як кар'єру! Прочитайте цю статтю, щоб дізнатися більше про те, де навчитися програмуванню на C #.

C # повністю об'єктно-орієнтований. Рідкісний атрибут, якого не вдається досягти більшості поширених мов. Об'єктно-орієнтоване програмування (ООР)

пропонує ефективність та гнучкість, наприклад, створення програмного забезпечення для управління продуктивністю програм (APM). C # разом з іншими мовами створює такий тип програмного забезпечення.

Впровадження рішення APM може допомогти розробникам програмного забезпечення оптимізувати продуктивність під час постановки, виробництва та пост-продукції. Він надає статистичні дані щодо продуктивності програми та способи усунення проблем, коли вони виникають.

Розробка програмного забезпечення вимагає простих в обслуговуванні та масштабованих мов програмування. C # - це мова програмування, яка має ці атрибути. Строгий характер його статичного кодування робить програми надійно послідовними. Це дозволяє розробникам зробити це просто.

## **2.2 Вибір середовища розробки**

Існує багато середовищ для розробки, а саме можна виділити декілька видів.

Текстовий редактор, транслятор-це компілятор та інтерпретатор та інші.

Давайте розберемось з кожним із вищеперечисленого. Текстовий редактор-це самостійна комп'ютерна програма для створення текстових даних та файлів. Транслятор- це програма для відтворення написаної програми.

Для розробки було прийнято рішення обрати середовище розробки транслятор. Тому що програма яка написана має відобразитись у вікні виводу даних – це вікно компілятора.

Середовище розробки - це сукупність процедур та інструментів для розробки, тестування та налагодження програми чи програми.

Зазвичай середовище розробки має три рівні серверів, які називаються розробкою, постановкою та виробництвом. Всі три рівні разом зазвичай називають DSP. Для розробки додатку було прийнято рішення використовувати інтегровану середовище розробки Visual Studio.

Visual Studio – це продукт компанії Microsoft включаючи інтегровану середовище розробки та багато різних інструментів для програмування .

У Visual Studio має велику кількість функцій та інструментів для програмування на різні мови програмування майже на всіх мовах програмування які існують. Для наглядного прикладу нижче буде представлено скріншот з основним інтерфейсом Visual Studio рис. 2.3.

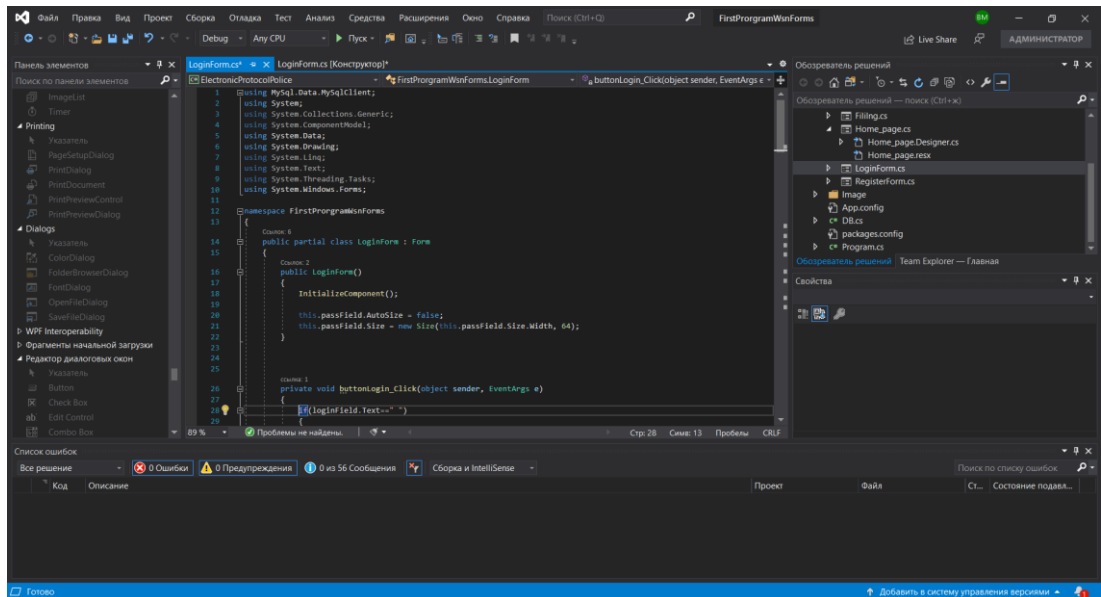


Рисунок 2.1 – Інтерфейс інтегрованої середовища розробки

## 2.3 Microsoft .NET Framework

.NET Framework - це середовище виконання під час виконання, яке керує програмами, націленими на .NET Framework.

Він складається із загальномовної середовища виконання, яка забезпечує управління пам'яттю та інші системні послуги, та великої бібліотеки класів, дозволяє програмістам скористатися надійним кодом для всіх основних областей розробки додатків.

.NET Framework - це середовище керованого виконання для Windows, яке надає різноманітні послуги своїм запущеним додаткам. Він складається з двох основних компонентів: загальномовної середовища виконання (CLR), що є механізмом виконання, який обробляє запущені програми, та бібліотеки класів .NET Framework, яка забезпечує бібліотеку перевіреного коду, що багаторазово

використовується, який розробники можуть викликати з власних програм . Послуги, які .NET Framework надає запущеним додаткам, включають наступне:

Управління пам'яттю. У багатьох мовах програмування програмісти відповідають за розподіл і вивільнення пам'яті та за обробку часу життя об'єкта. У програмах .NET Framework CLR надає ці послуги від імені програми.

Поширена система типів. У традиційних мовах програмування базові типи визначаються компілятором, що ускладнює взаємодію між мовами. У .NET Framework базові типи визначаються системою типів .NET Framework і є загальними для всіх мов, націлених на .NET Framework.

Велика бібліотека класів. Замість того, щоб писати величезну кількість коду для обробки типових операцій програмування низького рівня, програмісти використовують легкодоступну бібліотеку типів та їх членів із бібліотеки класів .NET Framework.

Структури та технології розвитку. .NET Framework включає бібліотеки для певних областей розробки програм, таких як ASP.NET для веб-програм, ADO.NET для доступу до даних, Windows Communication Foundation для сервісо-орієнтованих програм та Windows Presentation Foundation для настільних програм Windows.

Мовна сумісність. Мовні компілятори, націлені на .NET Framework, видають проміжний код із назвою Common Intermediate Language (CIL), який, у свою чергу, компілюється під час виконання загальномовною середовищем виконання. Завдяки цій функції програми, написані однією мовою, доступні для інших мов, і програмісти зосереджуються на створенні програм на бажаних мовах.

### 2.3.1 .NET Framework для користувачів

Якщо ви не розробляєте програми .NET Framework, але використовуєте їх, вам не потрібно мати конкретних знань про .NET Framework або його роботу. Здебільшого фреймворк є повністю прозорим для користувачів.

Якщо ви використовуєте операційну систему Windows, на вашому комп'ютері може бути вже встановлено .NET Framework. Крім того, якщо ви

встановлюєте програму, яка вимагає .NET Framework, програма встановлення програми може встановити певну версію платформи на ваш комп'ютер. У деяких випадках може з'явитися діалогове вікно із запитом на встановлення .NET Framework.

Якщо ви щойно спробували запустити програму, коли з'явиться це діалогове вікно, і якщо ваш комп'ютер має доступ до Інтернету, ви можете перейти на веб-сторінку, яка дозволяє встановити відсутні версії .NET Framework.

Для отримання додаткової інформації див. Посібник із встановлення.

Загалом, не слід видаляти версії .NET Framework, які встановлені на вашому комп'ютері. Цьому є дві причини:

Якщо програма, яку ви використовуєте, залежить від конкретної версії .NET Framework, ця програма може зламатися, якщо її видалити.

Деякі версії .NET Framework - це місцеві оновлення попередніх версій. Наприклад, .NET Framework 3.5 - це місцеве оновлення до версії 2.0, а .NET Framework 4.8 - місцеве оновлення до версій 4 - 4.7.2. Щоб отримати додаткові відомості, див. Версії та залежності .NET Framework.

У версіях Windows до Windows 8, якщо ви вирішите видалити .NET Framework, завжди використовуйте програми та функції з панелі керування, щоб видалити його. Ніколи не видаляйте версію .NET Framework вручну. У Windows 8 та новіших версіях .NET Framework є компонентом операційної системи, і його не можна самотійно видалити.

Кілька версій .NET Framework можуть одночасно співіснувати на одному комп'ютері. Це означає, що вам не потрібно видаляти попередні версії, щоб встановити пізнішу версію.

### 2.3.2 .NET Framework для розробників

Якщо ви розробник, виберіть будь-яку мову програмування, яка підтримує .NET Framework, щоб створювати свої програми. Оскільки .NET Framework забезпечує мовну незалежність та сумісність, ви взаємодієте з іншими програмами

та компонентами .NET Framework незалежно від мови, з якою вони були розроблені.

Щоб розробити програми або компоненти .NET Framework, виконайте такі дії:

Якщо він не попередньо встановлений у вашій операційній системі, встановіть версію .NET Framework, на яку буде націлено вашу програму. Останньою версією є .NET Framework 4.8. Він попередньо встановлений в Windows 10 May 2019 Update і доступний для завантаження в попередніх версіях операційної системи Windows.

Системні вимоги до .NET Framework див. У розділі Системні вимоги. Інформацію про встановлення інших версій .NET Framework див.

У Посібнику з інсталяції. Додаткові пакети .NET Framework випускаються поза діапазоном, що означає, що вони випускаються постійно поза будь-яким регулярним або запланованим циклом випуску.

Виберіть мову або мови, які підтримує версія .NET Framework, які ви збираєтеся використовувати для розробки своїх програм. Доступна низка мов, зокрема Visual Basic, C #, F # та C ++ / CLI від Microsoft. (Мова програмування, яка дозволяє розробляти програми для .NET Framework, дотримується специфікації Common Language Infrastructure (CLI).)

Виберіть та встановіть середовище розробки, яке використовуватиметься для створення ваших програм та підтримує вибрану вами мову програмування. Інтегрованим середовищем розробки Microsoft (IDE) для програм .NET Framework є Visual Studio. Він доступний у ряді видань.

## **2.4 Windows Form**

Для розробки додатку було прийнято рішення обрати такий набір керованих бібліотек Windows Form призначений для розробки розширених клієнтських додатків. Це графічний API для відображення даних та управління взаємодією

користувачів із простішим розгортанням та кращим захистом у клієнтських програмах.

Windows Forms пропонує велику клієнтську бібліотеку, що забезпечує інтерфейс для доступу до власних елементів графічного інтерфейсу Windows та графіки з керованого коду. Він побудований з керованою подіями архітектурою, подібною до клієнтів Windows, а отже, його програми чекають введення користувачем для його виконання.

Windows Forms подібний до бібліотеки Microsoft Foundation Class (MFC) при розробці клієнтських програм. Він забезпечує обгортку, що складається з набору класів C++ для розробки програм Windows. Однак він не забезпечує стандартну програму програм, як MFC.

Кожен елемент керування у програмі Windows Forms є конкретним екземпляром класу. Макетом елемента керування в графічному інтерфейсі та його поведінкою управляють за допомогою методів та засобів доступу. Windows Forms надає різноманітні елементи керування, такі як текстові поля, кнопки та веб-сторінки, а також параметри створення спеціальних елементів керування. Він також містить класи для створення пензлів, шрифтів, піктограм та інших графічних об'єктів (таких як лінія та коло).

Windows Forms Designer - це інструмент у Visual Studio.NET, який використовується для вставки елементів керування у форму та упорядкування їх відповідно до бажаного макета, з можливістю додавання коду для обробки їхніх подій, які реалізують взаємодію користувачів. Табличні дані, прив'язані до XML, бази даних тощо, можуть відображатися за допомогою елемента керування DataGridView у вигляді рядків та комірок.

Налаштування додатків - це ще одна особливість Windows Forms для створення, зберігання та підтримання інформації про стан виконання у формі XML, яку можна використовувати для отримання налаштувань, які надають перевагу користувачеві, таких як позиції на панелі інструментів та останні списки. Ці налаштування можна використовувати повторно в майбутньому додатку

Деякі з найкращих практик побудови програм Windows Forms включають:

Класи Windows Forms можна розширити, використовуючи успадкування, щоб розробити структуру програми, яка може забезпечити високий рівень абстракції та багаторазового використання коду.

Форми повинні бути компактними, елементи керування обмеженими розміром, який може запропонувати мінімальну функціональність. Крім того, створення та видалення елементів керування динамічно може зменшити кількість статичних елементів керування.

Форми можна розбивати на шматки, упаковані в збірки, які можуть автоматично оновлюватись і ними легко керувати з мінімальними зусиллями.

Проектування програми для бездержавного забезпечення забезпечує масштабованість та гнучкість із легкістю налагодження та обслуговування.

Додатки Windows Forms повинні розроблятися з урахуванням рівня необхідної довіри, потреби запитувати дозволи та обробляти винятки безпеки, де це необхідно.

Форму Windows не можна передати через межу домену програми, оскільки вона не призначена для маршалювання між доменами додатків.

Презентаційна платформа Windows (WPF) - це найновіша технологія візуалізації користувальницьких інтерфейсів у програмах графічного інтерфейсу Windows із такими функціями, як підтримка 2D / 3D, інтерактивна візуалізація даних та читабельність вмісту. Він покладається на DirectX, а не на об'єкти GDI (графічний інтерфейс пристрою) для забезпечення моделі програмування, де користувальницький інтерфейс відокремлений від бізнес-логіки. Однак завдяки можливості взаємодії з WPF (де завгодно), Windows Forms є хорошим вибором для програм, які не потребують графічного інтерфейсу, багатого на мультимедіа, та інших функцій WPF, таких як шаблони даних / управління, друкарські функції та функції рендерингу тексту. Використання Windows Form для розробки дизайну застосунку

Для розробки дизайну застосунку було прийнято рішення використовувати Windows Form зручний у використанні та з можливістю відразу перевірити зроблений результат. Ось зображено на рис.2.2.



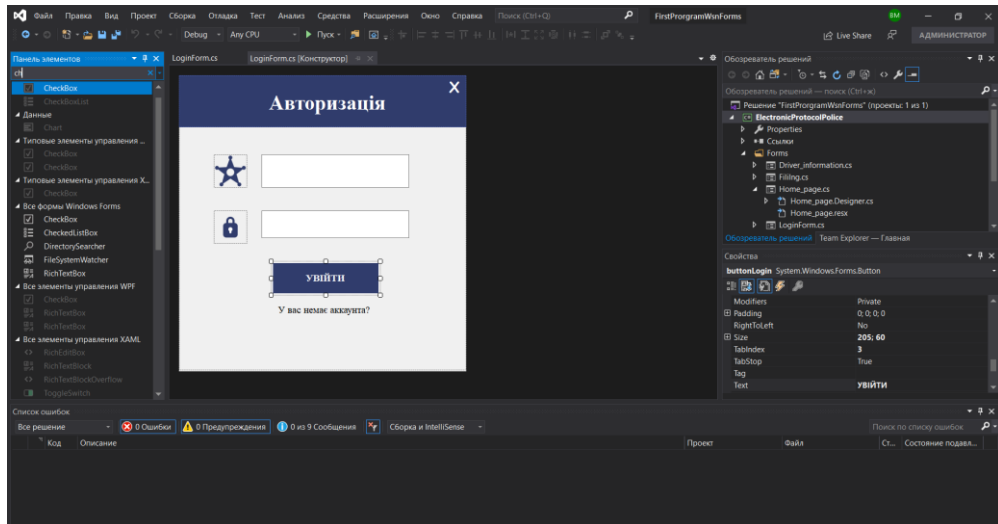


Рисунок 2.2 – Розробка вікна авторизації

Можливість швидко коригувати свої данні. Також є можливість вибору подій для ваших елементів. Зображено на рис. 2.3.

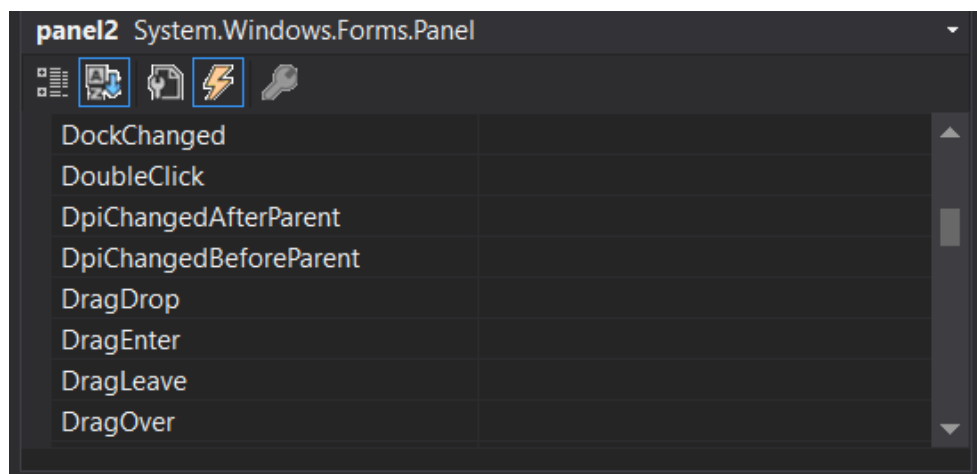


Рисунок 2.3 – Різномноманіття вибору подій

## 2.5 База даних phpMyadmin

Існує багато різноманітних видів баз даних. Але я зупинив свій вибір на базі даних phpMyadmin- це інструмент для адміністрування даних написаний на мові програмування PHP. Основна мета інструменту phpMyAdmin- керувати даними і в нього це не погано виходить звісно якщо правильно налаштувати систему на

роботу. Для того щоб легше було сприймати інформацію про цю базу даних нижче буде зображено рис. 2.4.

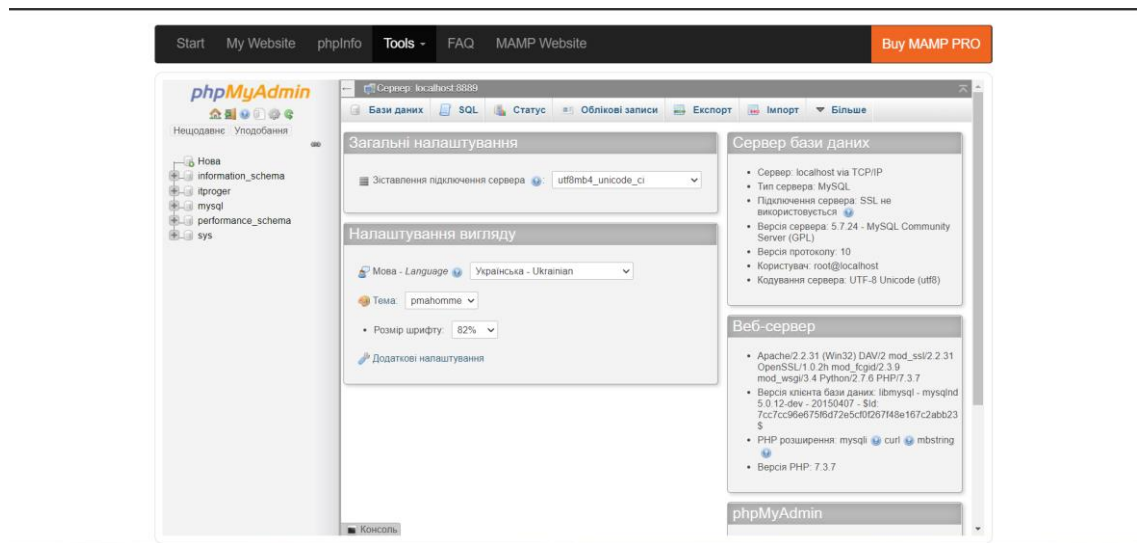


Рисунок 2.4 – Головне вікно бази даних phpMyAdmin

З лівого боку зображено бази даних на серверах MySQL. Так давайте ж розберемось що таке MySQL - це система управління базами даних, яка дозволяє управляти реляційними базами даних. Це програмне забезпечення з відкритим кодом, яке підтримується Oracle. Це означає, що ви можете використовувати MySQL, не платячи ні копійки. Крім того, якщо ви хочете, ви можете змінити його вихідний код відповідно до ваших потреб.

Незважаючи на те, що MySQL є програмним забезпеченням з відкритим кодом, ви можете придбати комерційну ліцензійну версію в Oracle, щоб отримати послуги преміум-підтримки.

MySQL досить легко освоїти в порівнянні з іншим програмним забезпеченням для баз даних, таким як Oracle Database або Microsoft SQL Server.

MySQL може працювати на різних платформах UNIX, Linux, Windows тощо. Ви можете встановити його на сервері або навіть на робочому столі. Крім того, MySQL надійний, масштабований і швидкий.

## **3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ.**

### **3.1 Завдання та можливість програмного забезпечення**

Коли трапляється дорожньо-транспортна пригода і водії не можуть по якимось поважним причинам скласти євро протокол. Вони викликають наряд поліції, який приїжджає на виклик і складає протокол вручну, але це в минулому. З додатком який розробляється спеціально для таких випадків. Стане зручніше і набагато швидше скласти протокол не затримуючи ні себе ні інших учасників.

Завдання, яке має виконувати додаток- це насамперед коректно працювати та обробляти данні які вводяться в поля. Зберігати данні які водяться та мати можливість їх відтворювати.

Немало важливо розробити додаток, щоб він був інтуїтивно зрозумілий інтерфейс для використання. Який буде зручний у користуванні будь-якому користувачеві який захоче користуватись додатком.

Основні можливості додатку який розробляється: функціонал вводу даних в систему, функціонал виводу даних, збереження в базу даних інформацію яка вводиться

Розглянемо можливості цього програмного забезпечення детальніше.

Перше вікно на яке потрапляє - це авторизація користувача. На якому можна обрати делька сценаріїв такі, як ввести логін "номер жетону" та пароль. Якщо ви не зареєстровані то потрібно натиснути на текст реєструватись.

Якщо ви не зареєстровані то програма вас не допустить до роботи і буде вимагати ввести пароль та логін. Якщо ви зареєстровані в додатку. То ви переходите.

На основне вікно в ньому ви можете ввести дату, час, країну, місцезнаходження та інші функції які представлені в програмі.

При умові якщо всі данні заповнені окрім свідків, тому що свідки ДТП не завжди є.

Ви переходите на сторінку заповнення інформації про транспортні засоби двох сторін ситуації. Після того, як поля транспортних засобів будуть заповнені ви можете. Зберегти данні натиснувши кнопку зберегти

Перейти далі до заповнення останнього поля інформації про водіїв. При умові якщо ви заповните поля інформації про водія. При натисканні кнопки завершити програма завершить свою роботу .

Головна особливість додатку – це збереження вивід інформації у файл Microsoft Word. Додаток інтуїтивно зручний та швидкий в роботі. Вся інформація яка вводиться зберігається в базі даних phpMyAdmin.

### **3.2 Моделювання об'єктів проектування**

Раніше люди які мали відношення до програмування. Їм було важко уявити як має виглядати додаток і яка в нього бізнес логіка. Щоб таких проблем не виникало таких проблем винайшли UML діаграми аббревіатура моделювання програмних рішень, структур додатків, розшифровується як Unified Modeling Language. UML - це багата мова поведінки системи та бізнес-процесів.

Існує багато типів UML діаграм такі, як:

Структурні діаграми

- Діаграма компонентів;
- Діаграма профілю;

Складена структурна діаграма

- Поведінкові діаграми;
- Діаграма діяльності;
- Діаграма автомата;
- Діаграма послідовності;

Але в роботі представлено тільки 2 типи UML діаграми про них детальніше.

#### **3.2.1 Діаграма класів**

Перший тип UML діаграми це діаграми класів:

Діаграми класів є основним будівельним елементом будь-якого об'єктно-орієнтованого рішення. Він показує класи в системі, атрибути та операції кожного класу та взаємозв'язок між кожним класом.

У більшості інструментів моделювання клас має три частини. Назва вгорі, атрибути посередині та операції або методи внизу. У великій системі з безліччю пов'язаних класів класи об'єднуються для створення діаграм класів. Різні відносини між класами відображаються різними типами стрілок.

Нижче наведено зображення схеми класів додатку який розробляється рис.3.1.

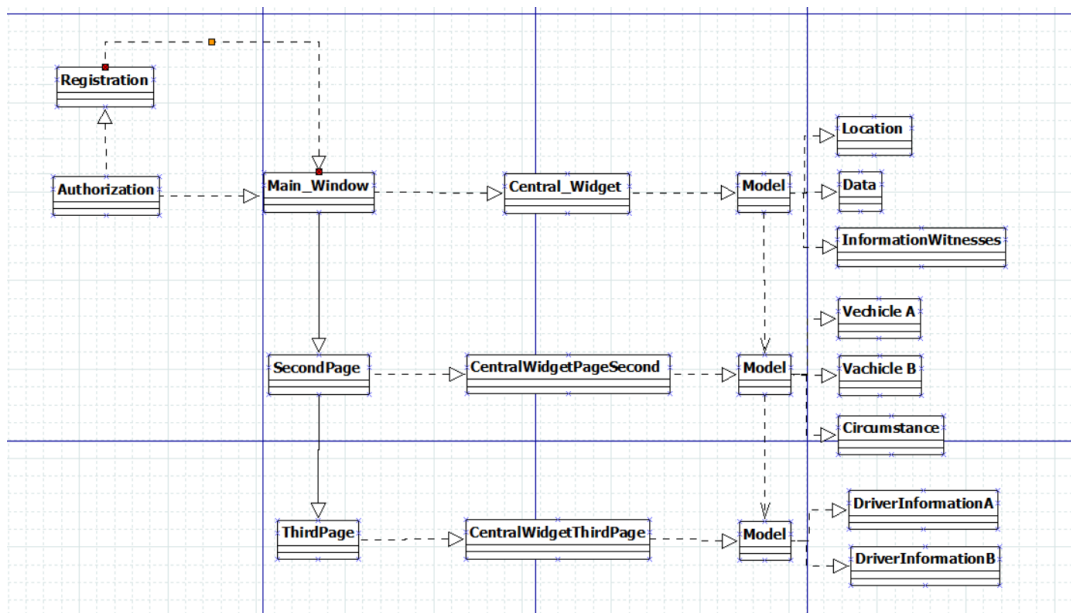


Рисунок 3.1 – UML діаграма класів

### 3.2.2 Діаграма використання

Другий тип UML діаграми це діаграма використання:

Як найбільш відомий тип діаграми поведінкових типів UML, діаграми використання дають графічний огляд суб'єктів, які беруть участь у системі, різні функції, необхідні цим акторам, і те, як ці різні функції взаємодіють.

Це відмінна відправна точка для будь-якого обговорення проекту, оскільки ви можете легко визначити основних учасників та основні процеси в системі. Ви можете створювати схеми випадків використання за допомогою нашого

інструменту або негайно розпочати роботу, використовуючи наші шаблони випадків використання.

Взаємозв'язок діаграми використання показано в прикладі рис.3.2.

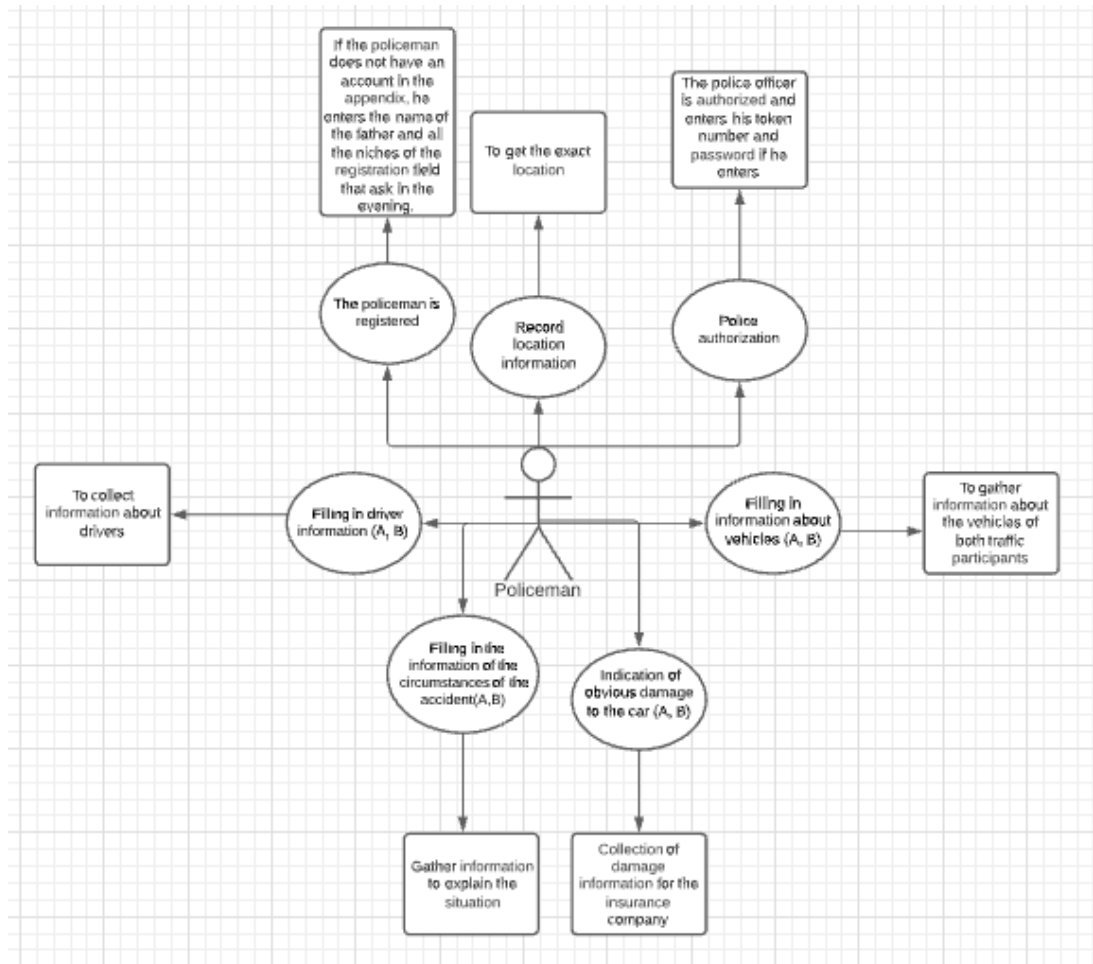


Рисунок 3.2 – UML діаграма використання

### 3.3 Використання бази даних phpMyAdmin

На сьогоднішній день ми всі не уявляємо життя без бази даних. База даних вона практично всюди наприклад, в таких галузях, як медицина, правоохоронні органи, освіта, економічній та багато інших важливих галузях. Існує велика кількість видів баз даних, а саме

- Cosmos DB;
- ArangoDB;

- Couchbase Server;

Але у вищезгаданих баз даних не підтримує MySQL. Так що ж таке MySQL . MySQL - це система управління реляційними базами даних з відкритим кодом (СУБД) із моделлю клієнт-сервер. Було прийнято використовувати для збереження базу даних phpMyAdmin. Це хороший вибір використовувати для збереження саме цю базу даних. phpMyAdmin-це інструмент для адміністрування даних написаний на мові програмування PHP.

Розглянемо шлях збереження бази даних phpMyAdmin. На сторінці авторизації в застосунку. (LoginForm) обробляється кнопка buttonLogin\_Click. Клас MySqlConnection представляє набір команд та даних підключення до бази даних використовується для заповнення набору даних та оновлення бази даних в MySQL. DataTable представляє таблицю в якій зберігаються данні які вводяться. INSERT FROM- цей оператор бере данні із таблиці і виводить їх. В програмі яка розробляється існує ще один оператор який виконано кардинально іншу функцію і він називається INSERT INTO– цей оператор означає додавання даних в таблицю зображених на рис 3.4. Всі перераховані класи та методи зображені на рис. 3.3.

```

command:
private void buttonLogin_Click(object sender, EventArgs e)
{
    if(loginField.Text==" ")
    {
        MessageBox.Show("Введіть логін");
        return;
    }
    if(passField.Text=="")
    {
        MessageBox.Show("Введіть пароль");
        return;
    }
    Form1 frm2 = new Form1();
    frm2.Show();
    this.Hide();
    String loginUser = loginField.Text;
    String passUser = passField.Text;

    DB db = new DB();
    DataTable table = new DataTable();

    MySqlConnection adapter = new MySqlConnection();

    MySqlCommand command = new MySqlCommand("SELECT * FROM `users` WHERE `login` = @u1 AND `password` = @p ",db.getConnection());
    command.Parameters.Add("@u1", MySqlDbType.VarChar).Value=loginUser;
    command.Parameters.Add("@p", MySqlDbType.VarChar).Value = passUser;

    adapter.SelectCommand = command;
    adapter.Fill(table);
}

```

Рисунок 3.3 – Підключення та запис полів в базу даних phpMyAdmin

```

DB db = new DB();
Datatable table = new Datatable();

SqlCommand command = new MySqlCommand("INSERT INTO 'vehicle_a' ('surname', 'name', 'address', 'zip_code', 'country', 'phone_or_email', 'br
name', 'brand_type', 'state_registration_number_plate', 'country_of_registr

command.Parameters.Add("@surname", SqlDbType.VarChar).Value = surname_vehicle_a.Text;
command.Parameters.Add("@name", SqlDbType.VarChar).Value = name_vehicle_a.Text;
command.Parameters.Add("@address", SqlDbType.VarChar).Value = address_vehicle_a.Text;
command.Parameters.Add("@zip_code", SqlDbType.VarChar).Value = zip_code_vehicle_a.Text;
command.Parameters.Add("@country", SqlDbType.VarChar).Value = country_vehicle_a.Text;
command.Parameters.Add("@phone_or_email", SqlDbType.VarChar).Value = phone_or_email_vehicle_a.Text;
command.Parameters.Add("@brand_type", SqlDbType.VarChar).Value = brand_type_vehicle_a.Text;
command.Parameters.Add("@state_registration_number_plate", SqlDbType.VarChar).Value = state_registration_number_plate_vehicle_a.Text;
command.Parameters.Add("@country_of_registration_1", SqlDbType.VarChar).Value = country_of_registration_1_vehicle_a.Text;
command.Parameters.Add("@state_registration_number", SqlDbType.VarChar).Value = state_registration_number_vehicle_a.Text;
command.Parameters.Add("@country_of_registration_2", SqlDbType.VarChar).Value = country_of_registration_2_vehicle_a.Text;
command.Parameters.Add("@name_policy", SqlDbType.VarChar).Value = name_policy_vehicle_a.Text;
command.Parameters.Add("@number_policy", SqlDbType.VarChar).Value = number_policy_vehicle_a.Text;
command.Parameters.Add("@number_green_card", SqlDbType.VarChar).Value = number_green_card_vehicle_a.Text;
command.Parameters.Add("@certificate_is_valid_from", SqlDbType.VarChar).Value = certificate_is_valid_from_vehicle_a.Text;
command.Parameters.Add("@certificate_is_valid_until", SqlDbType.VarChar).Value = certificate_is_valid_until_vehicle_a.Text;
command.Parameters.Add("@address_agency", SqlDbType.VarChar).Value = address_agency_vehicle_a.Text;
command.Parameters.Add("@phone_agency", SqlDbType.VarChar).Value = phone_agency_vehicle_a.Text;
command.Parameters.Add("@email_agency", SqlDbType.VarChar).Value = email_agency_vehicle_a.Text;
command.Parameters.Add("@circumstance_vehicle_a", SqlDbType.VarChar).Value = circumstance;

db.openConnection();

if (command.ExecuteNonQuery() == 1)
    MessageBox.Show("Інформація збережено");
else
    MessageBox.Show("Інформація не збережено");

db.closeConnection();

```

Рисунок 3.4 – Додавання даних в таблицю vehicle\_a

За допомогою класу DB відбувається з'єднання з базою даних зображено на рис. 3.5.

```

Ссылка: 16
class DB
{
    MySqlConnection connection = new MySqlConnection("server=localhost;port=8889;User=root;password=root;database=itproger");

    Ссылка: 8
    public static object CommandFactory { get; private set; }

    Ссылка: 6
    public void openConnection()
    {
        if (connection.State == System.Data.ConnectionState.Closed);
        connection.Open();
    }

    Ссылка: 6
    public void closeConnection()
    {
        if (connection.State == System.Data.ConnectionState.Open) ;
        connection.Close();
    }

    Ссылка: 8
    public MySqlConnection getConnection()
    {
        return connection;
    }
}

```

Рисунок 3.5 – З'єднання з базою даних phpMyAdmin

Збереження даних відбувається в базі даних phpMyAdmin. База даних де зберігається вся інформація про користувача і даних яких він вводить знаходиться в базі даних itproger рис. 3.6.



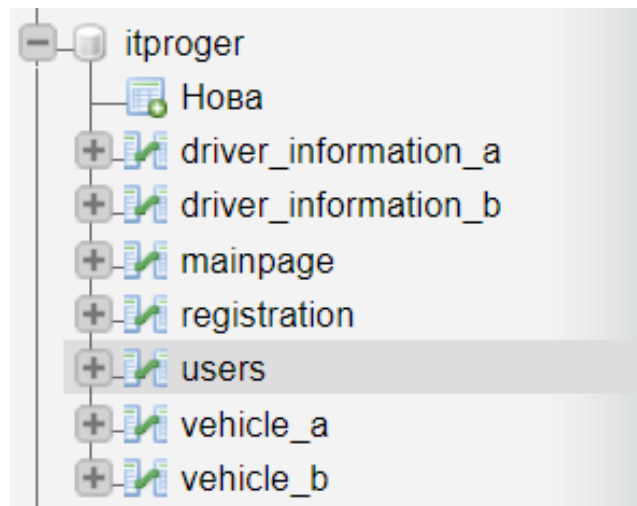


Рисунок 3.6 – Таблиці в базі даних itproger

На попередньому рисунку було зображено багато різних таблиць ось одна із цих таблиць рис. 3.7 це одна із таблиць ( vehicle\_a) таблиця з інформацією про транспортний засіб А.

	id	fieldname	datatype	collation	charset	is_nullable	comment
<input type="checkbox"/>	1	id	int(11)		UNSIGNED	Hi	Немає AUTO_INCREMENT
<input type="checkbox"/>	2	surname	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	3	name	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	4	address	varchar(30)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	5	zip_code	varchar(7)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	6	country	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	7	phone_or_email	varchar(15)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	8	brand_type	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	9	state_registration_number_plate	varchar(30)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	10	country_of_registration_1	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	11	state_registration_number	varchar(30)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	12	country_of_registration_2	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	13	name_policy	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	14	number_policy	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	15	number_green_card	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	16	certificate_is_valid_from	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	17	certificate_is_valid_until	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	18	name_agency	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	19	address_agency	varchar(20)	utf8_general_ci		Hi	Немає
<input type="checkbox"/>	20	country_agency	varchar(20)	utf8_general_ci		Hi	Немає

Рисунок 3.7 – Структура таблиці vehicle\_a

### 3.4 Запуск застосунку

Застосунок запускається з файлу типу exe. Для запуску додатку потрібно відкрити exe файл на якому знаходиться застосунок. Відкриється додаток на першій сторінці – це авторизація. рис 3.8.

Рисунок 3.8 – Активне вікно авторизації

### 3.5 Тестування застосунку

Застосунок має працювати чітко без збоїв забезпечувати належну якість продукту. Для тестування застосунку було створено набір методів.

Тест-кейс- документ який має в собі набір кроків для перевірки одного очікуваного результату.

Таблиця 3.1 – Тест-кейс для тестування функціоналу авторизації

Функціонал авторизації користувача		
Передумова: Відкрити сторінку авторизації в застосунку		
Крок тесту	Дані тесту	Очікуваний результат
Ввід даних в поле логіна логін	<ul style="list-style-type: none"> <li>- Ввести логін “Номер жетону”</li> <li>- Ввести пароль</li> <li>- Натиснути текст “У вас не має акаунта”</li> <li>- Натиснути кнопку “Увійти”</li> </ul>	<ul style="list-style-type: none"> <li>- Користувач увійшов в систему</li> <li>- Відкривається головна сторінка застосунку</li> </ul>

Таблиця 3.2 – Тест-кейси для тестування функціоналу реєстрації

Функціонал реєстрації користувача		
Передумова: Відкрити з сторінку реєстрації при натисканні на текст “ У вас не має акаунта ”		
Крок тесту	Дані тесту	Очікуваний результат
Реєстрація з коректними значеннями	<ul style="list-style-type: none"> <li>- Ввести ім'я користувача</li> <li>- Ввести прізвище користувача</li> <li>- Ввести логін “Номер жетону”</li> <li>- Ввести звання</li> <li>- Натиснути кнопку «Зареєструватися»</li> </ul>	<ul style="list-style-type: none"> <li>- Користувача зареєстровано в системі</li> <li>- Відкривається головна сторінка застосунку</li> </ul>

Таблиця 3.3 – Тест-кейси для тестування функціоналу основного

Функціонал сторінки користувача		
Передумова: Щоб перейти на головну сторінку додатку потрібно натиснути кнопку “Зареєструватись” або за умови якщо є акаунт в базі даних. видється правильний пароль і логін .		
Крок тесту	Дані тесту	Очікуваний результат
Заповнення полів які потребують обов'язкового заповнення	<ul style="list-style-type: none"> <li>- Дата та час записуються автоматично в систему</li> <li>- Ввести прізвище користувача</li> <li>- Ввести логін “Номер жетону”</li> <li>- Ввести звання</li> </ul>	<ul style="list-style-type: none"> <li>- Повинна зберегтися вся інформація, в системі яка вводилась користувачем.</li> <li>- Відкривається головна сторінка застосунку</li> <li>- При натисканні кнопки</li> </ul>

Продовження таблиці 3.3 – Тест-кейси для тестування функціоналу основного

Функціонал сторінки користувача		
Передумова: Щоб перейти на головну сторінку додатку потрібно натиснути кнопку “Зареєструватися” або за умови якщо є акаунт в базі даних. видється правильний пароль і логін .		
Крок тесту	Дані тесту	Очікуваний результат
	- Натиснути кнопку “Зареєструватися”	“Зареєструватися” повинно перейти на наступну сторінку заповнення транспортних засобів А, В

Таблиця 3.4 – Тест-кейси для тестування функціоналу таблиць з заповненням транспортних засобів

Функціонал сторінки користувача		
Передумова: Щоб перейти на головну сторінку додатку потрібно натиснути кнопку “Зареєструватися” або за умови якщо є акаунт в базі даних. видється правильний пароль і логін .		
Крок тесту	Дані тесту	Очікуваний результат
Заповнення полів які потребують обов’язкового заповнення	<ul style="list-style-type: none"> <li>- Дата та час записуються автоматично в систему</li> <li>- Ввести прізвище користувача</li> <li>- Ввести логін “Номер жетону”</li> <li>- Ввести звання</li> <li>- Натиснути кнопку “Зареєструватися”</li> </ul>	<ul style="list-style-type: none"> <li>- Повинна зберегтися вся інформація, в системі яка вводилась користувачем.</li> <li>- Відкривається головна сторінка застосунку</li> <li>- При натисканні кнопки “Зареєструватися” повинно перейти на наступну сторінку заповнення транспортних засобів А, В</li> </ul>

Таблиця 3.5 – Тест-кейси для тестування функціоналу таблиць з заповненням транспортних засобів

Функціонал сторінки користувача		
Передумова: Щоб перейти на сторінку заповнення інформації про водія додатку потрібно натиснути кнопку “Перейти далі”		
Крок тесту	Дані тесту	Очікуваний результат
Заповнення полів які потребують обов’язкового заповнення	<ul style="list-style-type: none"> <li>- Заповнення даних водія транспортний засіб А;</li> <li>- Зберегти введену інформацію транспортного засобу А ;</li> <li>- Заповнення даних водія про транспортний засіб В;</li> <li>- Зберегти введену інформацію транспортного засобу А ;</li> <li>- Вибір обставин дорожньо-транспортної пригоди;</li> <li>- Натиснути кнопку «Завершити заповнення »;</li> </ul>	<ul style="list-style-type: none"> <li>- Заповнюватись поля які потребують заповненню;</li> <li>- При натисканні кнопки “Завершити заповнення” система має без помилок;</li> <li>- завершити заповнення;</li> <li>- Зберегтися вся інформація, в базі даних яка вводилась користувачем;</li> </ul>

## 4 ІНСТРУКЦІЯ КОРИСТУВАЧА

Для того щоб запустити застосунок потрібно мати на комп'ютері якому буде запускається з файлу типу exe.

Для того щоб зареєструватись або зайти в існуючий профіль. Натиснувши кнопку «Зареєструватись» відбувається реєстрація користувача зображена на рис. 4.1. При натисканні кнопки «Увійти» відкривається сторінка авторизація рис. 4.1.

The image shows two side-by-side windows. The left window is titled 'Авторизація' (Authorization) and contains two input fields: 'Введіть номер жетону' (Enter token number) and 'Введіть пароль' (Enter password). Below the password field is a blue button labeled 'УВІЙТИ' (Log In) and a link 'У вас немає аккаунта?' (Don't have an account?). The right window is titled 'Реєстрація' (Registration) and contains five input fields: 'Введіть ім'я' (Enter name), 'Введіть прізвище' (Enter surname), 'Введіть номер жетону' (Enter token number), 'Введіть звання' (Enter rank), and 'Введіть пароль' (Enter password). Below the password field is a blue button labeled 'ЗАРЕЄСТРУВАТИСЬ' (Register) and a link 'Авторизуватись' (Log In).

Рисунок 4.1 – Авторизація та реєстрація користувача

Після реєстрації при умові якщо користувач правильно вписав всі поля які вказані. Користувач переходить на головну сторінку застосунку. Де виведено назва країни, місце знаходження показано на рис. 4.2.

Рисунок 4.2 – Вікно для заповнення інформації

Після того як користувач заповнить вказані поля інформацією при натисканні кнопки «Перейти далі» застосунок переходить на наступне вікно для заповнення інформацією про технічний засіб (А,В) та обставини що призвели до дорожньо-транспортної пригоди зображено на рис. 4.3.

Рисунок 4.3 – Вікно для заповнення інформації про транспортні засоби та обставин ДТП

При умові якщо потрібні поля для складання протоколу про дорожньо-транспортну пригоду будуть заповнені та при натисканні на кнопки “Зберегти” інформацію буде збережено. Натиснувши кнопку «Перейти далі» система відкриє наступне вікно для заповнення даними інформацію про водія (А,В) показується на рис. 4.4.

Рисунок 4.4 – Вікно для заповнення інформації про водія

Після того як користувач заповнить всю інформацію про водія (А,В) при натисканні кнопки «Зберегти» з обох сторін вікна. Всі заповнені поля збережуться. Для виходу із застосунку потрібно натиснути на кнопку «Завершити заповнення» при натисканні вказаної вище кнопки користувач вийде з системи.



## ВИСНОВКИ

Робота присвячена розробці складання протоколу про дорожньо-транспортну пригоду.

Проведено дослідження, в якому було обґрунтовано актуальність та наукову новизну роботи.

Досліджено причини виникнення які можуть призвести до дорожньо-транспортної пригоди та структуру технічної складової автомобіля, яка може спричинити дорожньо-транспортну пригоду.

Враховуючи переваги та недоліки існуючих застосунків було складено алгоритм для покращення системи складання протоколу про дорожньо-транспортну пригоду. Було визначено вимоги та кількість розділів та функціональні можливості.

Описано програмні засоби які було застосовано для розробки програмного забезпечення. Визначено що оптимальним рішення є використання середовища розробки Visual Studio та мови програмування C#.

Якість та працездатність додатку забезпечують мануальні тести. Створені тестові випадки для перевірки основного функціоналу.

Застосунок в основному буде використовуватись в органах правопорядку для допомоги складання електронного протоколу.

На основі результатів виконаних досліджень розроблено систему для складання протоколів про Дорожньо-транспортну пригоду для поліцейських при виконанні своїх обов'язків.

В роботі виконано аналіз існуючих застосунків для операційної системи Windows;

В недалекому майбутньому планується розробити додаткові функції застосунку такі як вікно з недавніми заповненими даними та можливістю їх редагування, зміни та видалення з системи та бази даних. В подальшому планується розробити новий дизайн застосунку та переробити структуру додатку.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey. // National highway traffic safety administration. – 2015. – С. 2.
2. Troelsen A. C# 6.0 and the .NET 4.6 Framework / A. Troelsen, P. Japikse., 2015. – 1433 с. – (Williams).
3. Lippert E. C# in Depth, Fourth Edition / Eric Lippert., 2019. – 528 с.
4. L. Thai T. .NET framework essentials / Thuan L. Thai., 2001. – 320 с.
5. Delaille M. PhpmysqladminStarter / Mark Delaille., 2012. – 42 с. – (Packt Publishing).
6. Delisle M. Creating Your MySQL Database: Practical Design Tips and Techniques / Marc Delisle., 2006. – 139 с.
7. Dyer R. MySQL in a Nutshell / Russell Dyer., 2008. – 566 с. – (O'Reilly Media).
8. Car Accidents and their prevention [Електронний ресурс] – Режим доступу до ресурсу: [https://www.huffpost.com/entry/top-15-causes-of-car-accidents\\_b\\_11722196](https://www.huffpost.com/entry/top-15-causes-of-car-accidents_b_11722196).
9. Causes of Accidents - Human Factors [Електронний ресурс] – Режим доступу до ресурсу: <https://www.newhampshireinjurylawfirm.com/causes-of-accidents-human-factors.html>.
10. Accidents Caused by Poor Road Conditions [Електронний ресурс] – Режим доступу до ресурсу: <https://www.injuryclaimcoach.com/road-conditions.html#:~:text=Common%20road%20conditions%20that%20contribute,Mis sing%20or%20inadequate%20guardrails>.
11. Car Accidents and Bad Road Conditions [Електронний ресурс] – Режим доступу до ресурсу: <https://www.edgarsnyder.com/car-accident/cause-of-accident/bad-roads/>.
12. Accidents Caused by poor road [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gilmanbedigian.com/accidents-caused-by-poor-road-quality-and-conditions-in-dc/>.
13. Animals on the roadway how to prevent a car accident [Електронний ресурс] – Режим доступу до ресурсу: <https://www.havahart.com/wildlife-on-the-road>

14. Risk drivers pose to themselves and other drivers by violating traffic rules [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.tandfonline.com/doi/abs/10.1080/15389588.2016.1177637?journalCode=gspi20&>.
15. What is C# used for? [Электронный ресурс] – Режим доступа до ресурсу:  
<https://stackify.com/what-is-c-used-for/>.
16. Development Environment [Электронный ресурс] – Режим доступа до ресурсу: <https://www.techopedia.com/definition/16376/development-environment#:~:text=A%20development%20environment%20is%20a,referred%20to%20as%20the%20DSP.>
17. Information and general information about the Visual Studio [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.microsoft.com/uk-ua/visualstudio/get-started/visual-studio-ide?view=vs-2019>.
18. What is .NET Framework? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/dotnet/framework/get-started/>.
19. Work and structure of Windows forms [Электронный ресурс] – Режим доступа до ресурсу: <https://www.techopedia.com/definition/24300/windows-forms-net>.
20. All types and kinds of UML diagrams [Электронный ресурс] – Режим доступа до ресурсу: <https://creately.com/blog/diagrams/uml-diagram-types-examples/>.
21. Types of Database Management Systems [Электронный ресурс] – Режим доступа до ресурсу: <https://www.c-sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/>.
22. What is MySQL? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.hostinger.com/tutorials/what-is-mysql>.
23. What are the main functions MySQL? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mysqltutorial.org/what-is-mysql/>.
24. phpMyAdmin Database Management Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.siteground.com/tutorials/phpmyadmin/database-management/>.

# ДОДАТОК



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра Інженерії програмного забезпечення



РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СКЛАДАННЯ ПРОТОКОЛУ ПРО  
ДОРОЖНЬО-ТРАНСПОРТНУ ПРИГОДУ НА МОВІ C#

Виконав:  
Маринскас Вадим Миколайович  
Науковий керівник:  
Негоденко Олена Василівна

Київ 2021

## Мета, об'єкт, предмет дослідження

*Мета роботи* – засоби розробки додатку для складання електронного протоколу про дорожньо-транспортну пригоду.

*Об'єкт дослідження* – підвищення ефективності заповнення електронного протоколу при дорожньо-транспортній пригоді.

*Предмет дослідження* – основні принципи функціонування електронного протоколу про дорожньо-транспортну пригоду.

## Актуальність

*Актуальність моєї дослідницької роботи* полягає у тому, що в поліцейських виникає з кожним днем все більше роботи, для зменшення часу складання протоколу, тому було прийнято рішення розробити додаток для допомоги працівникам правоохоронних органів складати електронний протокол про дорожньо-транспортну пригоду.

## Вибір мови програмування

Дана мова використовує об'єктно-орієнтований підхід до програмування у всьому. Це означає, що тобі потрібно буде описувати абстрактні конструкції на основі предметної області, а потім реалізовувати між ними взаємодія.

Програмування на C # - одна з найпопулярніших мов програмування у світі розробки програмного забезпечення.

## Середовище розробки

Середовище Visual Studio дозволяє розробляти додатки, використовуючи різні мови програмування: Visual C#, Visual Basic, Visual F#, Visual C++, Python . Також існує можливість розробляти додатки не тільки під Windows, а і під інші популярні платформи: Android, iOS.

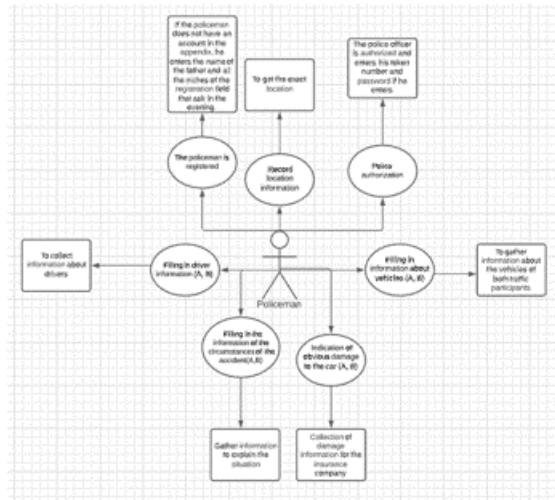
Версія Visual Studio Community є абсолютно безкоштовною для учнів, студентів та розробників програм з відкритим програмним кодом.



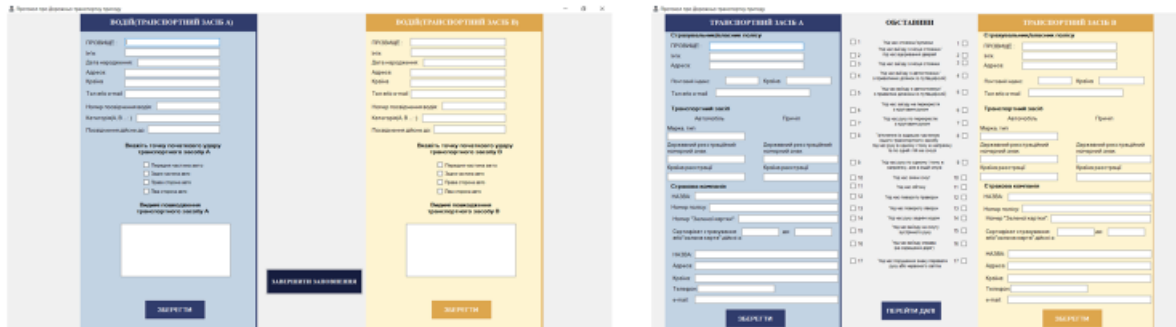
## Аналоги

Назва застосунку	Переваги	Недоліки
Електронний европротокол	<ul style="list-style-type: none"><li>- Багатий функціонал</li><li>- Простота у використанні</li><li>- Зберігання безпечно інформації в базі даних</li><li>- Можливість зробити фотозвіт</li><li>- Геолокація місця ДТП</li><li>- Прораховані всі варіанти які можуть призвести до аварії</li><li>- Перевірка введених даних</li><li>- Можливість відправлення на електронну пошту обом учасникам ДТП</li></ul>	Потребує підключення до мережі інтернет
Мій поліс	<ul style="list-style-type: none"><li>- Зручний в користуванні</li><li>- Можливість сканування QR- кодом</li><li>- Можливість складання електронного европротоколу</li><li>- Швидкість роботи додатку</li></ul>	Спливаюче вікно не несе ніякої потрібної інформації

# Моделювання об'єктів проектування



# Програмні засоби реалізації



# Апробація результатів дослідження

Lippert E. C# in Depth, Fourth Edition / Eric Lippert., 2019. – 528 с.

Information and general information about the Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/uk-ua/visualstudio/get-started/visual-studio-ide?view=vs-2019>.

## Висновок

На основі результатів виконаних досліджень розроблено систему для складання протоколів про Дорожньо-транспортну пригоду для поліцейських при виконанні своїх обов'язків.

В роботі виконано аналіз існуючих застосунків для операційної системи Windows;

В недалекому майбутньому планується розробити додаткові функції застосунку такі як вікно з недавніми заповненими даними та можливістю їх редагування, зміни та видалення з системи та бази даних. В подальшому планується розробити новий дизайн застосунку та переробити структуру додатку.



**ДЯКУЮ ЗА УВАГУ**