НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Механіко – технологічний факультет

УДК 629.33:681.518.54:004

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
технічного сервісу та інженерного
(назва кафедри)
менеджменту імені М.П. Момотенка

_____ **Іван РОГОВСЬКИЙ**
(підпис) **(ПІБ)**

«____» _____ **2024 р.**

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «Удосконалення методу OBD-2 комп'ютерної діагностики автомобіля»

Спеціальність 274 «Автомобільний транспорт»
(код і назва)

Освітня програма «Автомобільний транспорт»
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна, або освітньо-наукова)

Гарант освітньої програми
доктор технічних наук, професор _____ Войтюк Валерій Дмитрович
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи
к.т.н., доцент _____ Іщенко Валерій Васильович
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконала _____ Кулібаба Олександра Вікторівна
(підпис) (ПІБ)

КИЇВ – 2024

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Механіко – технологічний факультет

ЗАТВЕРДЖУЮ
Завідувач кафедри технічного сервісу та інженерного менеджменту імені М.П. Момотенка

д.т.н., проф. _____ Іван РОГОВСЬКИЙ
(науковий ступінь, вчене звання) (підпис) (ПІБ)

«____» _____ 2024 р.

З А В Д А Н Н Я
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТЦІ
Кулібабі Олександрі Вікторівни
(прізвище, ім'я, по батькові)

Спеціальність 274 «Автомобільний транспорт»
(код і назва)

Освітня програма «Автомобільний транспорт»
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна, або освітньо-наукова)

Тема магістерської кваліфікаційної роботи «Удосконалення методу OBD-2 комп'ютерної діагностики автомобіля»
затверджена наказом ректора НУБіП України від «07» грудня 2023 р. № 2224 «С»

Термін подання завершеної роботи на кафедру
(рік, місяць, число)

Вихідні дані до роботи:
1. Науково – технічна література; результати науково-дослідних робіт по літературних джерелах методу OBD-2 комп'ютерної діагностики автомобіля

Перелік питань, що підлягають дослідженню:
1. Аналіз предметної галузі
2. Комп'ютерна діагностика автомобіля OBD-II
3. Обґрунтування методів досліджень та послідовність етапів проведення наукових досліджень
4. Дослідження методів коп'ютерної діагностики автомобіля за технологією OBD2
5. Аналіз результатів дослідження
6. Розробка архітектури системи

Перелік графічного матеріалу Електронна презентація на 12 слайдах

Дата видачі завдання «10» листопада 2023 р.

Керівник магістерської кваліфікаційної роботи _____ Іщенко В.В.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Кулібаба О.В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота магістра містить: 76 с., 10 рис., 20 джерел. АВТОМОБІЛЬ, БОРТОВИЙ КОМП'ЮТЕР, КВАЛІФІКАЦІЙНА РОБОТА, КОМП'ЮТЕРНА ДІАГНОСТИКА, ПОМИЛКИ, ПРОБІГ, ASP.NET CORE, OBD2,VISUAL STUDIO, WEB–САЙТ.

Об'єктом дослідження є технологія OBD2 для комп'ютерної діагностики автомобіля.

Метою роботи є вивчення можливостей даного стандарту діагностування автомобіля, його можливостей для розробки програмної системи, що є аналогом бортового комп'ютера авто.

У результаті роботи здійснено дослідження технології OBD2 для діагностування автомобіля, виконано проектування системи для автомобілістів, що дозволяє діагностувати автомобіль за наявності спеціального сканеру.

CAR, ON-BOARD COMPUTER, QUALIFICATION WORK, COMPUTER DIAGNOSTICS, ERRORS, MILEAGE, ASP.NET CORE, OBD2, VISUAL STUDIO, WEB SITE.

The object of research is OBD2 technology for computer diagnostics of the car.

The purpose of the work is to study the possibility of this standard of diagnosing the car, its capabilities for the development of a software system that is analogous to theon-board computer of the car.

As a result of work the research of OBD2 technology for diagnosing the car is carried out, design of system for motorists which allows to diagnose the car in the presence of the special scanner is executed.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення.

ALDL – протокол діагностичного зв’язку збірки.

API (Application Programming Interface) – опис методів, за допомогою яких одна програма може взаємодіяти з іншою.

ARC (Automatic Reference Counting) – система керування пам’яттю в мові програмування Swift.

BLE (Bluetooth Low Energy) – технологія Bluetooth з наднизьким енер госпоживанням.

BR/EDR (Bluetooth Basic Rate/ Enhanced Data Rate) – класична техно логія Bluetooth.

Core Data – фреймворк від компанії Apple, вбудований в операційну систему iOS, MacOS, який дозволяє розробнику взаємодіяти з базою даних.

CSV (Comma-Separated Values) – текстовий формат, який використо вується для представлення табличних даних.

DIP (Dependency inversion principle) – принцип інверсії залежностей. DTC (Diagnostic Trouble Codes) – діагностичні коди помилок.

ECM – протокол для тестування модуля керування двигуном.

EOBD (European On Board Diagnostic) – Європейська бортова діагностична система.

HTTP (HyperText Transfer Protocol) – протокол передачі даних, що використовується в комп’ютерних мережах.

ISP (Interface segregation principle) – принцип розділення інтерфейсу. LE (Low Energy) – технологія Bluetooth з наднизьким енергоспоживанням.

LSP (Liskov substitution principle) – принцип підстановки Лісков.

MVC (Modal-View-Controller)– архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

MVP (Modal-View-Presenter) – архітектурний шаблон, похідний від MVC, що відділяє візуальне відображення та поведінку обробки подій у різні

класи.

MVVM (Model-View-View Model) – архітектурний шаблон, що зв’язує елементи інтерфейсу зі змінними, які їх описують.

OBD (On-Board Diagnostics) – загальна назва передбачених виробни- ком систем діагностики транспортних засобів.

OBDII – стандарт бортової діагностики, розроблений в середини 90-х років.

OCP (Open/closed principle) – принцип відкритості/закритості.

PID (Parameter Identification) – ідентифікатор параметра, що використовується для запиту діагностичної інформації.

Realm – система управління об’єктної базою даних з відкритим вихідним кодом, розрахована для мобільних пристроїв.

REST API (Representational State Transfer) – підхід до архітектури мережевих протоколів, які забезпечують доступ до інформаційних ресурсів.

SAE (Society of Automotive Engineers) – товаривство автотраспортних інженерів.

SOLID (single responsibility, open-closed, Liskov substitution, interface segregation и dependency inversion) – абревіатура складена з перших літер п’яти базових принципів об’єктно-орієнтованого програмування та дизайну запропонована Робертом Мартіном

SRP (Single responsibility principle) – принцип єдиного обов’язку. UART (Universal Asynchronous Receiver-Transmitter) – тип асинхронного приймача-передавача, компонентів комп’ютерів та периферійних пристроїв, що передає дані між паралельною та послідовною формами.

VIPER (View-Interactor-Presenter-Entity-Router) – архітектурний шаблон, що поділяє задачі на відображення змін, обробку подій та маршрутизацію.

VIP (View-Interactor-Presenter) – архітектурний шаблон, що створює коло викликів від представлення до інтерактора та від інтерактора до презентера, який викликає оновлення у представлені.

ВСТУП

Не дивлячись на швидкі темпи розвитку автомобільної індустрії, сьогодні велика кількість автомобілів є достатньо примітивними і навіть не мають бортового комп'ютера або мають, але він наділений дуже вузьким функціоналом. Але в той самий час майже всі сучасні автомобілі обладнані діагностичними портами, які дозволяють отримати безліч корисної інформації про авто.

Технологій комп'ютерної діагностики автомобіля OBD2 – це стандарт діагностики різних систем автомобіля, що виконується блоками управління автомобілем. Він був введений у 1996 році як обов'язковий для усіх автомобілів, що продаються в Сполучений Штатах Америки, а згодом і в усіх інших автомобілях.

Зараз кожен більш менш автомобіль обладнаний даним діагностичним портом, а кількість тих, що необладнана настільки низька – що її можна списати напохибку.

Тож з великою впевненістю можна сказати, що дана технологія є перспективною для дослідження.

На сьогоднішній день існує велика кількість подібних програм, шо дозволяють автомобілістам самостійно діагностувати власну машину. Але вони мають ряд певних недоліків. Усі ці програми можна розділити на 2 види: професійне програмне забезпечення та те, яке призначене для широкого використання.

Професійне ПЗ звичайно має широкий функціонал та надає велику кількістьможливостей користувачу. Але це програмне забезпечення по-перше вимагає професійного обладнання, і по-друге має велику вартість. Ці фактори перекреслюють будь-який сенс використання цього ПЗ широкими масами.

Інша категорія програм – це ті, які призначені для широкого використання. Вони в свою чергу мають невелику вартість, але переважна більшість має вузький функціонал, недоліки у своїй роботі або просто

незручний та не інтуїтивний інтерфейс.

Це наштовхнуло мене на ідею дослідження технологій комп’ютерної діагностики автомобіля для створення системи, що дозволить будь-якому автомобілісту власноруч отримувати інформацію про свою машину, яка раніше була доступна тільки на спеціалізованих станціях.

Метою дослідження – є вивчення можливостей даного стандарту діагностування автомобіля для подальшого застосування у розробці програмної системи, що є асистентом водія.

Сучасні машини – це дуже складні технологічні механізми, які наповнені нескінченною кількість різноманітних комп’ютерних систем, блоками керування певними вузлами. Тож сьогодні діагностика автомобіля – це складний процес. Адже із плином технічного прогресу автомобілі дуже стрімко розвиваються та стають більш комп’ютеризованими. Тому у разі якоїсь поломки майже неможливо її діагностувати без спеціалізованого обладнання та програмного забезпечення. Звичайно велика кількість автомобілів обладнана просунутими бортовими комп’ютерами, які здатні самостійно виявляти та попереджати помилки. Але далеко не всі машини мають подібні функції. А ті що мають не завжди можуть надати всю необхідну інформацію.

Під час виконання роботи було використано наступні методи дослідження:

- метод ідеалізації;
- аналіз;
- наукове дослідження.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Транспортні засоби є незамінною частиною життя кожної людини. Хтось користується громадським транспортом – автобусами, трамваями, метро. Хтось користується послугами таксі. Хтось придбав свій автомобіль та пересувається на ньому. Але те, що автомобілі оточують нас в повсякденному житті – це факт. Наприклад за статистикою у 2019 році в Україні на 1000 жителів нараховувалось 257 автомобілів. Лише в місті Києві офіційно перебуває більше 1000000 автомобілів [1].

Машина надає людині можливість з комфортом пересуватись містом, а також мати повну свободу своїх дій. Адже коли людина володіє власним автотранспортом, вона є незалежною від різноманітних графіків, розкладів. Це даєїй можливість вільно подорожувати містами або країнами.

Сьогодні автомобільна галузь досягла небачених раніше масштабів розвитку.У світі продається величезна кількість машин щороку. Від дешевих та простих до дорогих спортивних авто або авто представницького класу.

Автомобілі розвиваються та стають більш технологічними з року в рік.

Отримують нові функції, оснащення та пропонують більше різних опцій.

Проте дійсно передовими є авто преміум класу, які недоступні для переважної більшості покупців. Основна частина автомобілів які продаються в Україні і світі – це машини середнього та бюджетного сегментів.

1.2 Виявлення проблем та актуалізація рішень

Машинобудування на сьогодні одна з найрозвинутіших галузей. Це обумовлює наявність різних проблем [2].

Це екологічні проблеми. Сьогодні загальновідомим є факт, що навколишнє середовище є забрудненим, спричиненого автомобілями, що використовують ДВЗ.Сучасні викиди CO2 перевищують допустимі норми. Є

альтернативні варіанти - електромобілі. Однак, на перший погляд, це може здатися хорошим способом вирішення проблеми, але сьогодні це не змінить стан справ, оскільки для вироблення електроенергії використовуються способи, що забруднюють навколишнє середовище.

Тому необхідний комплексний підхід для вирішення цієї проблеми. Сюди входить будівництво електростанцій, що використовують альтернативні джерела енергії. Заохочення жителів використовувати більш сучасні транспортні засоби, які викидають в атмосферу менше шкідливих речовин. Стимулювання людей купуватимашини, які використовують електроенергію як паливо. Ця практика дуже поширена в розвинених країнах США та Європи. Там країна платить власнику автомобіля компенсації за придбання електромобілів. Електричні АЗС та інша інфраструктура також будуються, щоб полегшити користування електромобілями щодня.

Не варто забувати про якість доріг. Україна за цим показником є на одному з найнижчих місць у світі серед розвинених країн. Нерівні дороги становлять загрозуне тільки для технічного стану машини, але і для безпеки руху. Невеликий об’їзд навколо ями може спричинити аварійну ситуацію. Тому ремонт доріг є одним з головних пріоритетів, що впливає на безпеку дорожнього руху [2].

Загалом питання безпеки дорожнього руху дуже важливі. Великі компанії витрачають багато грошей щороку те щоб зробити машини безпечнішими, таких як гальмівні помічники, різні системи виявлення пішоходів, системи допомогиводієві та системи нічного бачення, які можуть виявляти перехожих і тварин утемряві. Вони навіть можуть уникати аварійні ситуації самостійно. Такі системи поступово з’являються у більш-менш недорогих автомобілях, але все ще недоступні для більшості водіїв. Цю проблему потрібно вирішити якомога швидше. Але проблема тут не лише в машинах та водіях, а й у дорогах, їх якості та недоліках їх конструкції. В Україні недостатньо уваги приділяється проектуванню доріг. Наприклад, за аналогією з багатьма розвиненими країнами можна запровадити більше

перехресть з круговим рухом, що значно зменшує шанс дорожньо-транспортних пригод. На пішохідних переходах доцільно будувати спеціальні «острівці безпеки», відокремлені бордюрами та спеціальними бамперами. Це може суттєво зменшити кількість ДТП за участю пішоходів.

Крім того, оскільки це одна з найбільших пасток як для пішоходів, так і для водіїв, це нерегульовані пішохідні переходи на багатосмугових дорогах. Автомобіль зупиняється перед таким пішохідним переходом, водій в іншій смузі не помічає це, пішохід минає перший автомобіль. У такій ситуації дуже складно попередити дорожньо-транспортну пригоду.

Однією з найпоширеніших проблем, з якими сьогодні стикаються водії, є проблема викрадення машин [2]. Стандартні засоби захисту автомобілів не є надійним захистом. На чорному ринку є багато нелегальних засобів, які дозволяють злодіям зламати стандартні запобіжні заходи та заволодіти машиною. Транспортнізасоби, обладнані системою входу без ключа, особливо вразливі до цієї загрози. Злодій, який має спеціальний пристрій для передачі сигналу від ключа до машини,може легко вкрасти її.

Тому водіям слід пам’ятати, що їм потрібно обладнати додаткові засоби, а не сподіватися на штатні системи захисту автомобіля. Наприклад, спеціальний GPS- сигналізатор, який може відстежувати автомобіль, і додатковий замок на капоті автомобіля тощо.

Потрібно мати на увазі, що всі ці автомобільні запобіжні заходи не гарантують, що машина завжди в безпеці і може відстрочити момент викрадення.

Сьогодні багато водіїв не мають сучасних транспортних засобів. Багато транспортних засобів навіть не мають бортового комп’ютера. Це багатофункціональний пристрій, який дозволяє водіям позбутися багатьох проблем. Найпоширенішим прикладом є розрахунок витрати пального на одиницюпробігу. Це показник, з яким кожен власник автомобіля має справу щодня. І без бортового комп’ютера завдання розрахунку споживання палива може доставити багато клопоту.

1.3 Актуальність та мета дослідження

Велика кількість сучасних авто є досить комп’ютеризованими та надаютьширокі можливості у діагностиці та взаємодії з внутрішніми блоками керування. Але це не використовується належним чином. У переважній більшості випадків автомобілісти користуються лише штатними можливостями свого автомобіля та його бортового комп’ютера за наявності. А він далеко не завжди надає всю необхідну для водія інформацію.

Метою дослідження є вивчення можливостей технології OBD-2 та застосування її у програмній системі, що є аналогом бортового комп’ютера автомобіля, вивчення можливостей інтеграції із сервісами, що надають потрібну інформацію для діагностики автомобіля, проектування системи асистента автомобіліста, яка надає функції діагностики автомобіля.

1.4 Постановка задачі

Головними задачами є:

– вивчення методів комп’ютерної діагностики за допомогою технології OBD-2 та застосування її у програмній системі;

– дослідження методів повного сканування та часткового діагностування авто;

– вивчення можливості створення універсальної системи, що дозволить взаємодіяти з будь-яким автомобілем та проводити високорівневу діагностику його систем, а також збирати різного роду інформацію про автомобіль, таку як помилкиу блоках, пробіги, кількість палива тощо;

– визначення архітектури та компонентів з яких буде складатися дана система.

1.5. Опис протоколу OBDII.

Проблема, яка розглядається в даному дипломному проекті – необхідність віддаленої діагностики автомобіля (On-board diagnostics або скорочено – OBD).

Історія розробки таких систем починається з 1968 року, колиVolkswagen впровадив свою першу систему для сканування інжекторних двигунів. Потім бортові комп'ютери починають з'являтися на споживчих транспортних засобах.

В 1980 році General Motors реалізує власний інтерфейс і протокол для тестування модуля керування двигуном (ECM) на лінії складання автомобіля. Протокол діагностичного зв'язку збірки (ALDL) транслюється зі швидкістю 160 біт/с.

В 1986 році з'являється оновлена версія протоколу ALDL, яка передається на рівні 8192 біт/с з напівдуплексною UART-сигналізацією.

В 1988 році Товариство автомобільних інженерів (SAE) рекомендує стандартизований діагностичний роз'єм і набір діагностичних тестових сигналів.

Появляється вимога, щоб всі нові автомобілі мали деякі базові можливості OBD. Ці вимоги, як правило, називають «OBD-I», хоча ця назва не застосовується до введення OBD-II.

З 1996 року специфікація OBD-II стає обов'язковою для всіх автомобілів, вироблених у Сполучених Штатах для продажу в США.

З 2001 року і Європейський Союз робить EOBD обов'язковою для всіх бензинових транспортних засобів, що продаються в Європейському Союзі, а з 2003 року ця вимога розповсюджується й на всі дизельні автомобілі.

З 2008 року специфікація OBD починає вимагатися й в Китаї.

На цей час існує багато засобів для діагностики автомобілів. Кожен із засобів має власні переваги та недоліки, в залежності від яких, формується різноманітна аудиторія користувачів. Але с точки зору звичайного користувача, методика використання будь-якого діагностичного приладу однакова:

Під'єднати апаратний засіб до відповідного порту в автомобілі.

Підключитися до нього за допомогою мережевого сервісу (Bluetooth або Wi-fi).

Використати додаток, який отримує дані з автомобіля та формує результати у зручній та зрозумілій для користувача формі (мається на увазі переклад байтів команд у декодований текст та визначення одиниць вимірювання).

В більшості ситуацій людина використовує діагностичне обладнання, щоб власноруч визначити причину несправності у роботі автомобіля. Але кваліфікація власника найчастіше не дозволяє робити висновки щодо стану тих чи інших компонентів авто.

1.2 Комп'ютерна діагностика автомобіля OBD-II

OBD-II є поліпшенням в порівнянні з OBD-I з точки зору як можливотей, так і стандартизації. Стандарт OBD-II визначає тип діагностичного роз'єму та його роз'єм, наявні протоколи електричної сигналізації та формат обміну повідомленнями. Він також надає список параметрів транспортного засобу для моніторингу разом з тим, як кодувати дані для кожного.

Стандарт OBD-II містить перелік стандартизованих діагностичних кодів. В результаті цієї стандартизації, один пристрій може використовувати бортовий комп'ютер для цих параметрів у будь-якому транспортному засобі. Стандартизація OBD-II спростила діагностику обладнання для викидів.

Діагностичні коди OBD-II складаються з 4 цифр, перед якими стоїть буква: P – для двигуна і трансмісії (силова передача), B – для корпусу, C – для шасі і U – для мережі. Виробники можуть також додавати власні параметри даних до їх конкретної реалізації OBD-II, включаючи запити даних в реальному часі, а також коди несправностей.

1.3 Ідентифікатори параметрів бортової діагностики

Ідентифікатори параметрів бортової діагностики (PIDs OBD-II) – це коди, що використовуються для запиту даних з транспортного засобу.

Стандарт SAE визначає багато кодів OBD-II. Виробники також визначають додаткові PID, які специфічні для їх транспортних засобів. Багато мотоциклів також підтримують OBD-II PID.

Існує 10 діагностичних режимів, описаних в останньому стандартіOBD-II SAE J1979:

01 – показати поточні дані;

02 – показувати дані стоп-кадру;

03 – показати збережені діагностичні коди проблем;

04 – очистити діагностичні коди проблем і збережені значення;

05 – результати випробувань, моніторинг датчиків кисню;

06 – результати випробувань, моніторинг інших компонентів/систем;

07 – показати діагностичні коди неполадок, які очікують на розгляд;

08 – керування роботою бортового компонента/системи;

09 – запит інформації про транспортний засіб;

0a – постійні діагностичні коди несправностей (dtcs).

Виробники автомобілів не зобов’язані підтримувати всі режими, але кожен виробник може визначити додаткові режими.

1.4 Аналіз існуючих рішень

На ринку представлено багато видів програмного забезпечення для бортової діагностики автомобіля. Вони дозволяють контролювати різноманітні вузли, так як двигун, трансмісія, паливна система, датчики температурита інші.

Забезпечення доступу до параметрів автомобіля покладається на OBDII-адаптер, який підтримує декілька протоколів передачі даних. Цей протокол визначається в залежності від виробника транспортного засобу та його року випуску.

Звідси, стек діагностичних даних може відрізнятися в залежності від автомобіля, до якого підключається адаптер. При цьому розробник програмного забезпечення не впливає на кількість даних, які отримує адаптер від автомобіля. Тому функціонал існуючих програмних засобів для аналізу відрізняється тим, які дані розробник бере з адаптеру та як їх використовує у своє- му додатку.

Розглянемо найбільш популярні програми для аналізу даних, що

надходять з OBDII-адаптеру. Серед них:

- Car Scanner;
- Вася Диагност;
- ScanMaster-ELM V2.1 RU;
- Carista OBD2;
- EOBD Facile - Car Diagnostic.

Всі вони мають як переваги, так й недоліки, які найчастіше з’являється в наслідок не сумісності з різними марками та модифікаціями автомобілів, методами підключення до автомобілів та передачі даних, ціною, мовною підтримкою та іншими чинниками.

Рис. 1.1. Інтерфейс програми «ВасяДиагност»

Рис. 1.2. CarScaner

Рис. 1.3. Інтерфейс програми ScanMaster-ELM

Рис. 1.4. Інтерфейс програми CARISTA OBD2

Рис. 1.5. Інтерфейс програми EOBD Facile car diagnostic

Порівняння можливостей декількох програмних засобів для діагностики автомобіля представлено в таблиці 1.1.

Таблиця 1.1

ПЗ	Car Scanner	Вася Диагност	ScanMaster-ELM	Carista OBD2	EOBD Facile - Car Diagnostic
Можливості	-графіки з відображенням показників автомобіля, - робота з декількома блоками -читання та скидання помилок, -показники витрат палива, -автоматичне визначення OBD протоколу.	-автопошук помилок, - кодування і програмування блоків управління, - побудова графіків, - видалення кодів помилок. - відомості про блоки управління, - тести виконавців.	- моніторинг параметрів двигуна, - побудова графіків залежностей, - перегляд стоп-кадрів систем, - перегляд і стирання помилок, - відображення статусу паливної системи.	- базова безкоштовна діагностика, -платна поглиблена діагностика, -пошук та скидання помилок.	- видалення кодів помилок (тільки преміальна версія), - відображення коду невизначеності OBD2 конкретних виробників, - відображення в режимі реального часу датчики автомобіля і можлива запису в файл.

Плат-форми	iOS; Android; Windows; Windows Phone	Windows 2000; Windows XP; Windows Server 2003; Windows Vista; Windows Server 2008; Windows 7, 8, 10; Windows Server 2008	Windows 2000; Windows XP; Windows 2003; Windows Vista 7, 8, 8.1	iOS; Android	iOS; Android; Windows Vista 7, 8, 8.1, 10; Mac OS X
Суміс-ність адапте-ра	Wi-Fi, Bluetooth OBD2 ELM327	VAG-COM 409.1; (з ELM 327 адапте-ром не працю є)	ELM327 Bluetooth; ELM327 USB	Bluetooth LE	Wi-Fi; Bluetooth OBD2 ELM327, ELM320, ELM322, ELM323
Розмір	124,8 МБ	6,71 МБ	482,97 КБ	29,7 МБ – iOS; 6,1 МБ – Android	51,5 МБ

Мова інтер-фейсу	російська; англійська; угорська; іспанська; італійська; китайська; корейська; німецька; польська; турецька; французька; чеська.	Російська.	російська; англійська.	англійська; німецька; японська; російська; іспанська.	Більше ніж 24 мови, вклю-чаючи анг-лійську, укра-їнську та ро-сійську.
Вироб-ник	Росія	Росія	ELM Electronics (Канада)	Prizmos Ltd. (Болгарія)	Outils OBD Facile (Фран-ція)
Ціна	5-6 USD		220 USD	20 USD (ада-птер); 40 USD в год	18-36 USD

1.5 Огляд існуючих аналогів

1.5.1 TOAD PRO

TOAD - це скорочення від Total OBD & ECU Auto Diagnostics. Це програмнезабезпечення для діагностики за допомогою технології OBD, яке виконує системи діагностики автомобіля та надає діагностичні звіти. Воно контролює електронні датчики в двигуні, трансмісії та системі викидів [3].

Дане ПЗ можна використовувати для перевірки інформації про транспортнийзасіб, читання помилок, очищення помилок, перегляду потоків даних в реальному часі, вилучення даних про автомобіль, виконання адаптації блоків і багато чого іншого.

TOAD Pro також надає можливості діагностики як через порт ODB1, так і через OBD2. Але за умови правильно обраної версії програми.

TOAD Pro здатний зберігати знімки даних про стан того чи іншого блоку в реальному часі, який робиться точно в той час, коли реєструється помилка.

Ці дані зможуть використовувати автомеханіки, щоб визначити, причину несправності в машині.

Переваги:

- надає широкі можливості діагностики;
- зчитування помилок у реальному часі;
- дає можливість очисти блоків автомобілів від помилок;
- має функцію збереження знімку даних блоку в певний момент часу;
- сумісний майже з усіма марками та моделями авто.Недоліки:
- дуже висока ціна;
- ПЗ сумісне лише з ОС Windows.
- потребує глибокого розуміння та навичок діагностування автомобіля.

1.5.2 AutoEnginuity’s ScanTool

Giotto ScanTool від AutoEnginuity - це потужна діагностична платформа для комп’ютера під керуванням ОС Windows [4]. Ця діагностична програма пропонує професійні варіанти розширеного діагностування для 58 автовиробників. Завдяки вдосконаленим розширенням надає змогу отримати доступ до ABS, подушок безпеки, силового агрегату, приладової панелі, трансмісії та десятків інших блоків автомобіля. Діагностика складається з даних в реальному часі, двонаправлених елементів керування, адаптивних та пошуку і очищення коду несправності.

Надає можливості з генерації графіків, які можуть панорамувати, масштабувати та відображати інформацію.

Переваги:

- надає широкі можливості діагностики;
- надає можливість розшифровування VIN коду;
- зчитування помилок у реальному часі;
- має можливість налаштування та калібрування блоків;
- сумісний майже з усіма марками та моделями авто, які були випущені після1996 року.

Недоліки:

- потребує придбання спеціального пристрою та не є сумісним зуніверсальними адаптерами;
- ПЗ сумісне лише з ОС Windows.

1.5.3 PCMScan

PCMSCAN - це повнофункціональний загальний сканер та діагностичний інструмент OBD-II, який підтримує широкий спектр апаратних інтерфейсів OBD-II [5]. Це дозволяє переглядати, складати графіки, реєструвати та відтворювати діагностичні дані в реальному часі через порт діагностичних OBD-II автомобіля. Це також дозволяє переглядати діагностичні коди несправностей автомобіля, дані про авто в реальному часі та іншу інформацію про транспортний засіб.

PCMSCAN підтримує майже все сучасні автомобілі.Переваги:

- надає широкі можливості діагностики;
- зчитування помилок у реальному часі;
- очистка помилок
- сумісний майже з усіма марками та моделями авто, які були випущені після1996 року.

Недоліки:

- ПЗ сумісне лише з ОС Windows
- потребує глибокого розуміння та навичок діагностування автомобіля.

1.5.4 ProScan

Це програмне забезпечення було створено для роботи як самостійний сканер або може бути інтегровано з іншими програми [6]. Це засіб для персонального комп’ютера, який може діагностувати та виправляти майже всі види помилок, пов’язаних з автомобілем. Його можна використовувати з багатьма сканерами та копірами, включаючи OBD-I , OBD II.

Переваги:

– має простий та інтуїтивний інтерфейс;

– генерація графіків;

– зчитування помилок у реальному часі;

– очистка помилок

– сумісний майже з усіма марками та моделями авто, які були випущені після 1996 року.

Недоліки:

– висока вартість.

1.6 Актуальність

Проаналізувавши програми аналоги – можна зробити висновки, що на сьогоднішній день на ринку представлена велика кількість програмних продуктів, які надають можливості комп’ютерної діагностики автомобіля у реальному часі з використанням універсальних зчитувальних OBD пристроїв.

Але більшість з цих продуктів мають направленість у першу чергу на професійних автомеханіків, що займаються діагностикою машин, мають професійне обладнання та широкі знання в області діагностування автомобіля та його будову.

Це обумовлює те, що подібні продукти не підходять для використання їх широкими масами автомобілістів, оскільки вони не мають ні дорогого обладнання,ні професійної освіти. Вони в цілому не потребують подібних професійних інструментів.

До того є такі рішення найчастіше коштують великих грошей і їх покупка є просто нерентабельною для звичайного автолюбителя, якому потрібно отримати деяку базову інформацію про своє авто.

Ще одним важливим фактором є орієнтованість подібного програмного забезпечення на операційну систему Windows, оскільки в більшості випадків це професійні застосування, які використовуються на станціях технічного обслуговування автомобілів, де є безпроблемний доступ до персонального комп’ютера і в цілому є прийнятним використання подібного пристрою.

В сучасному світі коли все програмне забезпечення є орієнтованим на

мобільні платформи, особливо актуальним є питання розробки системи, що дозволить простим автовласникам з легкістю діагностувати власне авто,використовуючи при цьому недорогий зчитувальний пристрій.

1.7 Огляд сканерів

1.7.1 BAFX Products Bluetooth Scanner

Цей сканер має один з кращих рейтингів на платформі Amazon. Він сумісний з Android і Windows і дозволяє прочитати коди Check Engine, перезавантажити/очистити Check Engine Light, прочитати дані різних сенсорів автомобіля, перевірити чи готовий автомобіль пройти тест на викид вихлопних газів і багато іншого. Ціна 22$. Його зображено на рисунку 1.1.

Рисунок 1.6 Сканер BAFX Products

1.7.2 Panlong Bluetooth OBD2 Car Diagnostic Scanner

Компактний сканер від компанії Panlong має непоганий рейтинг на Amazon, плюс може похвалитися привабливою ціною. Він дозволяє читати помилки і різні дані автомобіля, недоступні на панелі приладів. Ціна 18$. Його зображено на рисунку 1.2.

Рисунок 1.7 Сканер Panlong

1.7.3 ScanTool OBDLink LX Bluetooth

А це вже більш професійний OBD-II сканер від компанії ScanTool. Виробникзаявляє про збільшену швидкість роботи пристрою, в порівнянні з конкурентами, підтримку технології BatterySaver і різних інструментів для глибокої діагностики автомобіля. Ціна 50$. Його зображено на рисунку 1.3.

Рисунок 1.8 Сканер ScanTool OBDLink LX

1.7.4 iSaddle Super Mini Bluetooth OBD2 Scan Too

Компактний і недорогий OBD-II сканер. Підтримує безліч протоколів, працює з Windows-пристроями і Android-смартфонами. Ціна 12$. Його

зображено на рисунку 1.4.

Рисунок 1.9 Сканер iSaddle Super Mini

1.7.5 ELM327

Один з найпопулярніших та найпоширеніших мобільних сканерів на ринку, надає можливість з’єднання за допомогою Bluetooth, підтримує роботу з Android таWindows. Ціна 10$. Його зображено на рисунку 1.5.

Рисунок 1.10 Сканер ELM327

1.8 Висновки

На сьогоднішній день існує велика кількість доступних сканерів, шо є доступними для рядового користувача і також є універсальними та можуть взаємодіяти з різними автомобілями.

РОЗДІЛ 2 МЕТОДИ ТА ЕТАПИ ДОСЛІДЖЕНЬ

2.1 Обґрунтування методів досліджень та послідовність етапів проведеннянаукових досліджень

Аналіз - це метод наукового дослідження шляхом розкладання об’єктів на компоненти, а синтез - це поєднання частин, отриманих протягом аналізу. Методианалізу та синтезу в наукових дослідженнях органічно взаємопов’язані і можуть приймати різні форми в залежності від особливостей предмета, мети дослідження,ступеня знання предмета та глибини проникнення в його суть.

Метод ідеалізації – це духовна побудова об’єктів, яких в реальності не існує або в реальності неможливо. Метою ідеалізації є позбавлення реальних об’єктів деяких їх унікальних властивостей і надання їм певних нереальних та вигаданих властивостей.

Наукове дослідження – процес вивчення, експерименту, концептуалізації та перевірки теорії, пов’язаний з отриманням наукових знань.

Дослідження починаються з розробки програми. Програма обстеження - це документ, який регулює всі етапи, підготовку, організацію та етапи здійснення конкретного обстеження. Програма дослідження включає методологічні підходи та теоретичні демонстрації методологічних методів вивчення конкретних явищ чи процесів.

Початком наукових досліджень є детальний аналіз сучасного стану розглянутої проблеми. Це базується на пошуку інформації, який активно використовує джерела у глобальній комп’ютерній мережі Інтернет. На основі аналізу проблеми редагуються огляди та звіти, даються ключові класифікації напрямів та визначаються конкретні завдання дослідження.

Фактичною практикою наукових досліджень є вирішення спочатку поставлених завдань. Використовується математичне моделювання. Математичне моделювання передбачає кілька послідовних кроків. Це редагування математичноїмоделі дослідницького процесу на основі зібраних

даних, використання готової моделі дослідницького процесу на основі зібраних даних або використання готовоїмоделі з основами та коригуваннями. Допоміжний фактор. Для зручності визначення поставленої задачі математичний опис явища здійснюється у безрозмірних одиницях на основі теорії подібності. Потім розгляньте деякі умови та оберіть метод (аналітичний або наближений) для вирішення проблеми. Термін виконання; оптимальна вартість матеріалу. Результати експерименту обробляються за допомогою комп'ютера.

Завершення наукової розробки - це аналіз отриманих результатів та їх проектування. Теоретичні та експериментальні результати порівнюються та дається аналіз їх можливих відмінностей. Звіт про проведені наукові дослідження буде підготовлений та підготовлений відповідно до державних стандартів.

Виокремимо п'ять основних етапів прикладних досліджень.

– формулювання теми;

– формулювання цілей та завдань дослідження (огляд літератури, порівняння та критика проблемної інформації, узагальнення та висвітлення проблем за темами);

– теоретичні дослідження (дослідження фізичної природи явищ, формулювання гіпотез, виведення математичних залежностей та їх теоретичний аналіз);

– експериментальні дослідження (розробка експериментальних цілей та завдань, планування, вимірювальні засоби, постановка експериментів, проведення експериментів, обробка результатів);

– аналіз та проектування результатів досліджень (загальний аналіз теоретичних та експериментальних досліджень, порівняння цих результатів, аналіз відмінностей, уточнення теорії за потребою, проведення додаткових експериментальних досліджень).

2.2. Особливості розробки діагностичного мобільного додатка для операційної системи IOS

2.2.1 Аналіз особливостей даної розробки

Головна особливість даної розробки – необхідність компонувати різні програмні модулі для їх коректної роботи. Два головні модулі – це модуль отримання даних з OBDII-адаптеру (далі OBD Service) та модуль передачі декодованих даних на віддалений сервер (далі Remote Service).

Для мінімізації тестування та забезпечення гнучкості додатку, основні модулі повинні функціонувати незалежно один від іншого.

Кожен модуль виконує лише одну конкретну задачу, але не бере залишкової відповідальності. Наприклад, OBD Service не може відображати діагностичні дані на екрані мобільного пристрою або зберігати їх до локального сховища. Єдине його призначення – передати дані з адаптеру до додатку.

Такий підхід до розробки не новий і базується на виконанні принципів SOLID – це принципи об’єктно-орієнтованого програмування, які описують основні п’ять критеріїв, що забезпечують можливість підтримувати розроблену систему.

Опис критеріїв SOLID представлений у таблиці 2.1:

Таблиця 2.1

Опис критеріїв SOLID

Літера	Абревіатура	Назва	Опис
S	SRP	Принцип єдиного обов’язку (Single responsibility principle)	Визначає, що клас об’єкту маєвиконувати лише одну задачу, яку він інкапсулює
O	OCP	Принцип відкритос-ті/закритості (Open/closed principle)	Визначає, що класи, функції тазмінні повинні бути закритимидля зміни, але відкритими для розширення.
L	LSP	Принцип підстановки Лісков (Liskov substitution principle)	Клас наслідник повинен доповнювати, а не заміщати поведінку базового класу
I	ISP	Принцип розділення інтерфейсу (Interface segregation	Класи не повинні містити методів, які вони не використовують.

		principle)	
D	DIP	Принцип інверсії залежностей	Модулі вищого рівня не повинні залежати від модулів нижчого рівня. Абстракції не повинні залежати від деталей реалізації. Деталі реалізації повинні залежати від абстракцій.

Аналіз переваг та недоліків принципів SOLID приведено в таблиці 3.

Таблиця 3

Аналіз переваг та недоліків принципів SOLID

Принципи SOLID	Переваги	Недоліки
SRP	Зменшення кількості дублюючого коду	Зростання кількості класів призводить до ускладнення системи, але гарна структуризація врівноважить цей недолік
	Зменшення ймовірності змін у вже затвердженому класі	
	Відповідність назв клас на опису їх функціональності	
OCP	Легкість додавання нового функціоналу без необхідності дублювання та редагування вже існуючого коду	Легкість додавання нового функціоналу без необхідності дублювання та редагування вже існуючого коду
LSP	Дозволяє створювати гнучку систему наслідування	Цей принцип не збільшує кількість елементів системи (класів, функцій тощо), тому можна визначити, що він не має недоліків
ISP	При об’явлені нової реалізації інтерфейсу не треба реалізувати не потрібні (пусті методи)	Велика кількість інтерфейсів призводить до ускладнення системи
DIP	Дозволяє створювати інтуїтивно зрозумілі ієрархії наслідування від абстрактних класів до реалізацій	Цей принцип не збільшує кількість елементів системи (класів, функцій тощо), тому можна визначити, що він не має недоліків

Проведений аналіз доводить доцільність слідування принципам SOLID при розробці магістерської роботи.

2.2 Архітектурні аспекти мобільного додатку

Оскільки програмний продукт, що розробляється повинен підтримувати можливість подальшого впровадження нових функцій та модифікуваннявже існуючих, виникає необхідність розробити архітектурну структуру додатку.

Існує безліч архітектурних рішень для мобільних додатків. Далі представлені найбільш уживанні з них:

- MVC (Modal View Controller);
- MVP (Modal View Presenter);
- MVVM (Modal View ViewModel);
- VIPER (View Interactor Presenter Entity Router);
- VIP (View Interactor Presenter), також відомий як Clean Architecture.

Слід зазначити ознаки гарної архітектури:

- збалансований розподіл обов’язків між сутностями з жорсткимиролями;
- можливість тестування - зазвичай випливає з першої ознаки;
- простота використання і низька вартість обслуговування.

Розподіл зменшує навантаження на мозок, коли ми намагаємося з’ясувати, як працює та чи інша сутність. Таким чином, найпростіший спосіб зменшити складність полягає в поділі обов’язків між декількома сутностями за принципом єдиної відповідальності. (див. перший принцип SOLID)

Можливість тестування визначає, наскільки легко буде писати юніттести, а найчастіше - чи можна їх писати взагалі. Тести дозволяють розробникам виявляти помилки, які виникають на етапі виконання додатку та оберігають від необхідності стикатися с критичними помилками вже на пристроях користувачів, коли виправлення було б можливо якнайменш через тиждень.

Варто відзначити, що кращий код - це код, який ніколи не був написаний. І чим менше у вас коду, тим менше помилок. Тому бажання писати менше коду зовсім не говорить про те, що розробник лінується. А вибираючи найрозумніше рішення, потрібно завжди враховувати вартість його підтримки.

Проаналізуємо відомі архітектури окремо.

MVC

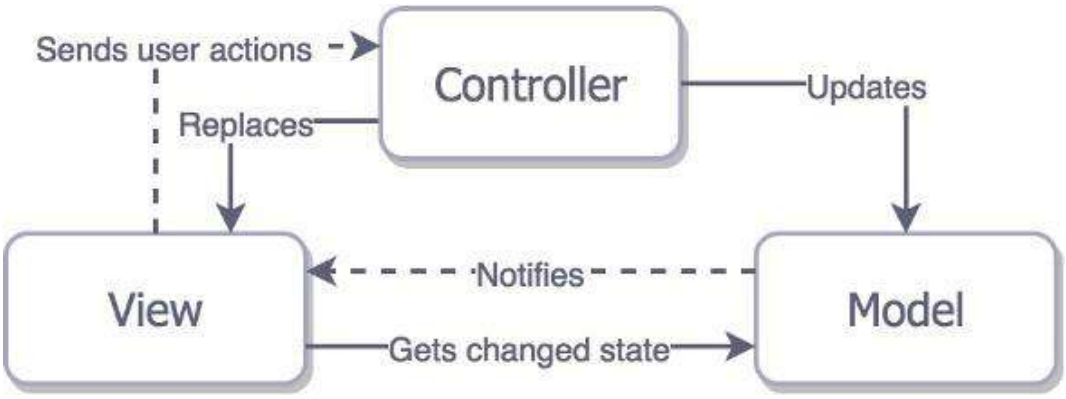


Рисунок 2.1 – Структура архітектури MVC

У традиційному MVC, який ще не був запропонований як офіційна архітектура від Apple, View не зберігає свого стану. Controller «рендерить» View кожен раз при змінах Model. Наприклад, модальне вікно буде повністю перевантажуватись, коли достатньо лише завантажити новий текст повідомлення. Хоча можна реалізувати традиційний MVC у iOS додатку, це не має сенсу через архітектурної проблеми: все три сутності тісно пов’язані, кожна з них знає про двох інших. Це сильно знижує можливість повторного використання кожного з елементів. З цієї причини традиційний MVC не є доцільним варіантом при виборі архітектури.

Cocoa MVC

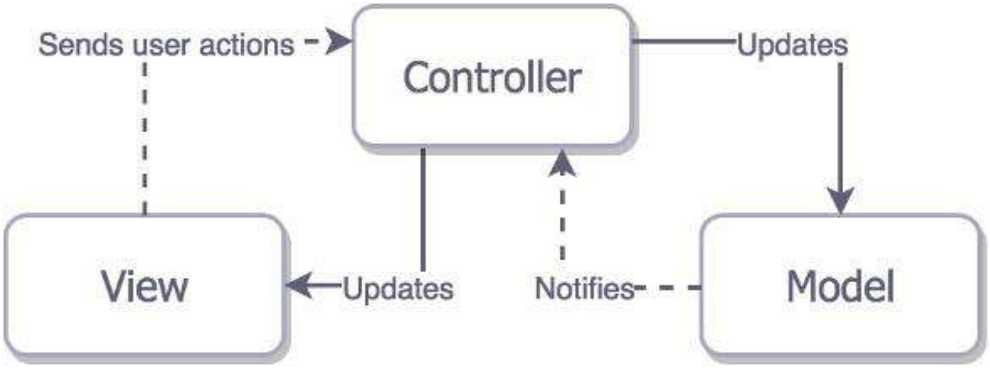


Рисунок 2.2 – Структура архітектури Cocoa MVC

Controller є зв’язуючою ланкою між View і Model, отже, дві останніх не знають про існування один одного. З першого погляду, все зрозуміло. Але існує і інша інтерпретація цієї архітектури як Massive View Controller.

Cocoa MVC заохочує вас писати Massive View Controller, тому що

контролер настільки залучений в життєвий цикл View, що важко вважати його окремою сутністю. У більшості випадків вся відповідальність View полягає в тому, щоб відправити дії до контролера. В підсумку все закінчується тим, що View Controller стає делегатом і джерелом даних.

Складується враження, що Cocoa MVC є досить поганим вибором архітектурного партерну. Але його оцінка з точки зору ознак хорошої архітектури, визначених вище:

- розподіл: View і Model насправді розділені, але View і Controller тісно пов’язані;
- тестування: через поганий розподіл можна реалізувати unit-тести лише для Model;
- простота використання: найменша кількість коду у порівнянні з іншими патернами.

Cocoa MVC – це розумний вибір, якщо розробник не планує витрачати багато часу на більш складні патерни, а проект розробки не поділяється на багато компонентів.

MVP

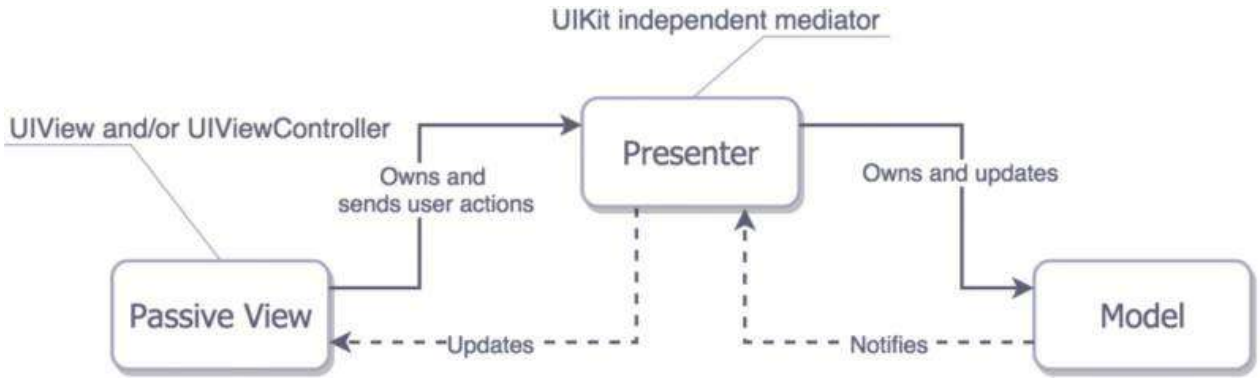


Рисунок 2.3 – Структура архітектури MVP

Посередник в MVP – Presenter, не має відношення до життєвого циклу View Controller. View може бути легко замінена Mock-об’єктами для тестування, тому в Presenter немає layout-коду, але він відповідає за оновлення View.

З точки зору MVP, підкласи UIViewController насправді є View, а не Presenter. Ця різниця забезпечує гарні можливості для тестування, але

знижується швидкість розробки, тому що дані і події пов’язуються вручну саме між View і Presenter.

Давайте подивимося на ознаки гарної архітектури для MVP:

- розподіл: велика частина відповідальності розділена між Presenter і Model, а View лише комутує їх;
- тестування: ми можемо перевірити більшу частину бізнес-логіки завдяки бездіяльності View;
- простота використання: кількість коду в два рази більше в порівнянні з MVC, але в той же час ідея MVP дуже проста.

MVP в iOS означає чудову можливість для написання unit-тестів і багато коду, тому цей архітектурний патерн можна використовувати для виконання цього дипломного проекту.

MVVM

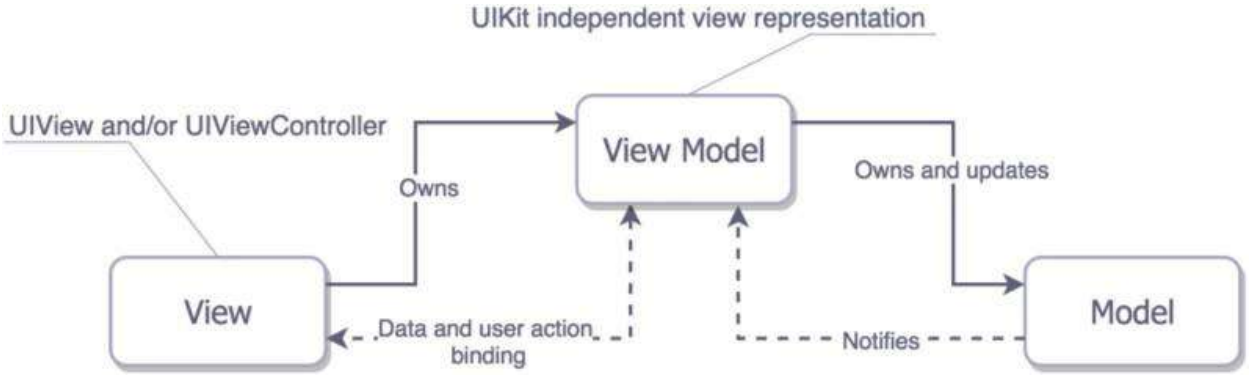


Рисунок 2.4 – Структура архітектури MVVM

MVVM є найновішим із патернів. View Model – це незалежне від UIKit уявлення View. View Model викликає зміни в Model і самостійно оновлюється з уже оновленої Model. І так як Біндінг відбувається між View і View Model, то перша, відповідно, теж оновлюється.

І знову повернемося до оцінки ознак хорошої архітектури:

- розподіл: в MVVM View має більше обов’язків, ніж View з MVP. Тому що перша оновлює свій стан з View Model за рахунок установки прив’язок між View та Viw Model, тоді як друга направляє всі події в Presenter і не оновлює себе власноруч (це робить Presenter);

- тестування: View Model не знає нічого про View, це дозволяє нам з легкістю тестувати її;
- простота використання: той же обсяг коду, як в нашому прикладі MVP, але в реальному додатку, де вам доведеться направити всі події з View в Presenter і оновлювати View вручну, MVVM буде набагато стрункішою.

MVVM є дуже привабливим патерном, так як він поєднує в собі переваги вищезазначених підходів і не вимагає додаткового коду для оновлення View через зв'язки між View та View Model. Проте, тестування все ще знаходиться на хорошому рівні.

VIPER

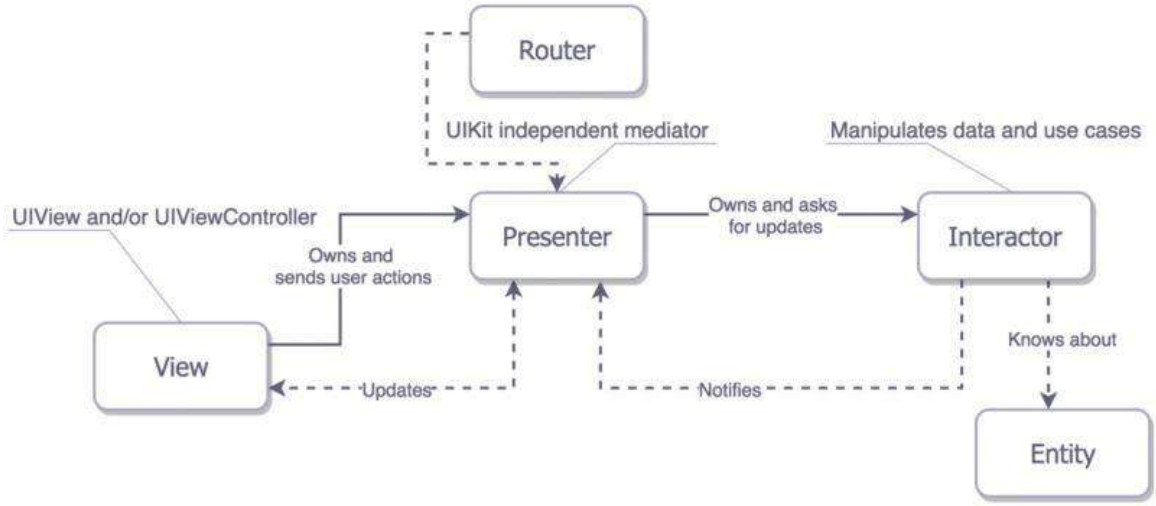


Рисунок 2.5 – Структура архітектури VIPER

VIPER робить ще один крок в бік розподілу обов'язків і замість трьох шарів пропонує п'ять.

Interactor містить бізнес-логіку, пов'язану з даними (Entities): наприклад, створення нових екземплярів сутностей або отримання їх з сервера. Для цих цілей використовуються Сервіси, які розглядаються швидше як зовнішні залежності, а не як частина модуля VIPER.

Presenter містить бізнес-логіку, пов'язану с UI (але UIKit-незалежну), викликає методи в Interactor.

Entities – прості об'єкти даних.

Router несе відповідальність за переходи між VIPER-модулями.

Повернемося до ознак гарної архітектури:

- розподіл: безсумнівно, VIPER є лідером у розподілі обов’язків;
- тестування: чим краще розподіл, тим легше написання тестів;
- простота використання: перші два переваги йдуть за рахунок величезної збільшення кількості коду.

Використання VIPER може бути обумовлено у дуже складних проектах, які об’єднують якнайменш 10 модулів, та потребують постійної модифікації без зайвих витрат часу. Гнучка система дозволяє легко долучати або видаляти функціонал, але розробка основного каркасу потребує написання багатьох інтерфейсів та їх реалізацій.

Аналіз кількох основних архітектурних патернів показав, що не існує одного ідеального рішення для кожного проекту. Вибір архітектури є питанням зважування компромісів у конкретній ситуації.

Найкращим варіантом є поєднання декількох архітектур в одному додатку. Більш прості екрани можуть проектуватися за MVC, а десь доцільніше розділити відповідальність між View та Presenter (або View Model).

2.3 Проектування та організація структури додатку

Для визначення структури додатку треба виділити декілька термінів, що стосуються організації робочого простору для додатків, що розробляються для платформи iOS.

На вершині ієрархії додатку стоїть Робоча Область (Workspace). Вона містить, власне, проект, який розробляється, а також залежності до інших проектів, бібліотек та зовнішніх утиліт.

Для імпорту бібліотек можна використовувати стандартні зв’язки, додаючи код фреймворку до існуючого проекту, але цей спосіб має декілька недоліків. Якщо у розробника використовуваної бібліотеки з’явиться необхідність оновити її або справити помилки, нова версія не буде доступна всім тим, хто використовував цю бібліотеку раніше. Це примушує розробника вручну встановлювати нову версію фреймворку. С цього випливає наступна проблема: як визначити момент часу, коли стала доступна нова версія? Звісно

можна перевіряти це вручну, але не виключно, що встановлено багато різ- них фреймворків. Це може змусити розробника робити перевірки версій так часто, як він перевіряє електрону пошту, або робить комміт до системи контролю версій. Звичайно, такі перевірки витрачають зайвий час.

Необхідність відстеження актуальних версій фреймворків призвела до появи незамінного інструменту як менеджер залежностей рівня додатків – Cocoa Pods.

Cocoa Pods – це дуже потужний інструмент, яким має володіти кожний iOS-розробник. За допомогою нього ми можемо легко, швидко і просто підключати різні бібліотеки та утиліти, які значно полегшують нам життя при розробці додатків.

Podfile – це специфікація, в якій позначені всі залежності, необхідні для нашого проекту. Podfile завжди створює неявну ціль (target). Цей файл немає розширення і повинен знаходитеся в одній папці з файлом .xcodepoj проекту.

Приклад Podfile:

```
platform
:ios, ‘11.0’
target ‘myapp’
do
      pod ‘Reachability’, ‘~>
      3.1.1’pod ‘sqlite3’
end
```

Спочатку вказується мінімальна версія iOS, яка буде підтримувати розроблений додаток. В полі «target» вказується назва додатку, а в блоці «doend» список бібліотек, які треба встановити. За бажанням, можна вказувати параметр версії. Це може знадобитися, тільки якщо актуальна версія містить критичні помилки та не може використовуватися у проекті.

Cocoa Pods зчитує Podfile обираючи необхідну версію кожної бібліотеки в залежності від параметрів додатку та пристроїв, на яких додаток повинен запускатися.

Для запуску менеджера залежностей треба виконати команду «pod install» із термінала Mac, знаходячись у директорії проекту.

Класична структура мобільного додатка для платформи iOS представлена на рис. 2.6:

Рисунок 2.6 – Структура додатку для мобільної платформи iOS

2.4 Технологія

2.4.1 Мова програмування

Мобільні додатки для платформи iOS можуть розроблятися використовуючи різноманітний стек технологій. Існує два шляхи, що залежать від вибору мови програмування – Objective-C або Swift.

Objective-C – це перевірена часом мова, яка не мала аналогів до 2014 року, коли з'явилась перша версія Swift. Objective-C має C-подібний синтаксис та дозволяє вручну керувати пам'яттю. Це гарне рішення для додатків, які були створені ще для iOS 7.0 та потребують підтримки та оновлення.

Але майбутнє за Swift. Ця мова була створена компанією Apple виключно для розробки додатків під iOS. В порівнянні з Objective-C був значно спрощений синтаксис. Наприклад, більше не треба писати на кожен клас окремий заголовок-інтерфейс, що значно прискорює розробку. Говорячи о швидкодії, Swift також бере верх. Його новітня система автоматичного контроль за пам'яттю, яка називається ARC (Automatic Reference Counting), рахує активні посилання до кожного об'єкту – якщо кількість посилань дорівнює нулю, об'єкт буде видалений із пам'яті. Також Swift підтримує

компіляцію файлів Objective-C. Це дає змогу використовувати перевірені часом фрейм-ворки у нових проектах.

Можна з повною впевненістю сказати, що Swift - це майбутнє. Все пізнається в порівнянні і, лише відпрацювавши певну кількість проектів, можна стверджувати, чи зручна вибрана мова, і чи може вона ефективно виконувати задачі, які стоять перед розробником.

Незважаючи на песимізм і консерватизм багатьох компаній, заснований на безлічі заяв про те, що Swift сируватий і має безліч недоліків, ця мова активно розвивається. На даний момент остання версія 5.0 є рішенням 99% скарг на попередні версії, а також має нову систему компіляції, яка збирає проект на 30-40% швидше, ніж передній Swift 4.

РОЗДІЛ 3 ДОСЛІДЖЕННЯ МЕТОДІВ КОП'ЮТЕРНОЇ ДІАГНОСТИКИ АВТОМОБІЛЯ ЗА ТЕХНОЛОГІЄЮ OBD2

3.1 Історія OBD2

Комп'ютерна діагностика автомобіля (англ. On-board diagnostics) - це діагностика систем автомобіля, яка виконується блоками керування різних агрегатів у автомобілі. Вона використовується як засіб само діагностування і виведення відповідних помилок на екран бортового комп'ютера авто за його наявності, так і для спеціалізованої діагностики на станціях технічного обслуговування.

Стандарт OBD2 було введено 1996 року в Сполучених Штатах Америки, а згодом він поширився всіх автовиробників.

Почалося все 1980 року, коли корпорація General Motors вперше запровадила протокол для діагностування авто ALDL (Assembly Line Diagnostic Link) та інтерфейс для перевірки роботи двигуна ECM.

Протокол ALDL фактично був попередником OBD2, адже міг постійно взаємодіяти з автомобілем та зчитувати дані з усіх блоків у автомобілі.

У 1984 році автовиробники почати активно впроваджувати системи комп'ютерної діагностики в свої автомобілі. Головні цілі були – підвищення активної безпеки автомобіля, зменшення викидів та збільшення комфорту для пасажирів.

Потім у 1991 році Каліфорнійська рада з питань повітряних ресурсів зобов'язала обладнувати всі нові автомобілі за стандартом OBD. Це був переломний момент у сфері комп'ютерної діагностики автомобілів. Але тоді основною метою було спонукати виробників машин розробляти більш ефективні системи контролю за викидами.

У 1996 році стандарт OBD-II, який і досі є актуальним, став обов'язковим дляавтомобілів, що продавалися новими у Сполучених Штатах Америки.

У 2000 році у Європейському Союзі ввели власну версію стандарту

EOBD – вона базується на стандартній OBD-II, має більш широку мережу контролерів. Вона є обов’язковою для всіх бензинових машин, шо були випущені після 2001 року, а також для всіх дизельних починаючи з 2004 року.

3.1.1 ALDL

GM ALDL (діагностичне посилання збірної лінії) вважається попередником стандарту OBD-I. Цей інтерфейс вироблявся в різних варіантах. Різні версії мали незначні відмінності в розпіновці і швидкості передачі даних. Раніше версії використовували швидкість передачі 160 біт/с, тоді як пізніші версії мали пропускну здатність до 8192 біт/с і використовували двонаправлений зв’язок з різними блоками в автомобілі.

3.1.2 OBD-I

Регулюючий намір OBD-I полягав у тому, щоб заохотити автовиробників розробляти надійні системи контролю викидів, які залишаються ефективними протягом терміну експлуатації автомобіля. Ідея полягала в тому, що, примусове щорічне тестування викидів для Каліфорнії, та відмови в реєстрації транспортним засобам, які не пройшли перевірку, стимулюватимуть водіїв, купувати транспортнізасоби, які б відповідали всім нормам. OBD-I в основному був невдалим, оскільки засоби звітності про специфічну діагностичну інформацію не були стандартизованими. Технічні труднощі з отриманням стандартизованої та надійноїінформації про викиди з усіх транспортних засобів призвели до неможливості ефективно виконувати річну програму випробувань.

Діагностичні коди несправностей в OBD-I, як правило, можна було знайти без дорогого спеціалізованого обладнання. Проте кожен виробник використовував власний з’єднувач, його розташування та процедуру для зчитування інформації з автомобіля. Також автомобілі обладнані OBD-I мали

менше доступних кодів несправностей , ніж для автомобілі, обладнані OBD-II.

3.1.3 OBD-1.5

Стандарт OBD-1.5.прийшов на зміну першій його генерації та та вважаєтьсячастковою реалізацію OBD-II, яку General Motors використовували на деяких транспортних засобах у 1994-1996 роках.

Для читання кодів, сформованих OBD-1.5, був необхідний спеціальний засібсканування, який відрізнявся від попередника.

3.1.4 OBD-II

OBD-II є продовженням розвитку технології. Він переважає попередників з точки зору можливостей та стандартизації. Ця версія стандарту запровадила єдиний тип діагностичного роз'єму.

OBD-II є покращенням порівняно з OBD-I як з точки зору функціональності, так і стандартизації. Стандарт OBD-II визначає тип діагностичного роз'єму та його вихід, доступні протоколи електричної сигналізації та формат обміну повідомленнями. Він також надає перелік параметрів транспортного засобу для спостереження та спосіб кодування інформації. У роз'ємі є елементи, які живлять сканер від автомобільного акумулятора, усуваючи необхідність підключати сканердо окремого джерела живлення. Однак деякі технічні працівники можуть підключити сканер до допоміжного джерела живлення, щоб захистити дані у випадку аномалії, коли транспортний засіб втрачає живлення через поломку. Нарешті, стандарт OBD-II надає перелік кодів DTC(diagnostic trouble code) тобто кодів помилок. В результаті цієї стандартизації зчитувальний пристрій може отримувати дані з вбудованого комп'ютера будь-якого автомобіля. OBD-II був доступний у двох моделях, OBD-IIA та OBD-IIB.

Специфікація OBD-II забезпечує стандартизований апаратний

інтерфейс (материнський 16-контактний (2x8) роз’єм J1962). На відміну від роз’єму OBD-I, який може знаходитись під капотом автомобіля, роз’єм OBD-II повинен знаходитися на відстані 0,61 м від рульового колеса в межах досяжності водія.

Доступні діагностичні дані OBD-II. OBD-II забезпечує доступ до даних з блоку управління двигуном (ЕБК) та надає цінне джерело інформації для усуненнянесправностей у салоні автомобіля. Стандарт SAE J1979 визначає, як запитувати список різних діагностичних даних та стандартних параметрів, доступних у ЕБК. Різні доступні параметри визначаються за допомогою "ідентифікаційного номера параметра" або PID, визначеного в J1979. Виробники не зобов’язані використовувати всі піди, перелічені у стандарті, і можуть включати власні піди, яких немає в списку. Список загальних кодів несправностей OBD-II,запропонованих SAE, див. Додаток Б. Деякі виробники часто додають запатентовані DTC, щоб поліпшити свої набори кодів OBD-II.

Дослідники з Університету Вашингтона та Університету Каліфорнії вивчилибезпеку навколо БД і виявили, що вони змогли отримати контроль над багатьма компонентами транспортного засобу за допомогою інтерфейсу. Крім того, вони змогли завантажити нову прошивку в блоки управління двигуном . Їхній висновокполягає в тому, що вбудовані в транспортні засоби системи не розробляються з урахуванням безпеки [7] – [8].

Повідомлялося, що злодії використовують спеціальні пристрої для перепрограмування OBD, щоб дозволити їм красти машини без використання ключа. Основні причини цієї вразливості у тому, що виробники транспортних засобів вдаються до розширення шини для цілей, відмінних від тих, для яких вонабула розроблена, та через відсутність автентифікації та авторизації [9].

3.2. Взаємодія зі сканером

Для підключення до автомобіля використовується стандартний діагностичний роз’єм OBD-II. Більшість серійних автомобілів, випущених

після 1996 року, вже оснащені таким роз’ємом. Хоча такий роз’єм діагностики і стандартизований, але в ньому підтримуються відразу кілька протоколів різних систем управління двигуном, тобто фізично використовуються різні контакти на роз’ємі, які повинен знати комунікаційний модуль сканеру. Відповідно в різних марках автомобілів можуть бути різні внутрішні шини отримання даних діагностики з боку управління двигуна.

Так чи інакше, але всі спеціалізовані рішення - це більш досконалі промислові вироби, в порівнянні зі звичайним пристроєм зчитування кодів діагностики на базі мікросхеми ELM327 [10] канадської компанії Elm Electronics. ELM327 - це універсальний перетворювач протоколів, використовуваних в діагностичних шинах автомобілів, в послідовний протокол типу RS-232.

Взаємодія з ELM327 здійснюється стандартними AT-командами, підтримуваними мікросхемою. Потрібно просто організувати обмін текстовими повідомленнями по протоколу RS-232. А саме фізичне з’єднання низького рівня по USB, Bluetooth або Wi-Fi реалізується із застосуванням мікросхем перетворення послідовного протоколу RS-232.Тобто, щоб перетворити автомобіль в пристрій IoT цілком достатньо, підключити мікросхему ELM327 до діагностичного роз’єму OBD-II і на виході цієї мікросхеми, наприклад, поставити перетворювач послідовного інтерфейсу в Bluetooth або Wi-Fi. Потім, можна зі свого смартфона або ж комп’ютера «зчитувати» діагностику автомобіля. Втім, таких готових модулів або блоків на ринку більш ніж достатньо. А їх ціна на AliExpress коливається в межах від декількох доларів до тисяч за професійні пристрої. Сканер ELM327 зображено на рисунку 3.1.

Рисунок 3.1 Сканер ELM327

Для взаємодії з автомобілем використовується найбільш поширений інтерфейс CAN (Controller Area Network). Свого часу, стандарт CAN, запропонований компанією Bosch, зробив помітний прогрес в розробці систем дляавтомобільної електроніки. Якщо автомобіль в мережі Інтернет з'явився тільки недавно, то концепція мережі всередині автомобіля існує вже з середини 80-х. Ідеядуже проста, і як Ethernet зробив прорив в комп'ютерних мережах, так і CAN став основою надійних комунікацій усередині автомобіля.

CAN bus - це автомобільна шина, розроблена Робертом Бошем, яка переважно прийнята в автомобільній і авіакосмічній індустрії. CAN це серійний протокол шини, з підключенням окремих систем і датчиків як альтернатива звичайному багатопровідному пучку. Дозволяє зв'язати автомобільні компоненти водин дріт званий мережею шини даних з високою швидкістю передачі інформації. До того як була випущена CAN шина, автомобілі мали велику кількість проводки яка була необхідна для з'єднання різних електронних компонентів. Тобто це шина,шо з'єднує між собою різні блоки та компоненти в автомобілі та дозволяє ім обмінюватися інформацією в реальному часі.

Раніше в автомобілях до центрального блоку управління двигуном сходилисяшини різних модулів і пристроїв. Послідовна шина CAN дозволила реалізовувати незалежні інтелектуальні модулі, які спілкуються один з одним

фактично за допомогою мережевого протоколу. При цьому значно зменшується кількість проводки всередині автомобіля.

Передача даних в CAN шині нагадує модель «видавець-передплатник», де кожен пристрій на шині має унікальний ідентифікатор і, коли передає дані один пристрій, то всі інші слухають, і приймають рішення на основі цього ідентифікатора - чи потрібні саме їм ці дані для прийому і обробки чи ні.

Саме ця особливість дозволяє користувачу приєднати діагностичний сканер через спеціальний порт та в режимі реального часу отримувати інформацію про свій автомобіль, зчитувати помилки тощо.

3.3.Методи комп’ютерної діагностики

Основним методом комп’ютерної діагностики є підключення спеціальних сканерів які з’єднуються з автомобільною електронікою та зчитують дані в цифровому вигляді.

Спочатку засоби комп’ютерної діагностики використовуються для зчитування не тільки кодів помилок, а й усіх цифрових даних, які прямо пов’язані з проблемою, що сталася. Тут потрібно зрозуміти, що сканер може повністюрозшифрувати виявлені несправності.

На другому етапі всі ці дані потребують подальшого підтвердження. І перш за все потрібно ретельно перевірити електронні системи автомобіля, щоб переконатися, що вона повністю придатна для використання

Далі сканер відображає дані в режимі реального часу. Ця функція може бути використана для перевірки сигналів датчиків та інших елементів управління для виявлення несправностей.

Існує кілька методів діагностування. Перший це повне сканування всіх систем автомобіля та аналіз отриманих даних для виявлення несправностей з усіх агрегатів автомобіля. Така діагностика займає більше часу оскільки проводиться опитування усіх основних ЕБК.

Інший метод базується на перевірці певних електронних блоків керування на основі попередньої фізичної діагностики. Найчастіше він

застосовується професійними автомеханіками які мають розуміння будови авто та діагностуванняпевних вузлів. Після виявлення проблем у конкретному агрегаті чи вузлі проводиться комп'ютерна перевірка для більш глибокого розуміння проблеми.

3.4.Взаємодія сканера з автомобілем

Сучасні автомобілі - це не лише механічні деталі, це також безліч електронних блоків. Вони контролюють роботу різних систем і агрегатів, контролюють їх стан, а також відстежують їх помилки в роботі. Зчитування цих метрик робить можливим виконання комп'ютерної діагностики та виправлення несправностей.

Багатофункціональний адаптер, який підключається до стандартного порту OBD-2, підтримує машини різних марок та моделей. Спеціальне програмне забезпечення для них містить у базі даних як стандартні коди помилок, які є загальними для більшості автомобілів, так і коди унікальні для кожної марки.

Окрім зчитування кодів помилок, найновіші сканери OBD-2 пропонують кілька варіантів налаштування електроніки автомобіля.

Сканер OBD2 може аналізувати як всю бортову мережу, так і окремі системи. Можливість відстеження таких показників, як частота обертів двигуна, швидкість,температура масла та антифризу, тиск масла та палива, тиск у впускному колекторіта випускному колекторі в режимі реального часу.

Вся інформація з автомобіля надсилаються за допомогою спеціального адаптера ELM327, який підключається через порт OBD-2 і передає всі дані через Bluetooth.

Перед початком комп'ютерної діагностики потрібно встановити з'єднання зі сканером та виконати команди для ініціалізації. Список основних з них наведенонижче.

– atz – відновлення адаптера до заводських налаштувань;

– atl0 – вимкнення символу переносу строки;
– ate0-1 – echo off/on;
– ath0-1 – headers off/on;
– atstff – максимальний тайм аут;
– atdp – автоматичне визначення протоколу автомобіля;
– atspN – вибір протоколу за кодом.

Зазвичай набір команд для ініціалізації має наступний порядок, він підходитьдля більшості машин, але в деяких випадках він може незначною мірою відрізнятися: atz, atl0, ate1, ath1, atstff, atdp, atsp0.

Для зчитування діагностичних даних використовуються спеціальні команди

- піди (англ. PID)(Parameter ID) - код, який використовується для запиту інформаціїз певного блоку в машині.

Існує певний набір кодів, які підтримуються на всіх автомобілях з портом OBD-2. Існує також набір команд для певних виробників та типів автомобілів.

Кожна машина має власний набір команд, які вона підтримує. Це означає, щоякщо ви працюєте з певним автомобілем, потрібно враховувати те, шо він може не підтримувати ту чи іншу команду. Щоб визначити це - потрібно зробити спеціальний запит. У відповідь на цей запит машина поверне всі підтримувані команди. Це слід зробити відразу після встановлення зв’язку та початкової ініціалізації.

Запити на отримання інформації та відповідні реакції автомобіля виконуються у шістнадцятковій системі числення.

Наприклад, виконується команда для встановлення всіх команд, які підтримуються автомобілем. Якщо автомобіль повернув значення BB1E3211, то потрібно спочатку перевести результат у двійкову систему числення BB1E3211(16) = 10111011000111100011001000010001(2).

Потім побітово визначити які команди підтримуються машиною, використовуючи спеціальні таблиці, приклад якої зображено на рисунку 3.2

1	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**1**
+		+	+	+		+	+				+	+	+	+				+
01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13

1	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**
+			+					+				+
14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20

Рисунок 3.2 – Таблиця кодів.

Тобто можна визначити, шо підтримуються команди 01, 03, 04, 05, 07, 08, 0C, 0D, 0E, 0F, 13, 14, 17, 1C, 20.

Після цього можна оперувати даними командами з автомобілем та бути впевненим у тому, що він їх зрозуміє.

3.5. Бази кодів

Після того як було визначено усі підтримувані автомобілем команди, можна посилати запити та отримувати потрібну інформацію. Але для того щоб інтерпретувати її у прийнятний для людини вигляд – потрібно розшифрувати коди, які повертає машина. Тобто мати певну базу значень для всіх можливих кодів.

Кожна команда має свій унікальний ідентифікатор, формат відповіді, тобто кількість байтів, які вона повертає, діапазон припустимих значень, а також одиниці виміру в яких повертається відповідь.

Як вже згадувалось раніше, спеціальні коди – ідентифікатори параметра, які використовується для запиту діагностичної інформації з автомобілів. Стандарт SAE J1979 визначає стандартний список таких кодів, але виробники мають право додавати свої специфічні коди. Також виробники можуть використовувати не всі коди, регламентовані стандартному SAE J1979 [11].

Стандарт SAE J1979 регламентує 10 режимів роботи:

– 0x01. Show current data - зчитування поточних параметрів роботи системи управління;

– 0x02. Show freeze frame data - отримання збереженої копії поточних

параметрів роботи системи управління на момент виникнення кодів несправностей;

– 0x03. Show stored Diagnostic Trouble Codes - зчитування збережених кодів несправностей;

– 0x04. Clear Diagnostic Trouble Codes and stored values - стирання кодів несправностей, результатів тестів датчиків кисню, результатів тестових моніторів;

– 0x05. Test results, oxygen sensor monitoring (non CAN only) - зчитування і перегляд результатів тесту датчиків кисню (не для шини CAN);

– 0x06. Test results, other component / system monitoring (Test results, oxygen sensor monitoring for CAN only) - зчитування результатів тестів, які контролюють роботу каталізатора, системи рециркуляції вихлопних газів (EGR), системи вентиляції паливного бака. (Зчитування і перегляд результатів тесту датчиків кисню тільки для шини CAN);

– 0x07. Show pending Diagnostic Trouble Codes (detected during current or last driving cycle) - запит результатів діагностики безперервно діючих тестів, виконуваних постійно, поки виконуються умови для проведення тесту. Ці тести контролюють склад паливо-повітряної суміші, пропуски запалювання, іншікомпоненти, що впливають на вихлоп;

– 0x08. Control operation of on-board component / system - управління бортовими системами;

– 0x09. Request vehicle information – запит інформації про автомобіль, що діагностується: VIN-код і та різні дані;

– 0x0A. Permanent DTC’s (Cleared DTC’s) - помилки, які були видалені.

Але виробники також вільні запроваджувати свої режими роботи з порядковим номером більше 9 для власних потреб. Також вони можуть не підтримувати деякі в існуючих стандартних режимів.

Кожна категорія команд відповідно має власний набір підів з їх тлумаченням, всі ці дані доступні у вільному доступі.

Зазвичай цих команд достатньо для отримання базової інформації про

автомобіль, а також для його поверхневої діагностики у разі виявлення будь-яких проблем.

3.6.Існуючі бібліотеки

На даний момент у вільному доступі є рішення на популярних мовах програмування, які дозволяють полегшити розробку програми для комп’ютерної діагностики. Вони беруть на себе базові низькорівневі функції, такі як встановлення з’єднання зі сканером, базові налаштування тощо

3.6.1. OBD.NET

Бібліотека написана на мові програмування C#, дозволяє полегшити розробку програмного забезпечення на платформі .NET для діагностики авто за допомогою сканерів ELM327 або STN1170.

Вона надає можливість:

- встановлення з’єднання за допомогою Bluetooth;
- встановлення з’єднання за допомогою проводу;
- конфігурація з’єднання зі сканером;
- завантаження інформації про різні параметри автомобіля;
- завантаження кодів помилок з автомобіля;
- відправка команд у блоки керування.

Має зручний спосіб встановлення через пакетний менеджер на платформі .NET.

3.6.2. Obd-java-api

Бібліотека написана на мові програмування Java, дозволяє полегшити розробку програмного забезпечення для діагностики авто за допомогою сканеру ELM327.

Вона надає можливість:

- встановлення з’єднання за допомогою Bluetooth;
- конфігурація з’єднання зі сканером;
- завантаження інформації про різні параметри автомобіля;
- завантаження кодів помилок з автомобіля.

РОЗДІЛ 4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

В ході дослідження було вивчено основні можливості технології OBD2 для комп’ютерного діагностування автомобіля. Основною метою було вивчення можливості створення системи, що дозволить діагностувати будь-яке сучасне авто,маючи при цьому універсальний автомобільний сканер.

Основною перевагою технології OBD2 – є уніфікованість портів, а також стандартів передачі даних та взаємодії з ними.

Було виявлено, шо все сучасні автомобілі, які обладнані діагностичними портами стандарту OBD2 мають універсальний стандарт взаємодії з автомобілем, а також певний уніфікований набір команд, який підпорядковується стандарту SAEJ1979 [11], що дозволяє створити єдиний інтерфейс для взаємодії з автомобілем.

При цьому є нюанси, такі як те, що автовиробники вільні відходити від даного стандарту, привносити свої нововведення у своїх автомобілях, але для моєїзадачі високорівневої діагностики – це не є проблемою. Адже спираючись на відкриті джерела – базові діагностичні команди є універсальними для більшості машин.

Тобто при наявності відкритої бази основних команд, а також їх тлумачення, з універсальним діагностичним приладом можна діагностувати більшість сучаснихавтомобілів.

Кінцевий користувач має придбати адаптер типу ELM327, завантажити спеціальний додаток та використовувати систему.

При розробці програмного продукту доцільно використати гнучку методологію розробки, адже під час написання системи потрібно бути готовими дозміни вимог та швидкого виправлення недоліків [12].

Оскільки за специфікою роботи з адаптером ELM327 час від часу запити виконуються довго, через те що передача даних за допомогою протоколу RS232 відбувається послідовно, доречно буде використати асинхронну модель взаємодії задаптером на основі повідомлень з використанням меседж брокеру

[13].

Основна проблема є в тому, щоб перевірити коректність роботи з тим чи іншим автомобілем, адже особливості роботи з ними можуть відрізнятися навіть неміж марками або моделями, а навіть між комплектаціями однієї і тієї ж моделі, в залежності від наявності певних електронних опцій.

Тож після розробки системи з основними функціями дуже важливим етапомє тестування на різних марках та моделях автомобілів.

Очевидно, що неможливо протестувати продукт на значному відсотку всіх автомобілів. Тож доцільно буде після закритого тестування організувати відкрите бета-тестування до якого заохотити власники різних автомобілів задля покриття більшого відсотку протестованих авто та збору даних щодо дефектів пов’язаних з конкретними моделями для подальшого їх усунення.

Але навіть в такому випадку після повноцінного релізу необхідно налагодити процеси збору відгуків від користувачів та процес оперативного усунення подібнихнедоліків.

РОЗДІЛ 5 РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ

Моделювання системи виконано з використанням мови UML. В цьому розділі відображено основні діаграми які зображують загальну архітектуру системи.

5.1 UML проектування ПЗ

5.1.1 Діаграма компонентів

На рисунку 5.1 зображено діаграму компонентів. Вона відображає складові частини системи.

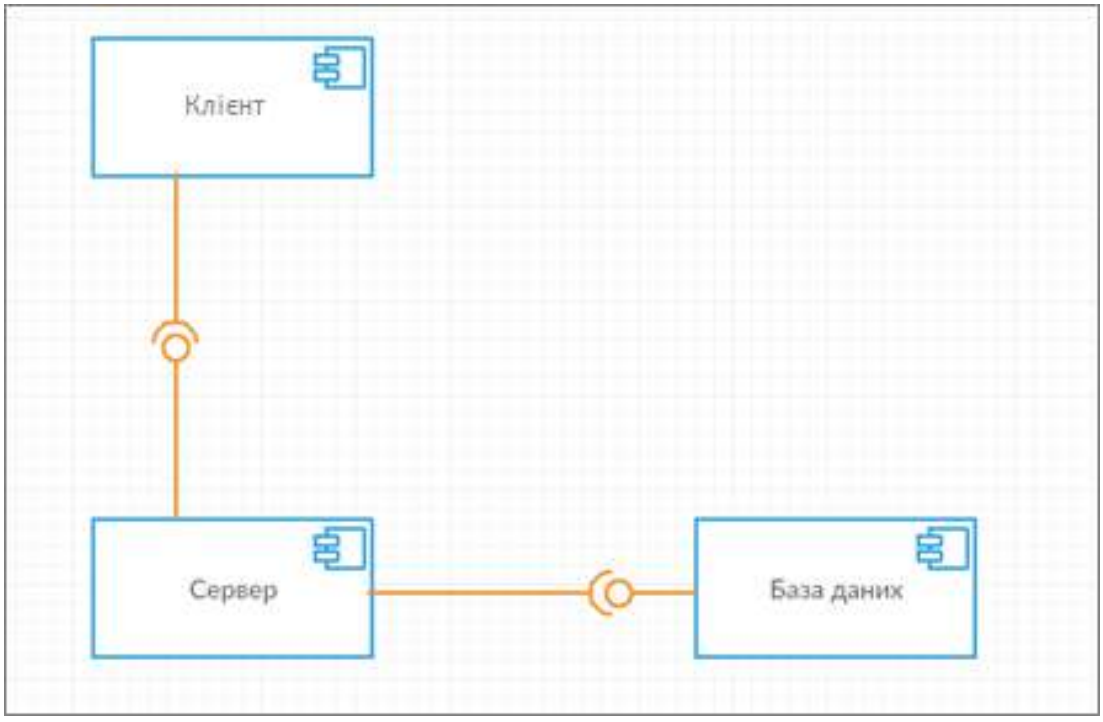
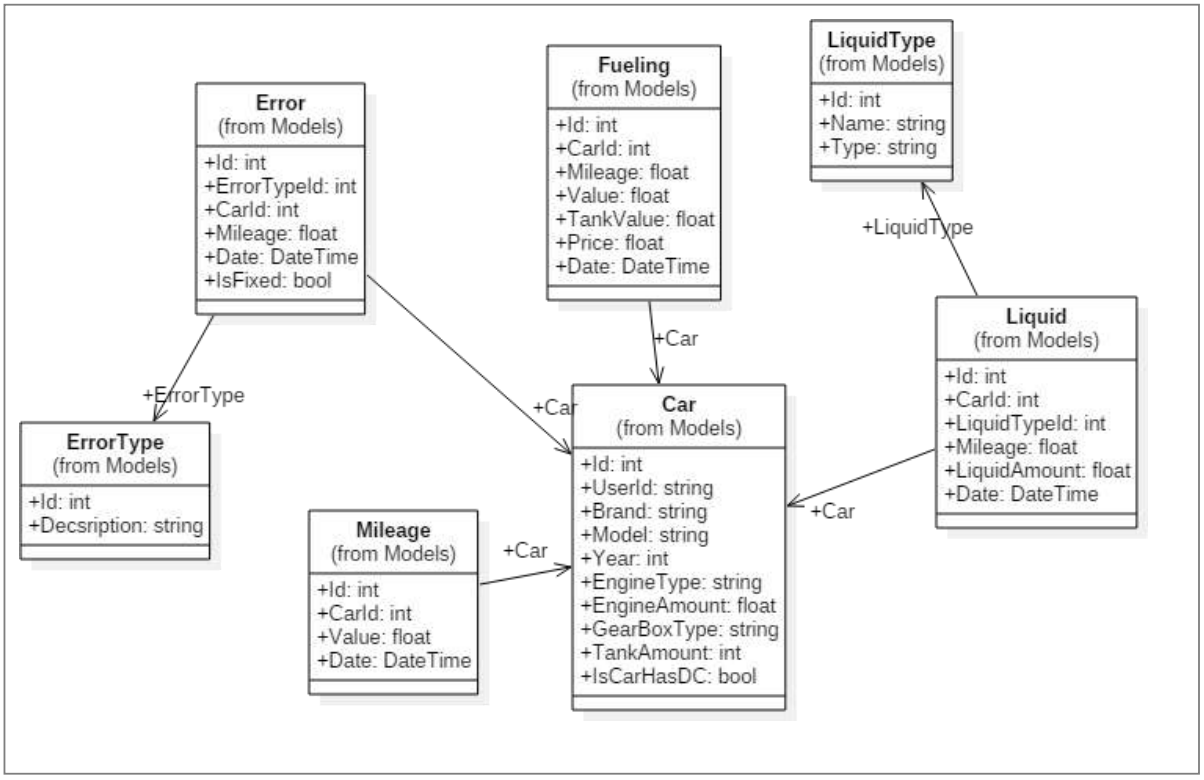


Рисунок 5.1 – Діаграма компонентів

На ній можна побачити, що система має такі основні компоненти, як клієнт, сервер та база даних. Два останніх компонента міститимусь у контейнері [14].

5.1.2 Діаграма класів

Діаграма класів відображає основні сутності системи, вона представлена нарисунку 5.2.



Кожен клас описує певну сутність.

Клас Error представляє сутність помилки.

Клас Fueling представляє сутність заправки автомобіля. Клас Liquid - це сутність, яка описує заміну деякої рідини.

Клас Liquid Type - це допоміжна сутність, яка зберігає тип рідини:Клас Car представляє сутність автомобіля.

Клас Mileage - це сутність яка зберігає дані про пробіг авто в певний момент часу.

5.2 Проектування архітектури ПЗ

Було вирішено використовувати багаторівневу архітектуру додатку розмежувати функціональні рівні та виділити наступні:

а) Data access layer. Він містить:

1) репозиторії які взаємодіють з БД [15];

2) класи моделей даних [16]

3) реалізація паттерну UnitOfWork [17];

б) Business logic layer. Він зберігає у собі:

1) сервіси, які виконують оброку запитів;

2) алгоритми обробки запитів;

3) інфраструктура;

4) виключення;

5) розширення;

6) бізнес сутності;

7) DTO моделі проміжні моделі даних [18];

в) Presentation layer містить:

1) контролери;

2) моделі даних для кінцевих клієнтів;

3) розмітку;

4) скрипти;

5) стилі.

5.3 Проектування бази даних

Було вирішено використовувати реляційну базу даних MsSQL, а також ORMEntity Framework Core для більш зручної та ефективної взаємодії з базою даних.

Під час процесу моделювання було вирішено обрати реляційну БД.

Буде використовується технологія ADO.NET. Зокрема, основою на ньому є Entity Framework Core 2. Це ORM для .NET Framework. Надає можливість взаємодіїз об’єктами за допомогою LINQ. Іншими словами, можна працювати з об’єктами бази даних так само, як і зі звичайними об’єктами C#.

Ця технологія також дозволяє використовувати підхід Code first, який включає автоматичне створення бази даних. Тобто створюється модель спеціального класу яка описує сутність системи. Потім вказуються різні обмеження, ключі та залежності. База даних на основі цих класів буде

створюватися автоматично.

Такий підхід значно полегшує роботу з базою даних та скорочує час розробки.Зрештою, ці моделі будуть використовуватися в майбутньому для роботи з базою даних. Цей підхід також знижує рівень вимог до знань розробників у галузі баз даних [19].

Entity Framework також має вбудований механізм міграції. Вони використовуються при зміні структури бази даних, тобто коли таблиці або стовпці додаються або видаляються, змінюються їх типи тощо. При використанні міграції розробникам потрібно лише виконати кілька команд у консолі. Крім того, усі ці сценарії зберігаються у проекті, і є можливість швидко застосувати зміни з однієї версії бази даних до іншої за допомогою декількох консольних команд, якщо це необхідно.

У БД буде зберігатися така інформація:

− автомобілі у таблиці Cars;

− помилки у таблиці Errors;

− інформація про заправки у таблиці Fuelings;

− інформація про пробіги у таблиці Mileages;

− інформація про заміни рідин у таблиці Oils.

Схема БД має вигляд зображений на рисунку 5.3.

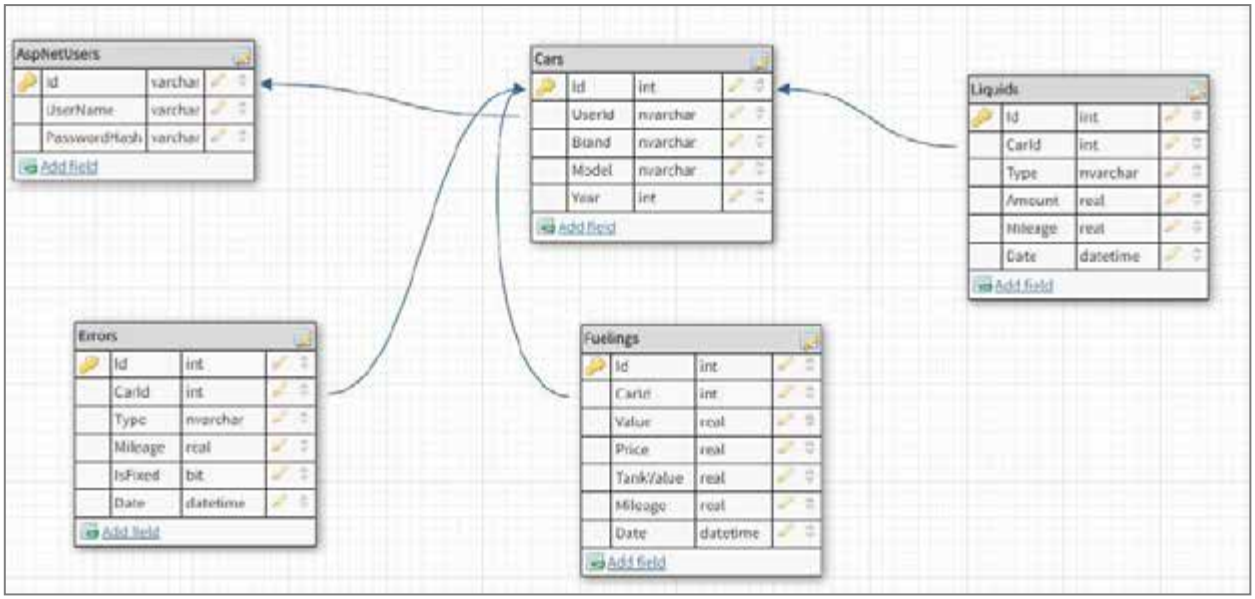


Рисунок 5.3 – Схема бази даних

Також у базі зберігатимуться дані про зареєстрованих користувачів, але ці таблиці будуть автоматично згенеровані бібліотекою ASP.NET Identity, яка відповідає за роботу з користувачами [20].

Це таблиці:

- AspNetRoleClaims;
- AspNetRoles;
- AspNetUser;
- AspNetUserClaims;
- AspNetUserLogins;
- AspNetUserRoles;
- AspNetUserTokens.

ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання кваліфікаційної магістерської роботи було проведено дослідження методів комп'ютерної діагностики за допомогою технології OBD2 длякомп'ютерної діагностики автомобіля. В його ході було розібрано даний стандарт, його історію розвитку, а також особливості використання у програмних системах.

Було досліджено такі методи комп'ютерної діагностики як повне скануваннята діагностування конкретних блоків.

Був виконаний аналіз ринку засобів для мобільної діагностики автомобілів, виявлено переваги та недоліки сучасних засобів.

Досліджено можливості щодо уніфікації системи для взаємодії з різними марками та моделями автомобілів згідно стандарту SAE J1979 який регламентує набір команд для відправлення та обробки запитів до автомобіля через діагностичний порт OBD2.

Порівняно різні моделі сканерів, які на даний момент представлені на ринку та обрано найбільш доступний та підходящий для даної задачі високорівневої діагностики автомобіля.

Розглянуто схему взаємодії з автомобілем за допомогою спеціальних команд, які дозволяють отримати будь-яку інформацію про транспортний засіб у режимі реального часу.

Було виконано планування та розробка макету системи з урахуванням тієї інформацію, що було накопичено під час дослідження.

У результаті була розроблена архітектура програмної системи, що дозволяєкористувачу вести облік витрат на свій автомобіль, контролювати витрати палива,заміни різних автомобільних рідин та діагностувати його за різними параметрами.

Була спроектована база даних для системи та базова система взаємодії між різними рівнями застосування.

Також були розглянуті бібліотеки для розробки під OBD2.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рівень автомобілізації в Україні в 2019 році. URL: https://ecolog-ua.com/news/nova-norma-do-yakoyi-maye-pragnuty-ukrayina-mobilnist-z-nulovymy- vykydamy-dlya-vsih (дата звернення: 04.03.2021).

2. Автомобільна промисловість / Вікіпедія – Україна. URL: http://uk.wikipedia.org/wiki/Автомобільна_промисловість (дата звернення: 05.03.2021).

3. TOAD Pro: Perfect OBD2 Programming Software For Laptop. URL: https://www.obdadvisor.com/toad-pro/ (дата звернення: 10.03.2021).

4. Autoenginuity ScanToolURL: https://www.autoenginuity.com/scantool/ (датазвернення: 20.03.2021).

5. PCMSCAN Features Palmer performance engineering, inc. URL: https://www.palmerperformance.com/products/pcmscan/index.php (дата звернення: 30.03.2021).

6. ProScan All-In-One Computer Aided Scanning Program URL: https://www.proscan.org/ (дата звернення: 01.04.2021).

7. Мастакар, Гаурав. "Експериментальний аналіз безпеки сучасного автомобіля" . Університет Вашингтона та Каліфорнійський університет Сан-Дієго.

8. Marks, Paul . "Гаджет на 25 доларів дозволяє хакерам захопити контроль над автомобілем" . Новий вчений. 2020.

9. Van den Brink, Rob (July 10, 2012). "Dude, Your Car is Pwnd" (PDF). SANSInstitute. Archived from the original (PDF) on February 23, 2015.

10. E / E Diagnostic Test Modes - Equivalent to ISO / DIS 15031-5: April 30, 2002 URL: https://www.sae.org/standards/ (дата звернення: 11.04.2021).

11. Голян В.В., Кравченко О.К. Порівняння моделей життєвих циклів програмного забезпечення з метою виявлення найефективнішого// Збірник наукових праць ХНУ ПС2019р. – 6 с.

12. Viktoriia Skovorodnikova, Vladyslav Apukhtin, Mariya

Shirokopetleva. The Relevance of Using Message Brokers in Robust Enterprise Applications Тези доповіді 2019 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2019 - pp. 305-309

13. Natalia Kravets, Tseshkovskyi N., Liutova K. S. Containers and virtual machinesin microsoft Azure // Science, society, education: topical issues and development prospects. Abstracts of the 7th International scientific and practical conference. SPC “Sci-conf.com.ua”. Kharkiv, Ukraine. 2020. P. 278-281. ЗАЮЗАТЬ

14. Язык программирования C# и .NET / Мова програмування C# та .NET / Metanit. URL: https://metanit.com/shart/general.php (дата звернення: 15.04.2021).

15. SQL: Полное руководство: Пер. с англ. – 2-е изд., перераб. и доп. – К.: Издательская группа BHV, 2016. – 816 с.

16. World’s largest developer community / Найбільша спільнота розробників у світі / Stackoverflow. URL: https://stackoverflow.com/ (дата звернення: 19.04.2021).

17. Шилдт, Г. C# 4.0 Полное руководство / Г. Шилдт. – М.: “Вильямс”, 2017.– 1056 с.

18. Основи систем баз даних / Дж. Д. Ульман - М. : Фінанси і статистика, 2023. – 334 с.

19. Уроки по C# і платформа .NET / Уроки по C# та платформі .NET / ProfessorWeb. URL: http://professorweb.ru/ - 04.06.2019 г. (дата звернення: 28.04.2021).

20. Protocol-Oriented Programming [Електронний ресурс] // Habr. – 2018. – Режим доступу до ресурсу: https://habr.com/ru/post/358804/.

21. Advanced iOS App Architecture (First Edition): Real-world app architecture in Swift / Rene Cacheaux & Josh Berlin, 2018. – Raywenderlich.

22. Data Structures and Algorithms in Swift, 2017. – Raywenderlich.

23. Design Patterns by Tutorials, 2018. – Raywenderlich.

24. Concurrency by Tutorials, 2017. – Raywenderlich.

25. iOS 10 Programming Fundamentals with Swift: Swift, Xcode, and Cocoa Basics / Matt Neuburg, 2017.

26. Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch, 2018.

27. Основы разработки приложений под iOS и macOS / Василий Усов, 2018.

28. Разработка приложений в среде Xcode для iPhone и iPad с использованием iOS SDK / Молли Маскри, 2017.

29. Swift Programming Language / Apple, 2019

30. Human Interface Guidelines / Apple, 2019.

31. 100 Things Every Designer Needs to Know About People / Susan Weinschenk, 2021