

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

**на тему: «Розробки мобільного додатку для моніторингу стану
курця мовою С#»**

Виконав: студент 4 курсу, групи ПД-41

спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Читулян В.О.

(прізвище та ініціали)

Керівник

Негоденко О. В

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Нормоконтроль

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В. Негоденко

« ____ » _____ 2021 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ
ЧИТУЛЯНУ ВАДИМУ ОЛЕГОВИЧУ

1. Тема роботи: «Розробки мобільного додатку для моніторингу стану курця мовою C#»

Керівник роботи Негоденко Олена Василівна, доцент кафедри затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Середовище розробки Microsoft Visual Studio

3.2. Xamarin

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

4.1. Аналіз обов'язків співробітників відділу кадрів

4.2. Аналіз та порівняння існуючих аналогів

- 4.3. Дослідження програмних засобів для розробки додатку
- 4.4. Розробити функціонал створеного додатку
- 5. Перелік графічного матеріалу
- 5.1. Мета, об'єкт, Предмет
- 5.2. Програми аналогів
- 5.3. Техічне завдання
- 5.5. Засоби реалізації
- 5.6. Діаграма користувача
- 5.7. Апробації
- 5.8. Висновки
- 6. Дата видачі завдання 19.04.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 22.04.21	
2	Аналіз існуючих аналогів	22.04.21 – 23.04.21	
3	Дослідження програмних засобів	23.04.21 – 01.05.21	
4	Моделювання об'єкту проектування	01.05.21 – 05.05.21	
5	Розробка функціоналу за стосунку	05.05.21 – 07.05.21	
6	Вступ, висновки, реферат	07.05.21 – 08.05.21	
7	Розробка презентації за стосунку	08.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент

Керівник роботи

РЕФЕРАТ

Текстова частина бакалаврської роботи 50 с., 22 рис., 96 джерел.

Ключові слова: Microsoft Visual Studio, C#, Xamarin.Forms, моніторинг стану курця.

Об'єкт дослідження – покращення процесу моніторингу стану курця.

Предмет дослідження – додаток для людей, що споживають тютюнові вироби або суміші для куріння.

Мета роботи – розробка програмного забезпечення для споживачів тютюнових виробів або сумішів для куріння.

Наукова новизна даної роботи полягає в наступному:

1. Розроблено алгоритм для покращення процесу моніторингу стану курця.
2. Встановлено, що використання мови програмування C# у поєднанні з середовищем розробки Visual Studio є вдалим рішенням для реалізації проекту.
3. Доведено, що мова C# у поєднанні з бібліотекою Xamarin.Forms надає можливість ефективної розробки додатка у порівнянні з іншими мовами програмування, а також робить додаток крос-платформним.
4. Розроблено додаток для покращення процесу моніторингу стану курця.

Упровадження розробленого застосунку допоможе користувачам краще контролювати власний стан, що пришвидшить процес відмови від тютюнових виробів або сумішів для куріння

Під час виконання роботи було проаналізовано існуючі аналоги, визначено їх переваги та недоліки, завдяки чому було сформоване технічне завдання проекту.

Проаналізовано можливості середовища розробки Microsoft Visual Studio. Розроблено технічне завдання, що задовольняють визначені специфікації.

Галузь використання – додаток може використовувати будь-яка людина(курець), яка бажає контролювати власний стан або намагається позбавитися залежності від куріння.

Ключові слова: додаток, залежність, моніторинг стану, C#.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1. Вплив паління на організм людини	11
1.2. Огляд існуючих рішень	19
РОЗДІЛ 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ.....	22
2.1. Вибір мови програмування	22
2.2. Вибір середовища розробки.....	28
2.3. Інструменти Xamarin	32
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ	35
3.1. Діаграма варіантів використання.....	35
3.2. Діаграма класів програмного продукту.....	37
3.3. Графічний інтерфейс користувача.....	44
3.4 Тестування ПЗ.....	47
РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА	50
ВИСНОВКИ	52
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	53

ВСТУП

В сучасному світі все більш явно спостерігається тенденція діджиталізації усіх галузей життя соціуму, починаючи від банальних задач накшатл закупівлі продуктів харчування і закінчуючи серйозними промисловими задачами.

Людство намагається перекласти на апаратне і програмне забезпечення якнайбільшу кількість проблем, які не може, або не має бажання вирішувати власноруч.

Однією з таких проблем кожної третьої людини в соціумі є тютюнопаління. В кожному випадку проблема має свій індивідуальний характер, але їх можна класифікувати і розбити на дві конкретні групи:

- проблема тютюнопаління в цілому (спроби покинути згубну звичку);
- проблема зловживання тютюнопалінням (вживання надмірної кількості тютюнових виробів);
- в даній ситуації під тютюнопалінням підрозуміваються не тільки класичні сигарети, але й різного роду електронні пристрої, такі як Glo, IQOS або Juul;

Виходячи з актуальності проблематики тютюнопаління і тенденції створення програмних засобів для вирішення проблем, об'єктом дослідження є програмний засіб для контролю стану курця.

Предмет дослідження полягає в аналізі методів і засобів для створення програмного засобу з контролю стану курця.

Мета роботи полягає в аналізі методів і засобів для створення програмного засобу з контролю стану курця для подальшого використання отриманих знань для створення власної програмної реалізації.

Для виконання поставленої мети слід виконати наступні завдання:

1. Вровести аналіз предметної області:

1.1. Проаналізувати вплив паління на організм людини.

- 1.2. Провести огляд та аналіз існуючих рішень.
2. Вибрати засоби для реалізації програмного продукту:
 - 2.1. Вибрати мову програмування.
 - 2.2. Обрати середовище розробки.
 - 2.3. Описати інші технології.
3. Провести проектування і розробку програмного продукту:
 - 3.1. Розробити діаграму варіантів використання.
 - 3.2. Розробити діаграму класів.
 - 3.3. Розробити графічний інтерфейс користувача.
4. Написати інструкцію користувача.

Основні методи дослідження: аналіз літературних і інших джерел, комп'ютерне моделювання і проектування, розробка програмного забезпечення.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Вплив паління на організм людини

Куріння - це практика, при якій речовина спалюється, а дим, що утворюється, вдихається для дегустації та всмоктування в кров. Найчастіше використовуваною речовиною є висушені листя рослини тютюну, які були згорнуті у невеликий прямокутник рулонного паперу, щоб створити маленький круглий циліндр, який називають «сигаретою». Куріння в першу чергу практикується як спосіб введення для рекреаційного вживання наркотиків, оскільки згоряння висушеного рослинного листя випаровується і доставляє активні речовини в легені, де вони швидко всмоктуються в кров і потрапляють до тканин організму. У разі куріння сигарет ці речовини містяться в суміші аерозольних частинок і газів і включають фармакологічно активний алкалоїд нікотин; випаровування створює нагрітий аерозоль та газ у формі, що дозволяє вдихати та глибоко проникати в легені, де відбувається всмоктування в кров активних речовин. У деяких культурах куріння також здійснюється як частина різних ритуалів, де учасники використовують це, щоб допомогти викликати транс-подібні стани, які, на їхню думку, можуть призвести їх до духовного просвітлення.

Куріння є однією з найпоширеніших форм рекреаційного вживання наркотиків. Куріння тютюну є найпопулярнішою формою, яку практикують понад мільярд людей у всьому світі, більшість з яких припадає на країни, що розвиваються. [1] До менш поширених наркотиків для паління належать конопля та опій. Деякі речовини класифікуються як важкі наркотики, такі як героїн, але їх використання дуже обмежене, оскільки вони, як правило, не доступні у продажу. Сигарети виготовляються переважно промисловим способом, але їх також можна скрутити вручну із сипучого тютюну та

рулонного паперу. Інші знаряддя для паління включають люльки, сигари, біді, кальяни та бонги.

Куріння може бути датовано ще 5000 р. До н. Е., і це було зафіксовано в багатьох різних культурах по всьому світу. Раннє куріння розвивалось у поєднанні з релігійними церемоніями; як підношення божествам, в ритуалах очищення або для того, щоб дозволити шаманам і жерцям змінити свій розум з метою ворожіння або духовного просвітлення. Після європейських розвідок і завоювання Америки практика куріння тютюну швидко поширилася на решту світу. У таких регіонах, як Індія та Африка на південь від Сахари, він поєднався з існуючою практикою куріння (переважно конопель). В Європі він запровадив новий вид соціальної діяльності та форму споживання наркотиків, яка раніше не була відомою.

Сприйняття навколишнього куріння змінювалось з часом і від місця до місця: святе і грішне, витончене і вульгарне, панацея і смертельна небезпека для здоров'я.

В останнє десятиліття 20 століття на куріння стали сприймати рішуче негативно, особливо в західних країнах. Куріння, як правило, має негативні наслідки для здоров'я, оскільки вдихання диму за своєю суттю створює труднощі для різних фізіологічних процесів, таких як дихання. Куріння тютюну є однією з основних причин багатьох захворювань, таких як рак легенів, інфаркт, ХОЗЛ, еректильна дисфункція та вроджені вади. [1] Доведено, що хвороби, пов'язані з курінням тютюну, вбивають приблизно половину курців, що тривають довго, порівняно із середнім рівнем смертності, з яким стикаються некурці. Куріння спричиняло понад п'ять мільйонів смертей на рік з 1990 по 2015 рік. [2] Небезпека куріння для здоров'я змусила багато країн запровадити високі податки на тютюнові вироби, публікувати рекламу, щоб перешкоджати вживанню, обмежувати рекламу, що сприяє вживанню, та надати допомогу кинути палити тим, хто палить. [1]

Вживання тютюну має переважно негативні наслідки для здоров'я людини, і стурбованість наслідками тютюну для здоров'я має давню історію. Дослідження зосереджені насамперед на курінні тютюнових сигарет [1] [2].

Тютюновий дим містить понад 70 хімічних речовин, що викликають рак. [3] Тютюн також містить нікотин, який є психоактивним наркотиком, що викликає сильну залежність. Коли курять тютюн, нікотин викликає фізичну та психологічну залежність. Сигарети, що продаються в слаборозвинених країнах, як правило, мають більш високий вміст смоли і рідше фільтруються, що потенційно збільшує вразливість до захворювань, пов'язаних з тютюнопалінням, у цих регіонах [4].

Вживання тютюну є єдиною найбільшою причиною смерті, яку можна запобігти, [5] Половина людей, які вживають тютюн, вмирає від ускладнень вживання тютюну. [3] За оцінками Всесвітньої організації охорони здоров'я (ВООЗ), щорічно тютюн спричиняє близько 6 мільйонів смертей (близько 10% всіх смертей), причому 600 000 з них трапляються у некурящих через пасивне паління. [3] [6] За оцінками, у 20 столітті тютюн спричинив 100 мільйонів смертей [3]. Подібним чином, Центри США з контролю та профілактики захворювань описують вживання тютюну як "найважливіший ризик для здоров'я людей у розвинених країнах і важливу причину передчасної смерті у всьому світі" [7]. В даний час кількість передчасних смертей у США від споживання тютюну на рік перевищує кількість працівників, зайнятих у тютюновій промисловості, на 4-1. [8] Згідно з оглядом 2014 року в *New England Journal of Medicine*, при збереженні існуючих моделей куріння в ХХІ столітті тютюн вб'є близько 1 мільярда людей, половина з яких - до 70-річного віку [9].

Вживання тютюну найчастіше призводить до захворювань, що вражають серце, печінку та легені. Куріння є основним фактором ризику інфекцій, таких як пневмонія, інфаркти, інсульти, хронічна обструктивна хвороба легень (ХОЗЛ) (включаючи емфізему та хронічний бронхіт), а також кілька видів раку (зокрема, рак легенів, рак гортані та рота, рак сечового

міхура та рак підшлункової залози). Це також викликає захворювання периферичних артерій та високий кров'яний тиск. Ефекти залежать від кількості років, які людина курить, і від того, скільки людина курить. Початок куріння раніше в житті та куріння сигарет із високим вмістом смоли збільшує ризик цих захворювань. Крім того, було показано, що тютюновий дим у навколишньому середовищі або пасивний дим спричиняє шкідливі наслідки для здоров'я людей різного віку. [10] Вживання тютюну є важливим фактором викиднів серед вагітних курців, і це сприяє ряду інших проблем зі здоров'ям плода, таких як передчасні пологи, низька вага при народженні та збільшує в 1,4-3 рази шанс синдрому раптової дитячої смерті (SIDS)). [11] Частота еректильної дисфункції приблизно на 85 відсотків вища серед курців чоловічої статі порівняно з некурящими [12] [13].

Кілька країн вжили заходів щодо контролю над споживанням тютюну з обмеженнями споживання та продажу, а також попереджувальними повідомленнями, надрукованими на упаковці. Крім того, закони про заборону куріння, що забороняють палити в громадських місцях, таких як робочі місця, театри, бари та ресторани, зменшують вплив пасивного паління та допомагають деяким людям, які палять, кинути палити, не маючи негативних економічних наслідків для ресторанів чи барів [3]. Податки на тютюн, що підвищують ціну, також ефективні, особливо в країнах, що розвиваються. [3]

Ідея про те, що вживання тютюну спричиняє деякі захворювання, включаючи рак порожнини рота, спочатку, наприкінці 1700-х та 1800-х років, була широко прийнята медичною спільнотою [14]. У 1880-х роках автоматизація скоротила вартість сигарет, а їх використання розширилося [15] [16]. Починаючи з 1890-х років, регулярно повідомлялося про асоціації вживання тютюну з раком та судинними захворюваннями; мета-аналіз із посиланням на 167 інших робіт був опублікований в 1930 р. і дійшов висновку, що вживання тютюну спричиняє рак [17] [18]. Протягом 30-х років минулого століття публікувалися дедалі вагоміші дані спостережень, а в 1938

році Science опублікувала статтю, яка показує, що споживачі тютюну живуть значно коротше життя. Дослідження контролю над випадками були опубліковані в нацистській Німеччині в 1939 та 1943 рр., Та одне в Нідерландах у 1948 р., Але широку увагу вперше привернули п'ять досліджень з контролю за випадками, опубліковані в 1950 р. Дослідниками з США та Великобританії. Ці дослідження широко критикували, що вони показують кореляцію, а не причинність. Подальші подальші когортні дослідження на початку 1950-х років чітко виявили, що курці помирають швидше і частіше помирають від раку легенів та серцево-судинних захворювань [14]. Вперше ці результати були широко прийняті в медичній спільноті та опубліковані серед широкої громадськості в середині 1960-х років [14].

Основними ризиками вживання тютюну є багато видів раку, зокрема рак легенів, [61] рак нирок, [62] рак гортані та голови та шиї, [63] [64] рак сечового міхура, [65] рак стравоходу, [66] рак підшлункової залози [67] та рак шлунка. [68] Дослідження встановили взаємозв'язок між тютюновим димом, включаючи пасивне куріння, та раком шийки матки у жінок [69]. Є деякі дані, що свідчать про невеликий підвищений ризик розвитку мієлоїдного лейкозу, [70] плоскоклітинного синоназального раку, раку печінки, раку прямої кишки, раку жовчного міхура, надниркових залоз, тонкої кишки та різних видів раку у дітей [68]. Можливий зв'язок між раком молочної залози та тютюном досі не визначений [71] [потрібне медичне посилання]

Куріння сильно впливає на ризик раку легенів, до 90% випадків спричинене курінням тютюну [72]. Ризик розвитку раку легенів зростає із збільшенням кількості років куріння та кількості викурених сигарет на день [73]. Куріння може бути пов'язане з усіма підтипами раку легенів. Дрібноклітинна карцинома (SCLC) є найбільш тісно пов'язаною з майже 100% випадків, що виникають у курців [74]. Ця форма раку була ідентифікована за допомогою петель аутокринного росту, активації

протоонкогену та інгібування генів супресорів пухлини. SCLC може походити з нейроендокринних клітин, розташованих в бронху, які називаються клітинами Фейртера. [75]

Ризик померти від раку легенів до 85 років становить 22,1% для курця-чоловіка та 11,9% для курця-жінки, за відсутності конкуруючих причин смерті. Відповідні оцінки для некурців протягом усього життя становлять 1,1% ймовірності смерті від раку легенів для чоловіка європейського походження до 85 років та 0,8% ймовірності для жінки [76]

Вважається, що при курінні тривалий вплив сполук, що містяться в димі (наприклад, оксиду вуглецю та ціаніду), є причиною пошкодження легенів та втрати еластичності альвеол (рис. 1.1), що призводить до емфіземи та хронічної обструктивної хвороби легенів (ХОЗЛ). ХОЗЛ, спричинений курінням, - це постійне, невиліковне (часто термінальне) зменшення легеневої здатності, яке характеризується задишкою, хрипами, постійним кашлем з мокротою та пошкодженням легенів, включаючи емфізему та хронічний бронхіт [77]. Канцерогенний акролеїн та його похідні також сприяють хронічному запаленню, яке є при ХОЗЛ. [78]

Вдихання тютюнового диму викликає кілька негайних реакцій у серці та судинах. Протягом однієї хвилини частота серцевих скорочень починає зростати, збільшуючись на цілих 30 відсотків протягом перших 10 хвилин куріння. Окис вуглецю в тютюновому димі чинить негативні ефекти, зменшуючи здатність крові переносити кисень. [79]

Куріння також збільшує ймовірність серцевих захворювань, інсульту, атеросклерозу та захворювань периферичних судин. [80] [81] Кілька інгредієнтів тютюну призводять до звуження судин, збільшуючи ймовірність закупорки, а отже, і серцевого нападу або інсульту. Згідно з дослідженням, проведеним міжнародною групою дослідників, у людей до 40 років у п'ять разів частіше трапляється серцевий напад, якщо вони курять [82] [83].

Відомо, що вплив тютюнового диму посилює окислювальний стрес в організмі за допомогою різних механізмів, включаючи виснаження плазмових антиоксидантів, таких як вітамін С. [84]

Недавні дослідження американських біологів показали, що сигаретний дим також впливає на процес поділу клітин серцевого м'яза і змінює форму серця [85].

Вживання тютюну також пов'язане з хворобою Бургера (облітрованний тромбоангіїт), гострим запаленням і тромбозом (згортанням) артерій та вен рук і ніг [86].

Хоча куріння сигарет викликає більший ризик розвитку раку, ніж куріння сигар, курці сигар все ще мають підвищений ризик багатьох проблем зі здоров'ям, включаючи рак, у порівнянні з некурящими [87] [88]. Що стосується пасивного куріння, дослідження NIN вказує на велику кількість диму, що утворюється однією сигарою, кажучи: "сигари можуть внести значну кількість тютюнового диму в приміщення; і, коли велика кількість курців сигар скупчуються в сигарі У випадку, кількість виробленого ETS (тобто пасивного паління) є достатнім, щоб загрожувати здоров'ю тих, хто регулярно потребує роботи в таких умовах "[89].

Куріння має тенденцію до підвищення рівня холестерину в крові. Крім того, співвідношення ліпопротеїдів високої щільності (ЛПВЩ, також відомий як "хороший" холестерин) до ліпопротеїдів низької щільності (ЛПНЩ, також відомий як "поганий" холестерин), як правило, є нижчим у курців порівняно з некурящими. Куріння також підвищує рівень фібриногену і збільшує вироблення тромбоцитів (як тих, що беруть участь у згортанні крові), що робить кров густішою і з більшою ймовірністю згортається. Оксид вуглецю пов'язується з гемоглобіном (кисненосним компонентом еритроцитів), в результаті чого набагато стабільніший комплекс, ніж гемоглобін, зв'язаний з киснем або вуглекислим газом - результатом є постійна втрата функціональності клітин крові. Клітини крові природним чином переробляються через певний проміжок часу, що дозволяє створити

нові, функціональні еритроцити. Однак, якщо вплив окису вуглецю досягає певної точки перед тим, як їх можна переробити, настає гіпоксія (а пізніше і смерть). Усі ці фактори роблять курців більш ризикованими для розвитку різних форм атеросклерозу (зміцнення артерій). У міру прогресування атеросклерозу кров тече менш жорстко через жорсткі та звужені кровоносні судини, завдяки чому кров частіше утворює тромбоз (згусток). Раптова закупорка судини може призвести до інфаркту (інсульт або інфаркт). Однак варто також зазначити, що вплив куріння на серце може бути більш слабким. Ці стани можуть розвиватися поступово, враховуючи цикл загоєння куріння (організм людини зцілюється між періодами куріння), і тому у курця можуть розвиватися менш значні розлади, такі як погіршення або підтримка неприємних дерматологічних станів, наприклад екзема, через зменшення кровопостачання. Куріння також підвищує кров'яний тиск і послаблює судини. [90]

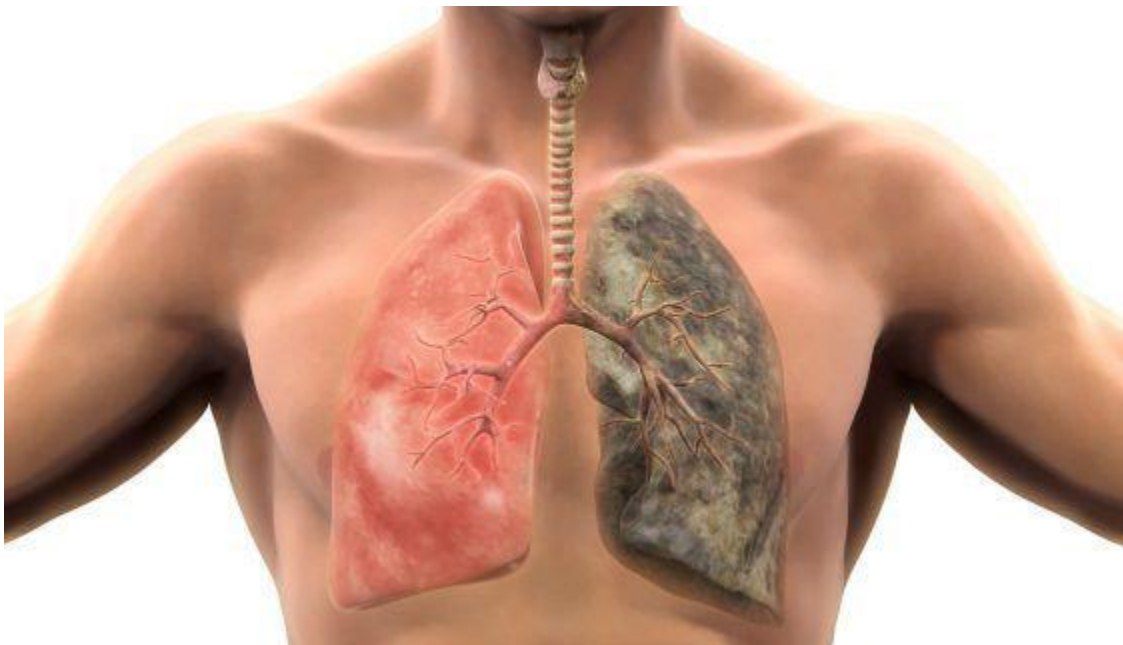


Рис. 1.1 — Порівняння легень курця та здорової людини

1.2. Огляд існуючих рішень

Додаток Kwit допомагає кинути палити в формі гри. У міру прогресу, користувач відкриває досягнення і нові рівні. Програма показує, скільки грошей на сигаретах зекономлено, який зміст чадного газу в крові і на скільки збільшилася тривалість життя. За спеціальні уроки потрібно платити додатково.



Рис. 1.2 — Інтерфейс Kwit

Програма «Не курю» дозволяє відстежувати зекономлені гроші, кількість невикурених сигарет, чистоту легких, стан організму. Прогрес в досягненні мети можна порівнювати з іншими користувачами програми. Також «Не палю» щодня надсилає поради та мотиватори від фахівців. Головне в цьому додатку - позитивний підхід до кидання курити. Крім того, додаток за 2 тижні підготує вас до поступової відмови від шкідливої звички.

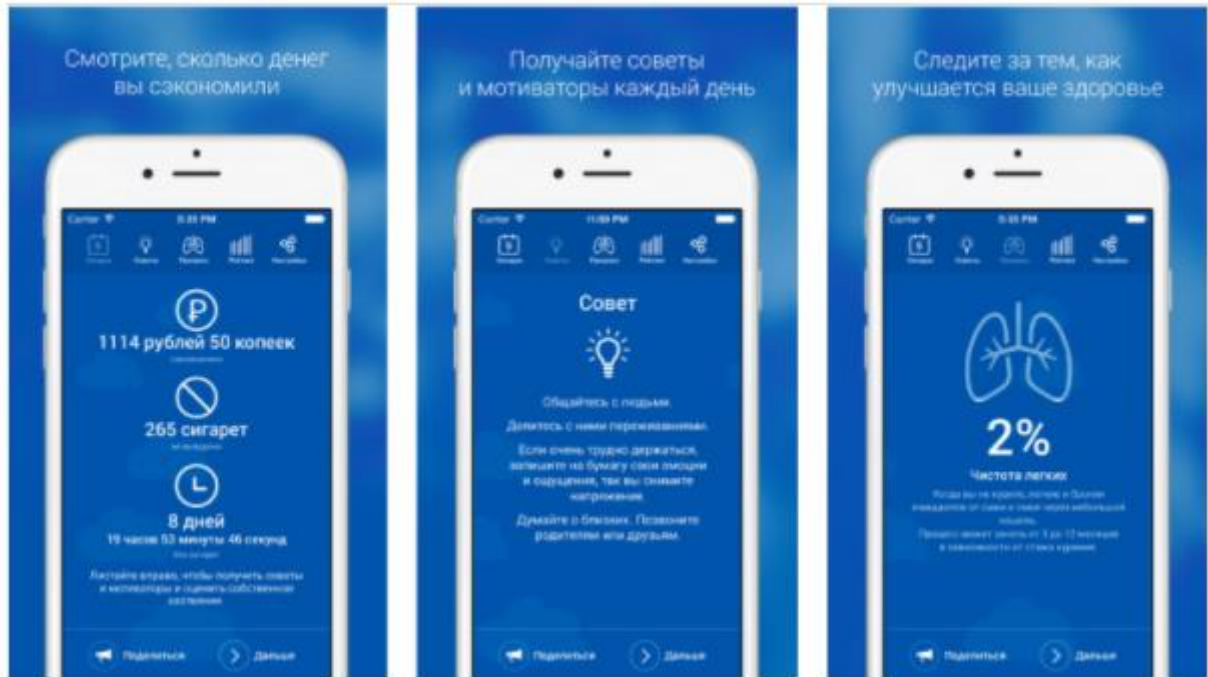


Рис. 1.3 — Интерфейс «Не курю»

«Моя остання сигарета» на графіках показує стан кровообігу, здоров'я легенів, рівня чадного газу, тягу до паління і рівень нікотину в організмі. Програма показує статистику померлих від нікотинової залежності в світі, відсоток ризику розвитку серцево-судинних захворювань і раку легенів.

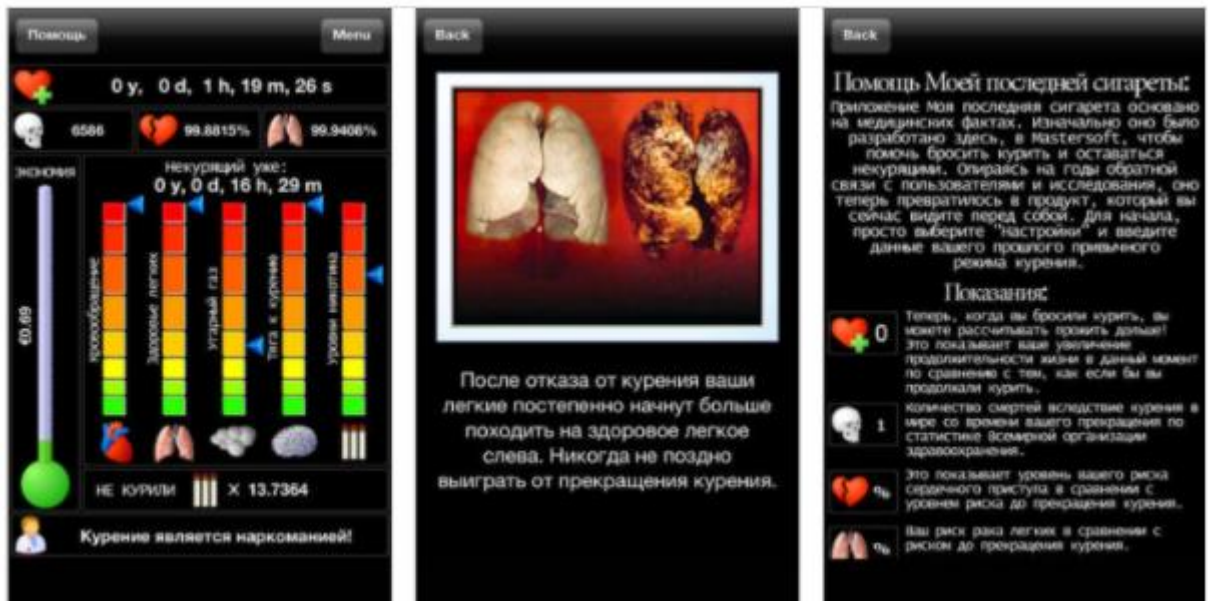


Рис. 1.4 — Интерфейс додатку «Моя последняя сигарета»

Додаток «Gotta Smoke?» буде корисний тим, хто кидає курити поступово. Потрібно відзначати момент куріння, щоб додаток вирахувало часовий інтервал до наступної перекуру. Додаток допомагає знизити кількість викурених сигарет, поступово збільшуючи проміжок між димом. Розробники обіцяють, що через 30 днів використання користувач кине палити зовсім.



Рис. 1.5 — Интерфейс додатку Gotta Smoke?

РОЗДІЛ 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Вибір мови програмування

C # (вимовляється як музична нота C #, але написаний зі знаком числа) - це універсальна багатопарадигмальна мова програмування, що включає строгу типізацію, лексичну область видимості, імператив, декларативний, функціональний, універсальний об'єкт -орієнтовані (на основі класів) та компонентно-орієнтовані дисципліни програмування. Він був розроблений Microsoft приблизно в 2000 році в рамках ініціативи .NET, а потім затверджений в якості міжнародного стандарту Ecma (ECMA-334) і ISO (ISO / IEC 23270: 2018). Mono - це назва безкоштовного проекту з відкритим вихідним кодом для розробки компілятора і середовища виконання для мови. C # є одним з мов програмування, розроблених для Common Language Infrastructure (CLI) (рис. 2.1).

C # був розроблений Андерсом Хейлсбергом, а його командою розробників в даний час керує Мадс Торгерсен. Остання версія - 8.0, випущена в 2019 році разом з Visual Studio 2019 версії 16.3.

Під час розробки .NET Framework бібліотеки класів спочатку були написані з використанням компілятора системи керованого коду під назвою «Simple Managed C» (SMC). У січні 1999 року Андерс Хейлсберг сформував команду для створення нової мови під назвою Cool, який позначав «С-подібна об'єктно-орієнтована мова». Microsoft розглядала збереження назви «Cool» в якості остаточної назви мови, але вирішила не робити цього з причин, пов'язаних з товарними знаками. До того часу, коли проект .NET був публічно оголошено на конференції професійних розробників в липні 2000 року, мову був перейменований в C #, а бібліотеки класів і середовище виконання ASP.NET були портований на C #.

Хейлсберг - головний дизайнер і головний архітектор C # в Microsoft, раніше займався проектуванням Turbo Pascal, Embarcadero Delphi (раніше CodeGear Delphi, Inprise Delphi і Borland Delphi) і Visual J ++. У своїх інтерв'ю і технічних статтях він заявив, що недоліки в більшості основних мов програмування (наприклад, C ++, Java, Delphi і Smalltalk) призвели до основ Common Language Runtime (CLR), що, в свою чергу, призвело до розробки сама мова C #.

З часу випуску C # 2.0 в листопаді 2005 року мови C # і Java розвивалися за все більш і більш розбіжним траєкторіях, ставши двома зовсім різними мовами. Одним з перших основних відхилень стало додавання узагальнень до обох мов з абсолютно різними реалізаціями. C # використовує *reflection* для надання «першокласних» універсальних об'єктів, які можна використовувати як будь-який інший клас, з генерацією коду, що виконується під час завантаження класу. Крім того, в C #, додано кілька основних функцій для програмування функціонального стилю, кульмінацією яких є розширення LINQ, випущені в C # 3.0, і підтримуюча його структура лямбда-виразів, методів розширення і анонімних типів. Ці функції дозволяють програмістам C # використовувати методи функціонального програмування, такі як замикання, коли це вигідно для їх застосування. Розширення LINQ і функціональний імпорт допомагають розробникам скоротити обсяг стандартного коду, який включається в загальні завдання, такі як запити до бази даних, аналіз файлу XML або пошук в структурі даних, зміщуючи акцент на реальну логіку програми, щоб поліпшити читаність і ремонтпридатність.

По своєму дизайну C # є мовою програмування, який безпосередньо відображає основну інфраструктуру спільної мови (CLI). Більшість його внутрішніх типів відповідають типам значень, реалізованих середовищем CLI. Однак специфікація мови не містить вимог до генерації коду компілятора, тобто не вказує, що компілятор C # має орієнтуватися на середовище виконання спільної мови, або генерувати загальний проміжний

мова (CIL), або генерувати будь-який інший конкретний формат. Теоретично, компілятор C # може генерувати машинний код, як традиційні компілятори C ++ та Fortran.

C # підтримує строго типізовані оголошення неявних змінних з ключовим словом var, а також неявно типізовані масиви з ключовим словом new [], за яким слід ініціалізатор колекції.

C # підтримує суворий логічний тип даних, bool. Оператори, які приймають умови, такі як while і if, вимагають вирази типу, що реалізує оператор true, такого як логічний тип. Хоча C ++ також має логічний тип, він може вільно перетворюватися в цілі і з цілих чисел, а вирази, наприклад, якщо (a) вимагають тільки, щоб a був перетворений в bool, дозволяючи a бути int або покажчиком. C # забороняє цей підхід «цілочисельне значення істина або брехня» на тій підставі, що примус програмістів використовувати вирази, які повертають саме bool, може запобігти певні типи помилок програмування, таких як if (a = b) (використання присвоювання = замість рівності ==).

C # більш безпечний для типів, ніж C ++. Єдиними неявними перетвореннями за замовчуванням є ті, які вважаються безпечними, наприклад, розширення цілих чисел. Це застосовується під час компіляції, під час JIT і, в деяких випадках, під час виконання. Не відбувається неявного перетворення між логічними значеннями і цілими числами, а також між членами перерахування і цілими числами (за винятком літерала 0, який може бути неявно перетворений в будь-який перераховується тип). Будь кероване перетворення повинне бути чітко позначено як явне або неявне, на відміну від конструкторів копіювання C ++ та операторів перетворення, які за замовчуванням неявні.

C # має явну підтримку коваріації і контраваріантності в універсальних типах, на відміну від C ++, який має деяку ступінь підтримки контраваріантності просто завдяки семантиці повертаються типів віртуальних методи.

Учасники перерахування розміщуються у своїй області видимості.

Мова C # не допускає глобальних змінних або функцій. Всі методи і члени повинні бути оголошені всередині класів. Статичні члени відкритих класів можуть замінювати глобальні змінні і функції.

Локальні змінні не можуть приховувати змінні вміщує блоку, на відміну від C та C ++.

C # має підтримку строго типізованих покажчиків на функції через ключове слово делегат. Як і псевдо-C ++ фреймворк Qt, в C # є семантика, зокрема навколишнє події стилю публікації-підписки, хоча C # використовує для цього делегати.

Керована пам'ять не може бути звільнена; замість цього він автоматично збирається. Збірка сміття вирішує проблему витоків пам'яті, позбавляючи програміста від відповідальності за звільнення пам'яті, яка більше не потрібна.

На відміну від C ++, C # не підтримує множинне успадкування, хоча клас може реалізовувати будь-яку кількість інтерфейсів. Це було дизайнерське рішення провідного архітектора мови, щоб уникнути ускладнення і спростити архітектурні вимоги у всьому CLI. При реалізації декількох інтерфейсів, які містять метод з однаковою сигнатурою, і. тобто два методу з одним і тим же ім'ям, що приймають параметри одного й того ж типу в одному і тому ж порядку, C # дозволяє реалізовувати кожен метод в залежності від того, через який інтерфейс викликається цей метод, або, як Java, дозволяє реалізувати метод один раз і мати один виклик за викликом через будь-який з інтерфейсів класу.

Однак, на відміну від Java, C # підтримує перевантаження операторів. Тільки найбільш часто перевантажені оператори в C ++ можуть бути перевантажені в C #.

C # має можливість використовувати LINQ через .NET Framework. Розробник може запросити будь-який об'єкт IEnumerable <T>, документи XML, набір даних ADO.NET і бази даних SQL. [60] Використання LINQ в C

дає такі переваги, як підтримка Intellisense, потужні можливості фільтрації, безпека типів з можливістю перевірки помилок компіляції і узгодженість даних для запиту з різних джерел. Є кілька різних мовних структур, які можна використовувати з C # з LINQ, і вони є виразами запитів, лямбда-виразами, анонімними типами, неявно типізованими змінними, методами розширення і ініціалізаторами об'єктів

У серпні 2001 року корпорації Microsoft, Hewlett-Packard і Intel Corporation виступили співавторами в уявленні специфікації для C #, а також інфраструктури спільної мови (CLI) в організацію по стандартизації Ecma International. У грудні 2001 року ECMA випустила специфікацію мови ECMA-334 C #. C # став стандартом ISO у 2003 році (ISO / ІЕС 23270: 2003 Інформаційні технології. Мови програмування - C #). ECMA раніше прийняла еквівалентні специфікації як друге видання C # в грудні 2002 року.

У червні 2005 року ECMA схвалив видання 3 специфікації C # і оновило ECMA-334. Доповнення включали в себе часткові класи, анонімні методи, нульові типи і універсальні шаблони (щось схоже на шаблони C ++).

У липні 2005 року ECMA представила в ІСО / МЕК JTC 1 за допомогою прискореного процесу останнього стандарту та відповідні ТЗ. Цей процес зазвичай займає 6-9 місяців.

Визначення мови C # і CLI стандартизовані у відповідності зі стандартами ISO і Ecma, які забезпечують розумну та недискримінаційну ліцензійну захист від патентних претензій.

Microsoft погодилася не пред'являти позов розробникам програмного забезпечення з відкритим вихідним кодом за порушення патентів в некомерційних проектах в частині структури, охопленої OSP. Microsoft також погодилася не застосовувати патенти на продукти Novell щодо платять клієнтів Novell, за винятком списку продуктів, в яких явно не згадується C#.NET або реалізація Novell .NET (The Mono Project). Тим не менш, Novell стверджує, що Mono не порушує жодних патентів Microsoft. Microsoft також уклала конкретну угоду про відмову у захисту патентних прав, пов'язаних з

плагіном браузера Moonlight, який залежить від Mono, якщо він отриманий через Novell

Microsoft очолює розробку еталонного компілятора C # з відкритим вихідним кодом і набору інструментів, що раніше носили кодова назва "Roslyn". Компілятор, який повністю написаний на керованому коді (C #), був відкритий, а функціональність відображена як API. Це дозволяє розробникам створювати інструменти рефакторінгу та діагностики.

Інші компілятори C # (деякі з яких включають реалізацію інфраструктури спільної мови і бібліотек класів .NET):

- Проект Mono надає компілятор C # з відкритим вихідним кодом, повну реалізацію загальномовної інфраструктури з відкритим вихідним кодом, включаючи необхідні бібліотеки інфраструктури, як вони зазначені в специфікації ECMA, і майже повну реалізацію пропрієтарних бібліотек класів Microsoft .NET до .NET 3.5. Починаючи з Mono 2.6, планів по впровадженню WPF не існує; WF планується до більш пізнього випуску; і є лише часткові реалізації LINQ to SQL і WCF.
- Проект DotGNU (в даний час припинений) також надає компілятор C # з відкритим вихідним кодом, майже повну реалізацію інфраструктури спільної мови, включаючи необхідні бібліотеки інфраструктури, як вони зазначені в специфікації ECMA, і підмножина деяких залишилися пропрієтарних класів Microsoft .NET. бібліотеки .NET 2.0 (не документовані або не включені в специфікацію ECMA, але включені в стандартний дистрибутив Microsoft .NET Framework). • Загальна мовна інфраструктура загального ресурсу Microsoft під кодовою назвою «Rotor» забезпечує реалізацію спільного джерела середовища CLR і компілятора C #, ліцензованого тільки для освітніх і дослідницьких цілей, а також підмножина необхідних бібліотек інфраструктури

Common Language Infrastructure в специфікації ECMA (вгору в C # 2.0 і підтримується тільки в Windows XP).

C # була обрана основною мовою розробки даного проекту з огляду на усе вищеприписане, а саме — на функціонал, який підтримує дана мова, операційні системи, які підтримуються та простота вивчення.

Загалом — функціонал C # найкраще підходить під дану розробку, за рахунок простої реалізації основних принципів ООП, простоти розробки форм та зрозумілого інтерфейсу.

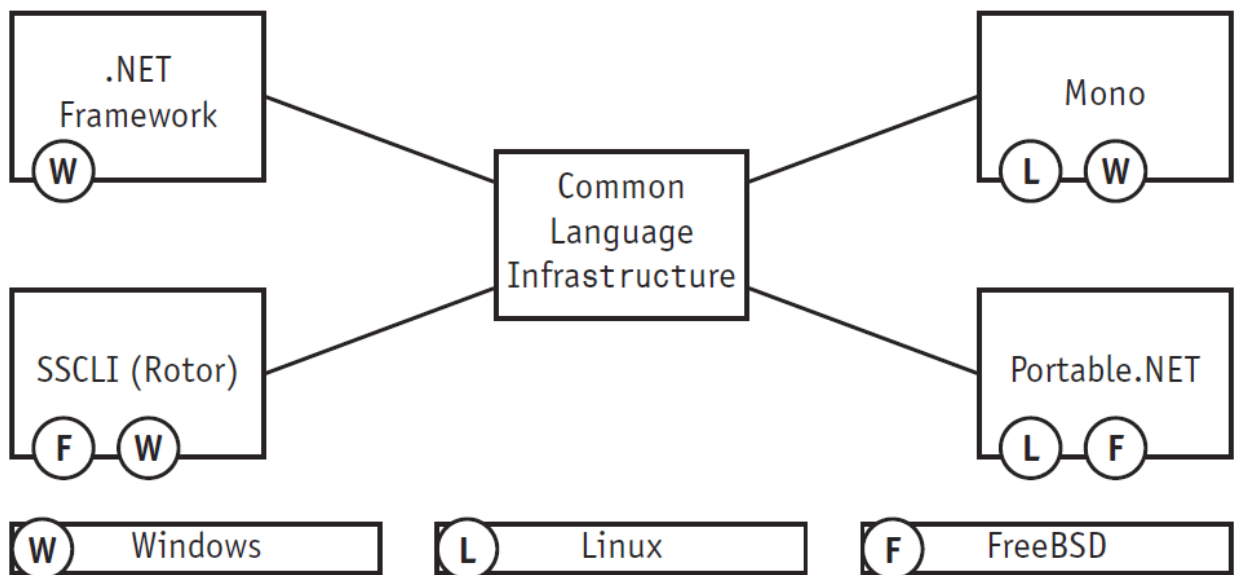


Рис. 2.1 – Структура Common Language Infrastructure

2.2. Вибір середовища розробки

Microsoft Visual Studio - це інтегроване середовище розробки (IDE) від Microsoft. Він використовується для розробки комп'ютерних програм, а також веб-сайтів, веб-додатків, веб-служб та мобільних додатків. Visual Studio використовує платформи Microsoft для розробки програмного забезпечення, такі як API Windows, Windows Forms, Foundation Presentation Foundation, Windows Store та Microsoft Silverlight. Він може створювати як власний код, так і керований код.

Visual Studio включає редактор коду, що підтримує IntelliSense (компонент доповнення коду), а також рефакторинг коду. Інтегрований налагоджувач працює як налагоджувач на рівні джерела, так і налагоджувач на рівні машини. Інші вбудовані інструменти включають кодовий профілер, конструктор для побудови програм GUI, веб-дизайнер, дизайнер класів та дизайнер схем бази даних. Він приймає плагіни, які покращують функціональність майже на кожному рівні - включаючи додавання підтримки для систем керування джерелами (наприклад, Subversion і Git) та додавання нових наборів інструментів, таких як редактори та візуальні дизайнери для мов, що задаються доменом або набори інструментів для інших аспектів розробки програмного забезпечення життєвий цикл (як клієнт Azure DevOps: Team Explorer).

Visual Studio підтримує 36 різних мов програмування та дозволяє редактору коду та налагоджувачу підтримувати (в різній мірі) майже будь-яку мову програмування за умови існування послуги, що залежить від мови. Вбудовані мови включають C, C ++, C ++ / CLI, Visual Basic .NET, C #, F #, JavaScript, TypeScript, XML, XSLT, HTML та CSS. Підтримка інших мов, таких як Python, Ruby, Node.js та M серед інших, доступна через плагіни. Java (і J #) підтримувалися в минулому.

Найбільш основне видання Visual Studio, спільноти, доступне безкоштовно. Гасло для видання Visual Studio Community - "Безкоштовне повнофункціональне IDE для студентів, відкритих джерел та індивідуальних розробників".

На даний момент підтримується версія Visual Studio - 2019 рік.

Visual Studio не підтримує жодної мови програмування, рішення чи інструменту внутрішньо; натомість це дозволяє підключати функціональність, кодовану як VSPackage. Після встановлення функціональність доступна як Сервіс. IDE надає три послуги: SVsSolution, яка надає можливість перераховувати проекти та рішення; SVsUIShell, який забезпечує функцію вікон та інтерфейсу користувача (включаючи вкладки,

панелі інструментів та вікна інструментів); та SVsShell, яка займається реєстрацією VSPackages. Крім того, IDE також відповідає за координацію та забезпечення зв'язку між службами. Всі редактори, дизайнери, типи проектів та інші інструменти реалізовані як VSPackages. Visual Studio використовує COM для доступу до VSPackages. SDK Visual Studio також включає в себе керовану рамку пакетів (MPF), яка є набором керованих обгортків навколо COM-інтерфейсів, які дозволяють писати пакети будь-якою мовою, сумісною з CLI. Однак MPF не забезпечує всю функціональність, що піддається інтерфейсам Visual Studio COM. Потім послуги можуть бути використані для створення інших пакетів, які додають функціональність Visual Studio IDE.

Підтримка мов програмування додається за допомогою спеціального VSPackage, який називається Language Service. Мовна служба визначає різні інтерфейси, які може реалізувати реалізація VSPackage, щоб додати підтримку різних функціональних можливостей. Функції, які можна додати таким чином, включають забарвлення синтаксису, завершення оператора, відповідність дужок, підказки інформації про параметри, списки членів та маркери помилок для складання фону. Якщо інтерфейс буде реалізований, функціонал буде доступний для мови. Мовні послуги реалізуються на мовній основі. Реалізації можуть повторно використовувати код з аналізатора або компілятора для мови. Мовні сервіси можуть бути реалізовані як у рідному коді, так і керованому коді. Для нативного коду можуть бути використані або рідні інтерфейси COM, або Babel Framework (частина Visual Studio SDK). Для керованого коду MPF включає обгортки для написання сервісів керованої мови.

Visual Studio не включає вбудовану підтримку управління джерелами, але вона визначає два альтернативних способи інтеграції систем управління джерелами з IDE. VSPackage управління джерелом може забезпечити власний індивідуальний інтерфейс користувача. Навпаки, плагін управління джерелом за допомогою MSSCCI (Microsoft Source Code Control Interface) надає набір функцій, які використовуються для реалізації різних функцій

управління джерелом, зі стандартним інтерфейсом користувача Visual Studio. MSSCCI вперше використовувався для інтеграції Visual SourceSafe з Visual Studio 6.0, але згодом був відкритий через SDK Visual Studio. Visual Studio .NET 2002 використовував MSSCCI 1.1, а Visual Studio .NET 2003 використовували MSSCCI 1.2. Visual Studio 2005, 2008 та 2010 використовують MSSCCI версії 1.3, яка додає підтримку перейменування та видалення розповсюдження, а також асинхронного відкриття.

Visual Studio підтримує запуск декількох екземплярів середовища (кожен зі своїм набором VSPackages). Екземпляри використовують різні вулики реєстру (див. Визначення MSDN терміна "вулик реєстру" у сенсі, що використовується тут) для зберігання конфігураційного стану та диференціюються їх AppId (ідентифікатор програми). Екземпляри запускаються специфічним для AppId .exe, який вибирає AppId, встановлює кореневий вулик та запускає IDE. VSP-пакети, зареєстровані для одного AppId, інтегруються з іншими VSP-пакетами для цього AppId. Різні випуски продуктів Visual Studio створені за допомогою різних додатків. Продукти випуску Visual Studio Express встановлюються разом із власними AppIds, але продукти Standard, Professional та Team Suite мають однаковий AppId. Отже, видання Express можна встановити поряд з іншими виданнями, на відміну від інших видань, які оновлюють ту саму установку. Професійне видання включає в себе набір VSPackages у стандартному виданні, а командний набір включає суперсет VSPackages в обох інших виданнях. Система AppId використовується Visual Studio Shell в Visual Studio 2008.

Дане середовище розробки було обрано по декількох критеріях:

- підтримка мови C#;
- інтуїтивно зрозумілий інтерфейс;
- можливості графічного редактору, підходящі до задач;

Так як усі критерії задовільнено — середовищем розробки було обрано саме Visual Studio 2019 (рис 2.2).

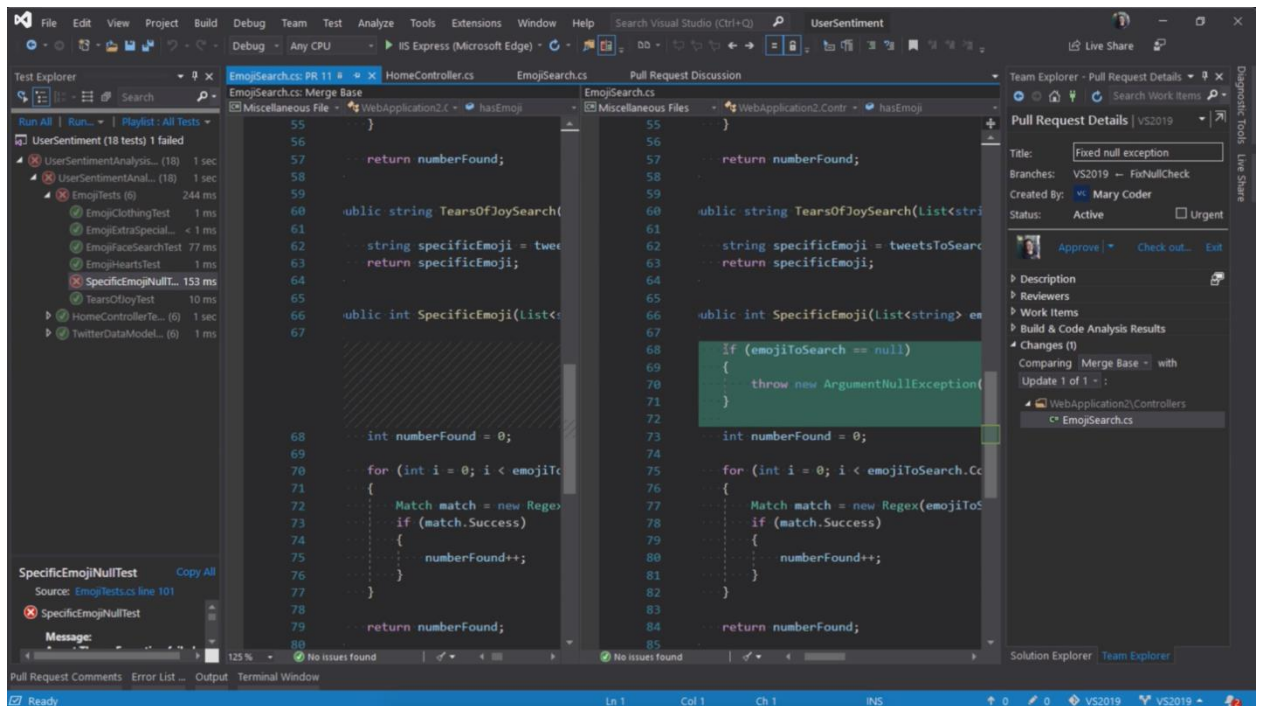


Рис. 2.2 — Інтерфейс середовище розробки Visual Studio 2019

2.3. Інструменти Xamarin

Xamarin - корпорація програмного забезпечення, що належить Microsoft у Сан-Франциско, заснована в травні 2011 року інженерами, що створили Mono, Xamarin.Android (раніше Mono для Android) та Xamarin.iOS (раніше MonoTouch), які є перехресними -реалізація платформ Спільної мовної інфраструктури (CLI) та Загальномовних специфікацій (часто звана Microsoft .NET)

Завдяки спільній кодовій базі C # розробники можуть використовувати інструменти Xamarin для написання власних програм для Android, iOS та Windows із власними користувальницькими інтерфейсами та спільного використання коду на багатьох платформах, включаючи Windows, macOS та Linux. За даними Xamarin, станом на квітень 2017 року понад 1,4 мільйона розробників використовували продукти Xamarin у 120 країнах світу.

24 лютого 2016 року Microsoft оголосила, що підписала остаточну угоду про придбання Xamarin.

Введено в Xamarin 3 28 травня 2014 року і дозволяє використовувати підмножини переносних елементів керування, які відображаються у власних елементах управління Android, iOS та Windows Phone. Windows Phone було припинено та вилучено на користь UWP.

Також можна орієнтуватись на інші різні платформи, такі як Tizen (від Samsung), GTK (Linux), WPF та macOS, навіть якщо вони залишились у Preview.

Ця система використовує XAML. Microsoft модифікувала цей фреймворк для роботи з універсальною платформою Windows.

Microsoft дозволяє розробляти власні мобільні пристрої за допомогою Blazor. Mobile Blazor Bindings дозволяють розробникам створювати власні Android та iOS за допомогою шаблонів програмування C #, .NET та веб.

Xamarin – це платформа для розробників .NET, що складається з інструментів, мов програмування та бібліотек для побудови багатьох різних типів програм. Xamarin постачає надбудови до Microsoft Visual Studio, що дозволяє розробникам створювати додатки для Android, iOS та Windows в середовищі IDE за допомогою завершення коду та IntelliSense. Xamarin для Visual Studio також має розширення, які забезпечують підтримку побудови, розгортання та налагодження програм на симуляторі або пристрої. Наприкінці 2013 року Xamarin та Microsoft оголосили про партнерство, яке включало подальшу технічну інтеграцію та програми для клієнтів, щоб їх спільні бази розробників могли будувати для всіх мобільних платформ. Крім того, Xamarin тепер включає підтримку портативних бібліотек класу Microsoft та більшість функцій C # 5.0, таких як async / await. Генеральний директор і співзасновник Xamarin Нат Фрідман оголосив про альянс під час запуску Visual Studio 2013 у Нью-Йорку.

Xamarin корисний при розробці програм для iOS та Android.

31 березня 2016 року Microsoft оголосила, що об'єднає все програмне забезпечення Xamarin з усіма версіями Microsoft Visual Studio, включаючи Visual Studio Community, і це додало різні функції Xamarin, які будуть попередньо встановлені у Visual Studio, такі як емулятор iOS.

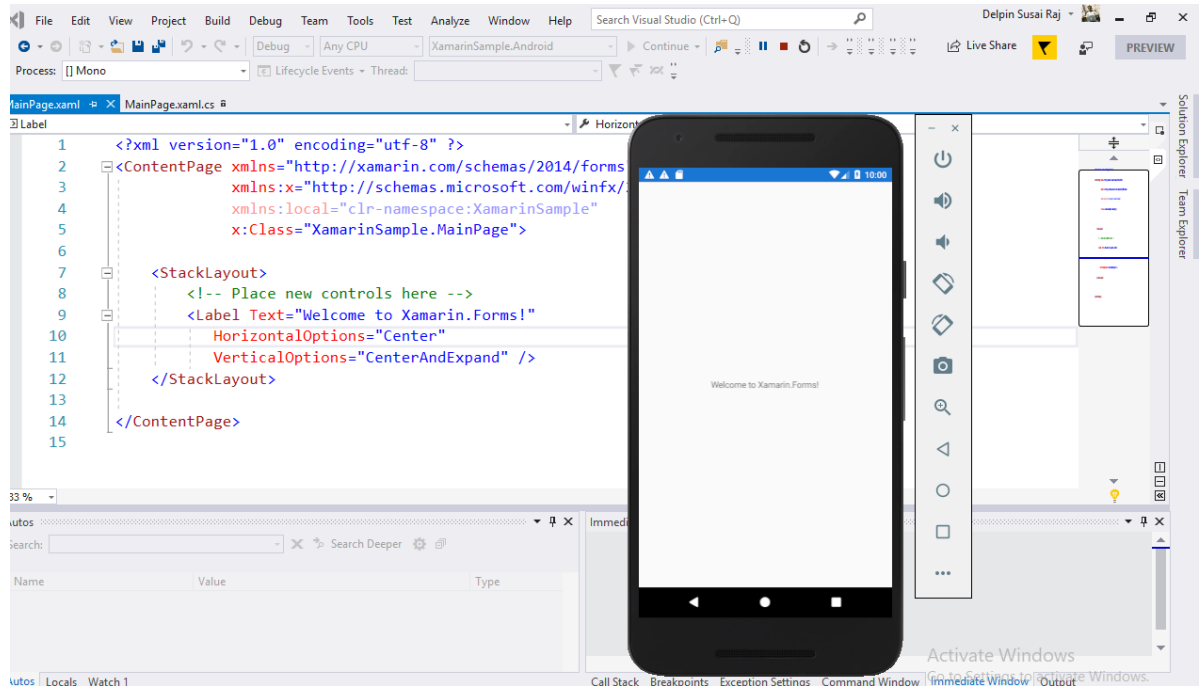


Рис. 2.3. – Використання Xamarin у поєднанні з Visual Studio

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1. Діаграма варіантів використання

Діаграма використання найпростіша - це представлення взаємодії користувача із системою, яка показує взаємозв'язок між користувачем та різними випадками використання, в яких користувач бере участь. Діаграма випадків використання може ідентифікувати різні типи користувачів системи та різні випадки використання, і часто вона супроводжується також іншими типами діаграм. Варіанти використання представлені колами або еліпсами.

Незважаючи на те, що сам випадок використання може детально вивчити кожну можливість, діаграма прикладів використання може допомогти забезпечити огляд системи на більш високому рівні. Раніше вже було сказано, що "схеми використання - це принципи вашої системи".

Через їх спрощений характер, схеми використання можуть бути хорошим інструментом комунікації для зацікавлених сторін. Креслення намагаються імітувати реальний світ і дають зацікавленій стороні уявлення про те, як буде розроблена система. Сіау та Лі провели дослідження, щоб визначити, чи взагалі існувала дійсна ситуація для схем використання або вони були непотрібними. Було виявлено, що діаграми випадків використання передають намір системи більш спрощеним чином зацікавленим сторонам і що вони "інтерпретуються більш повно, ніж діаграми класів".

Метою діаграми використання є відображення динамічного аспекту системи. Додаткові схеми та документація можуть бути використані для забезпечення повного функціонального та технічного уявлення про систему. Вони забезпечують спрощене та графічне представлення того, що система насправді повинна робити.

Елементи діаграми варіантів використання:

1. Рамки системи – прямокутник з еліпсами (прецедентами) всередині.
2. Актор - стилізований людський персонаж, означаючий набір ролей користувача (розуміється в широкому змісті: людина, зовнішня сутність, клас, інша система), взаємодіючого з деякою сутністю (системною, підсистемою, класом). Актори не можуть бути пов'язані між собою з іншим (за виключенням відносин щодо обробки / дослідження).
3. Прецедент - еліпс із надписом, що означає виконувану систематичну дію (може включати можливі варіанти). Надпис може бути ім'ям або описом (з точки зору актора) того, "що" робить система (а не "як"). Ім'я прецедента зв'язано з неперервним (атомарним) сценарієм - конкретною послідовністю дій. Під час сценарію актори обмінюються із систематичними повідомленнями. Сценарій може бути приведений на діаграмі прецедентів у відео UML-коментарі. З одним прецедентом може бути пов'язано кілька різних сценаріїв.

На рисунку 3.1 зображено діаграму варіантів використання, яка описує можливі дії користувача в системі.

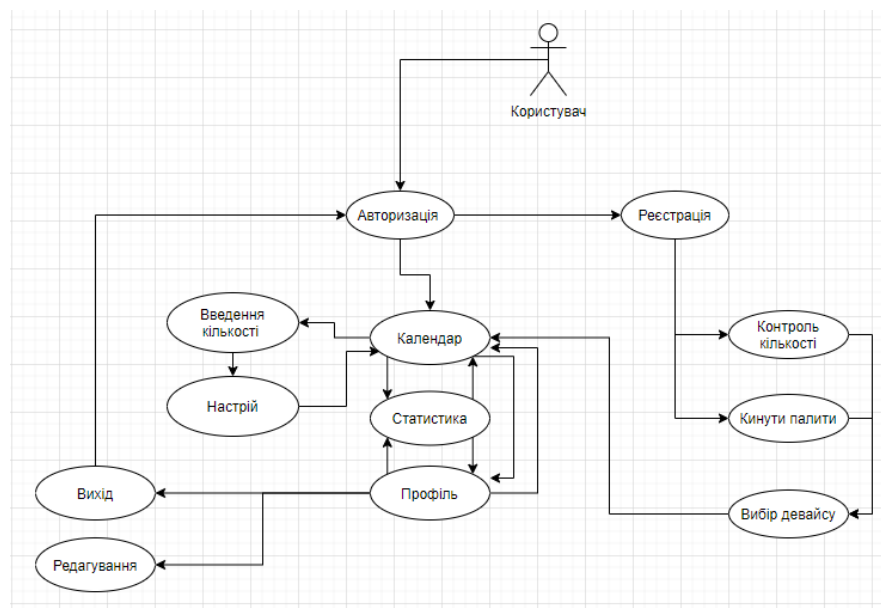


Рис. 3.1 — Діаграма варіантів використання

3.2. Діаграма класів програмного продукту

У програмній інженерії діаграма класів в Уніфікованій мові моделювання (UML) - це тип статичної структурної діаграми, що описує структуру системи, показуючи класи системи, їх атрибути, операції (або методи) та взаємозв'язки між об'єктами.

Діаграма класів є основним будівельним елементом об'єктно-орієнтованого моделювання. Він використовується для загального концептуального моделювання структури програми та для детального моделювання переведення моделей у програмовий код. Діаграми класів також можуть бути використані для моделювання даних. Класи на діаграмі класів представляють як основні елементи, взаємодії в програмі, так і класи, що програмуються.

На схемі класи представлені вікнами, які містять три відділення:

1. У верхньому відділенні міститься назва класу. Надруковано жирним шрифтом і відцентровано, а перша літера написана великими літерами;
2. Середній відсік містить атрибути класу. Вони вирівняні за лівим краєм, а перша буква мала;
3. У нижньому відділенні містяться операції, які може виконувати клас. Вони також вирівняні за лівим краєм, а перша буква – мала;

При проектуванні системи ряд класів ідентифікується та згруповується у схему класів, яка допомагає визначити статичні відносини між ними. При детальному моделюванні класи концептуального проекту часто поділяються на ряд підкласів.

Залежність - це семантичний зв'язок між залежними та незалежними елементами моделі. Він існує між двома елементами, якщо зміни у визначенні одного елемента (сервера або цілі) можуть спричинити зміни для іншого (клієнта або джерела). Ця асоціація є односпрямованою. Залежність

відображається у вигляді штрихової лінії з відкритою стрілкою, яка вказує від клієнта до постачальника.

Для подальшого опису поведінки систем ці діаграми класів можуть бути доповнені діаграмою стану або машиною стану UML.

Асоціація представляє родину посилянь. Двійкова асоціація (з двома кінцями) зазвичай представляється у вигляді рядка. Асоціація може пов'язувати будь-яку кількість класів. Асоціація з трьома ланками називається потрійною асоціацією. Асоціацію можна назвати, а кінці асоціації можна прикрасити іменами ролей, показниками власності, кратністю, видимістю та іншими властивостями.

Існує чотири різні типи асоціацій: двонаправлена, односпрямована, агрегаційна (включає агрегацію композиції) та рефлексивна. Двонаправлені та односпрямовані асоціації є найбільш поширеними.

Наприклад, клас польоту асоціюється з класом літака двонаправлено. Асоціація представляє статичне відношення, яке ділиться між об'єктами двох класів.

Агрегація є варіантом взаємозв'язку "має"; агрегація є більш конкретною, ніж асоціація. Це асоціація, яка представляє частково цілі або часткові стосунки. Як показано на зображенні, професор "має" клас для викладання. Як тип асоціації, агрегація може бути названа та мати ті самі прикраси, що і асоціація. Однак агрегація не може включати більше двох класів; це має бути бінарна асоціація. Крім того, навряд чи існує різниця між агрегаціями та асоціаціями під час реалізації, і діаграма може взагалі пропустити відносини агрегування. [7]

Агрегація може відбуватися, коли клас є колекцією або контейнером інших класів, але вміщені класи не мають сильної залежності життєвого циклу від контейнера. Вміст контейнера все ще існує, коли контейнер знищений.

В UML він графічно представлений у вигляді порожнистої форми ромба на вміщуючому класі одним рядком, що зв'язує його із вміщеним

класом. Сукупність - це семантично розширений об'єкт, який у багатьох операціях трактується як одиниця, хоча фізично він складається з декількох менших об'єктів.

Приклад: Бібліотека та студенти. Тут студент може існувати без бібліотеки, зв'язок між студентом і бібліотекою є агрегацією.

Це вказує на те, що один із двох пов'язаних класів (підклас) вважається спеціалізованою формою іншого (супер тип), а суперклас - узагальненням підкласу. На практиці це означає, що будь-який екземпляр підтипу є також екземпляром суперкласу. Зразкове дерево узагальнень цієї форми зустрічається в біологічній класифікації: людина - це підклас маймуни, який є підкласом ссавців тощо.

Графічне представлення UML узагальнення - це форма порожнистого трикутника на кінці суперкласу рядка (або дерева рядків), що зв'язує його з одним або кількома підтипами.

Відносини узагальнення також відомі як спадщина або відносини "є".

Суперклас (базовий клас) у відносинах узагальнення також відомий як "батьківський", суперклас, базовий клас або базовий тип.

Підтип у відносинах спеціалізації також відомий як "дочірній", підклас, похідний клас, похідний тип, клас успадкування або тип успадкування.

Зверніть увагу, що ці стосунки нічим не схожі на біологічні стосунки батьків та дітей: використання цих термінів надзвичайно поширене, але може ввести в оману.

Узагальнення може бути показано лише на діаграмах класів та на діаграмах використання.

При моделюванні UML взаємозв'язок реалізації - це взаємозв'язок між двома елементами моделі, в яких один елемент моделі (клієнт) реалізує (реалізує або виконує) поведінку, яку вказує інший елемент моделі (постачальник).

Графічне представлення UML реалізації - це порожниста форма трикутника на кінці інтерфейсу штрихової лінії (або дерева рядків), яка

з'єднує її з одним або кількома реалізаторами. Проста головка стрілки використовується на кінці інтерфейсу штрихової лінії, що з'єднує її з користувачами. У діаграмах компонентів використовується графічна умова «м'яч і сокет» (реалізатори виставляють кульку або льодяник, тоді як користувачі показують сокет). Реалізації можна показати лише на діаграмах класів або компонентів. Реалізація - це взаємозв'язок між класами, інтерфейсами, компонентами та пакетами, що з'єднує елемент клієнта з елементом постачальника. Зв'язок реалізації між класами / компонентами та інтерфейсами показує, що клас / компонент реалізує операції, пропонувані інтерфейсом.

Залежність - це слабша форма зв'язку, яка вказує на те, що один клас залежить від іншого, оскільки він використовує його в певний момент часу. Один клас залежить від іншого, якщо незалежний клас є змінною параметра або локальною змінною методу залежного класу. Це відрізняється від асоціації, де атрибут залежного класу є екземпляром незалежного класу. Іноді відносини між двома класами дуже слабкі. Вони взагалі не реалізовані зі змінними-членами. Швидше вони можуть бути реалізовані як аргументи функції-члена.

Представлення UML асоціації - це лінія, що з'єднує два пов'язані класи. На кожному кінці рядка є додаткові позначення. Наприклад, ми можемо вказати, використовуючи наконечник стрілки, що загострений кінець видно з хвоста стрілки. Ми можемо вказати власність шляхом розміщення кульки, ролі, яку відіграють елементи цього кінця, вказавши ім'я ролі та множинність екземплярів цієї сутності (діапазон кількості об'єктів, які беруть участь в асоціації з точки зору іншого кінця).

Класи сутності моделюють інформацію, якою обробляє система, а іноді і поведінку, пов'язану з цією інформацією. Їх не слід ідентифікувати як таблиці баз даних чи інших сховищ даних.

Вони намальовані як кола з короткою лінією, прикріпленою до нижньої частини кола. Як варіант, їх можна намалювати як звичайні класи із позначенням стереотипу «сутність» над назвою класу.

Оскільки рішення розбито на 6 проектів, то і діаграми класів створено для кожного з дев'яти проектів. Діаграми класів зображено на рисунках 3.2 – 3.7.

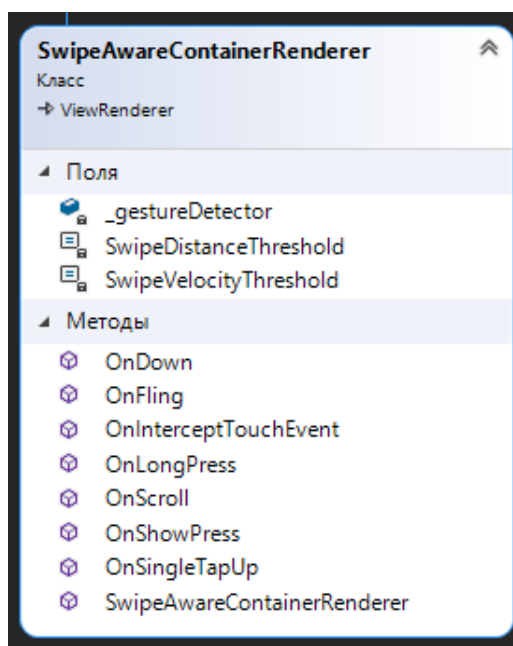


Рис. 3.2 — Діаграма класів проекту Calendar.Android

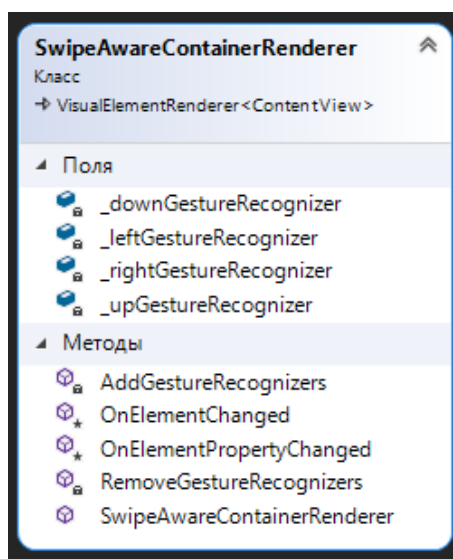


Рис. 3.3 — Діаграма класів проекту Calendar.iOS

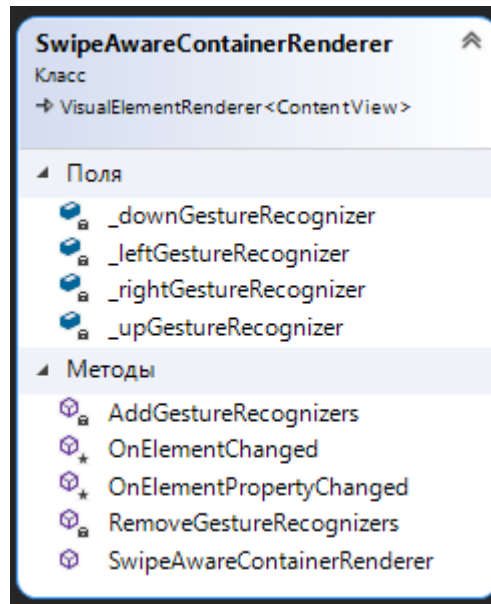


Рис. 3.4 — Діаграма класів проекту Calendar.Mobile

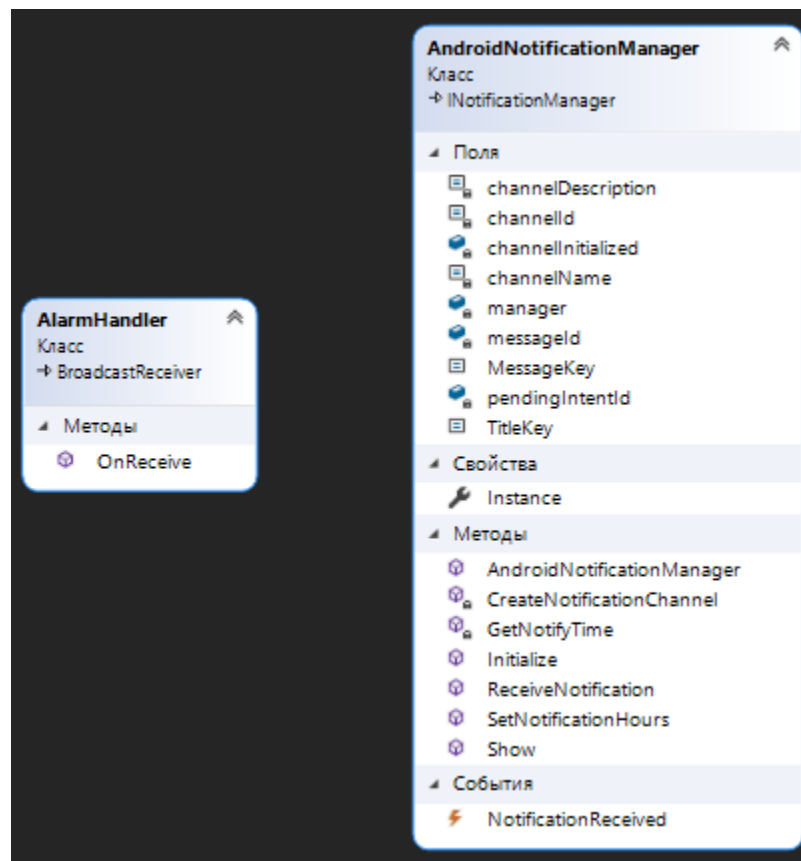


Рис. 3.5 — Діаграма класів проекту Fumes.Android

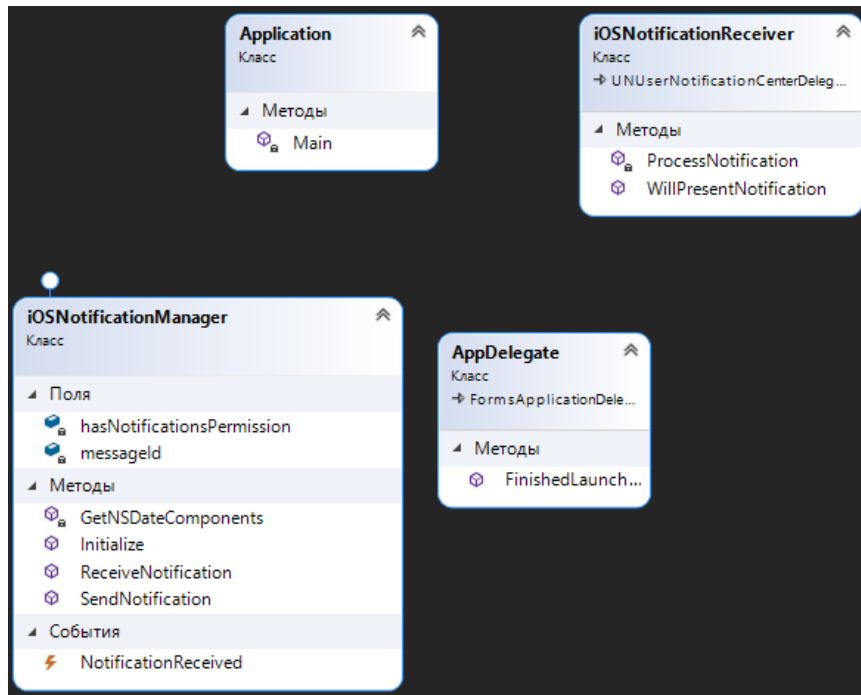


Рис. 3.6 — Діаграма класів проекту Fumes.iOS

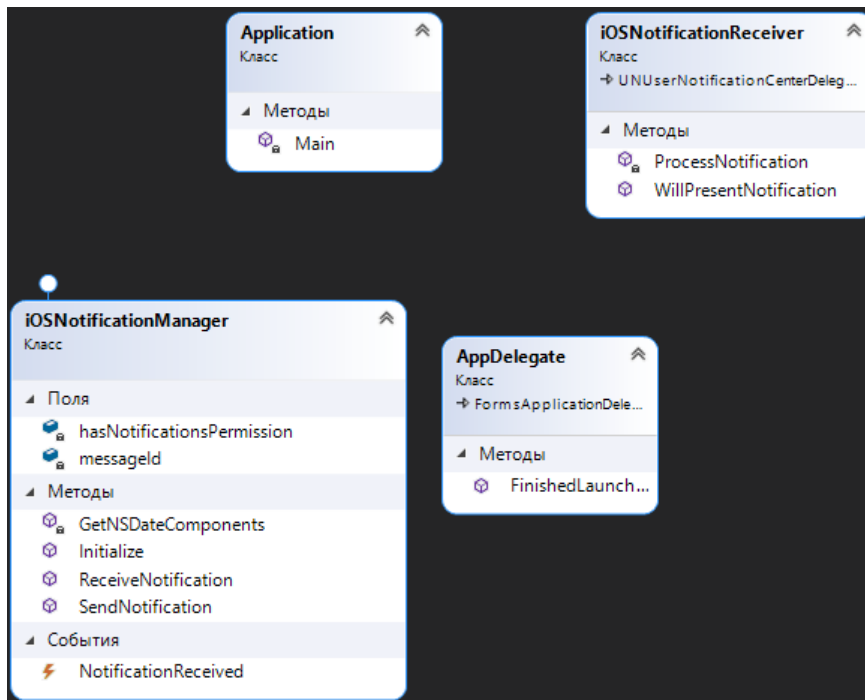


Рис. 3.7 — Діаграма класів проекту Fumes.iOS

3.3. Графічний інтерфейс користувача

Графічний користувальницький інтерфейс - це форма користувальницького інтерфейсу, що дозволяє користувачам взаємодіяти з електронними пристроями за допомогою графічних піктограм та звукового індикатора, таких як основне позначення, замість текстових інтерфейсів користувача, набрані мітки команд або текстова навігація. Графічні інтерфейси були введені у відповідь на сприйману круту криву навчання інтерфейсів командного рядка (CLI), які вимагають набору команд на клавіатурі комп'ютера.

Дії в графічному інтерфейсі зазвичай виконуються за допомогою прямого маніпулювання графічними елементами. Окрім комп'ютерів, графічні інтерфейси використовуються в багатьох портативних мобільних пристроях, таких як MP3-плеєри, портативні медіаплеєри, ігрові пристрої, смартфони та менші побутові, офісні та промислові елементи управління. Термін "графічний інтерфейс", як правило, не застосовується до інших типів інтерфейсів із нижчою роздільною здатністю, таких як відеоігри (де кращий дисплей (HUD) є кращим), або не включаючи плоскі екрани, такі як об'ємні дисплеї оскільки цей термін обмежений сферою двовимірних дисплеїв, здатних описувати загальну інформацію, за традицією досліджень інформатики в Дослідницькому центрі Xerox Palo Alto.

Розробка візуальної композиції та часової поведінки графічного інтерфейсу є важливою частиною програмування програмного забезпечення в області взаємодії людина-комп'ютер. Її метою є підвищення ефективності та простоти використання основного логічного дизайну збереженої програми, дисципліни проектування, що називається зручністю використання. Методи дизайну, орієнтованого на користувача, використовуються для того, щоб візуальна мова, введена в дизайн, була добре адаптована до завдань.

Помітні особливості графічного інтерфейсу програми іноді називають хромом або графічним інтерфейсом (вимовляється глейко). Зазвичай користувачі взаємодіють з інформацією, маніпулюючи візуальними віджетами, що дозволяють здійснювати взаємодії, відповідні виду даних, який вони мають. Віджети добре продуманого інтерфейсу обрані для підтримки дій, необхідних для досягнення цілей користувачів. Модель-вигляд-контролер дозволяє гнучкі структури, в яких інтерфейс не залежить і опосередковано пов'язаний з функціями програми, тому графічний інтерфейс можна легко налаштувати. Це дозволяє користувачам за бажанням вибирати або розробляти іншу обкладинку, а також полегшує роботу дизайнера по зміні інтерфейсу в міру розвитку потреб користувача. Хороший дизайн інтерфейсу стосується користувачів більше, а архітектури системи менше. Великі віджети, такі як вікна, зазвичай містять рамку або контейнер для основного вмісту презентації, наприклад веб-сторінки, повідомлення електронної пошти або малюнка. Менші, як правило, виступають інструментом введення користувачем.

Графічний інтерфейс може бути розроблений для потреб вертикального ринку як графічний користувальницький інтерфейс, специфічний для додатків. Приклади включають автоматичні касові автомати (банкомати), сенсорні екрани торгових точок (POS) у ресторанах, каси самообслуговування, що використовуються в роздрібному магазині, самоквитки та реєстрація авіаквитків, інформаційні кіоски у громадському просторі, наприклад залізничний вокзал чи музей, а також монітори або екрани управління у вбудованому промисловому додатку, що використовує операційну систему реального часу (RTOS).

До 1980-х років стільникові телефони та портативні ігрові системи також використовували графічні інтерфейси із сенсорним екраном для додатків. Нові автомобілі використовують графічні інтерфейси в своїх навігаційних системах та мультимедійних центрах або комбінаціях навігаційних мультимедійних центрів.

Графічний інтерфейс використовує поєднання технологій та пристроїв, щоб забезпечити платформу, з якою користувачі можуть взаємодіяти для завдань збору та виробництва інформації.

Ряд елементів, що відповідають візуальній мові, еволюціонував, щоб представляти інформацію, що зберігається в комп'ютерах. Це полегшує людям, які мають незначні навички роботи з комп'ютером, працювати та користуватися комп'ютерним програмним забезпеченням. Найпоширенішим поєднанням таких елементів у графічних інтерфейсах є парадигма вікон, піктограм, меню, покажчика (WIMP), особливо в персональних комп'ютерах.

Стиль взаємодії WIMP використовує віртуальний пристрій введення для відображення позиції інтерфейсу вказівного пристрою, найчастіше миші, і представляє інформацію, організовану у вікнах і представлену піктограмами. Доступні команди компілюються в меню, а дії виконуються, роблячи жести за допомогою вказівного пристрою. Менеджер вікон полегшує взаємодію між вікнами, програмами та віконною системою. Система вікон обробляє апаратні пристрої, такі як вказівні пристрої, графічне обладнання та позиціонування вказівника.

В персональних комп'ютерах усі ці елементи моделюються за допомогою метафори робочого столу для створення імітації, яка називається середовищем робочого столу, в якому дисплей представляє робочий стіл, на якому можна розміщувати документи та папки документів. Менеджери вікон та інше програмне забезпечення поєднують для імітації середовища робочого столу з різним ступенем реалізму.

Графічний інтерфейс розробленого додатку складається з 7 основних екранів, а саме:

- екран входу (рис. 3.8);
- екран реєстрації (рис. 3.9);
- екран вибору варіанту використання (рис. 3.10);
- екран вибору пристрою (рис. 3.11);

- екран статистики (рис. 3.12);
- екран календаря (рис. 3.13);
- екран профіля (рис. 3.14);

3.4 Тестування ПЗ

Обов'язковим атрибутом якісного програмного забезпечення є тестування.

Тестування поділяється на два види автоматизоване та мануальне.

Автоматизоване тестування включає в себе набір програмних засобів для створення та перевірки результатів створених тестів, що сприяє пришвидшенню процесу тестування, а також його спрощення.

Мануальне тестування є процесом ручної перевірки програмного забезпечення, в якому тестувальник відіграє роль користувача програми використовуючи всі наявні властивості та функції.

Таблиця 4.1– Тест-кейси для тестування функціоналу авторизації

Функціонал авторизації користувача		
Передумова: Відкрити додаток		
Крок тесту	Дані тесту	Очікуваний результат
Авторизація з коректними даними	<ul style="list-style-type: none"> - Введення логіну - Введення паролю - Натиснення кнопки «Увійти» 	<ul style="list-style-type: none"> - Користувача авторизовано - Відкривається головна сторінка додатку
Авторизація з	<ul style="list-style-type: none"> - Введення логіну - Введення паролю 	<ul style="list-style-type: none"> - Авторизація не відбулася

Функціонал авторизації користувача		
Передумова: Відкрити додаток		
Крок тесту	Дані тесту	Очікуваний результат
некоректними даними	- Натиснення кнопки «Увійти»	- Повідомлення про введення некоректних даних
Відновлення паролю	Натиснення кнопки «Забули пароль»	- Відкриття форми для скидання паролю.

Таблиця 4.2– Тест-кейси для тестування функціоналу збереження даних та перегляду статистики

Функціонал збереження даних та перегляд статистики		
Передумова: Увійти в обліковий запис		
Крок тесту	Дані тесту	Очікуваний результат
Введення та збереження особистих даних	- Натиснення кнопки актуальної дати на календарі - Введення особистих даних - Натиснення кнопки «Зберегти»	- З'являється вспливаюче вікно, для введення даних - Відбувається коректне внесення даних - Відбувається збереження даних в статистиці.
Перегляд сторінки статистики	- Відкрити сторінку статистики	- Відбувається автооновлення статистики

Таблиця 4.3 – Тест-кейси для тестування функціоналу збереження даних та перегляду статистики

Функціонал перегляду та редагуванню профіля		
Передумова: Увійти в обліковий запис		
Крок тесту	Дані тесту	Очікуваний результат
Перейти на сторінку	- Натиснення кнопки	- З'являється сторінка

Функціонал перегляду та редагуванню профіля		
Передумова: Увійти в обліковий запис		
Крок тесту	Дані тесту	Очікуваний результат
профіля	<ul style="list-style-type: none"> - Натиснення кнопки «Профіль» - Натиснення кнопки «Редагувати профіль» - Натиснення кнопки «Зберегти» 	<ul style="list-style-type: none"> - профілю - Відбувається перехід на сторінку редагування даних - Відбувається збереження даних профіля.
Налаштування нагадування	<ul style="list-style-type: none"> - Натиснення кнопки «Профіль» - Натиснення кнопки «Налаштувати нагадування» - Вибрати зручний час - Натиснення кнопки «Зберегти» 	<ul style="list-style-type: none"> - З'являється сторінка профілю - З'являється впливаюче вікно - Вибрані об'єкти відображаються коректно - Відбувається збереження даних нагадування

Таблиця 4.4 – Тест-кейси для тестування функціоналу збереження даних та перегляду статистики

Функціонал виходу з додатку		
Передумова: Увійти в обліковий запис		
Крок тесту	Дані тесту	Очікуваний результат
Перейти на сторінку профіля	<ul style="list-style-type: none"> - Натиснення кнопки «Профіль» - Натиснення кнопки «Вихід» 	<ul style="list-style-type: none"> - З'являється сторінка профілю - Відбувається перехід на сторінку Авторихації

РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА

Для входу в додаток слід виконати наступні дії:

- відкрити додаток;
- дочекатися повного завантаження інтерфейсу;
- ввести логін;
- ввести пароль;
- натиснути кнопку «Вхід»;

Для реєстрації в додатку порядок дій такий:

- відкрити додаток;
- дочекатися повного завантаження інтерфейсу
- натиснути кнопку «Зареєструватися»
- ввести логін та пароль
- обрати варіант використання
- обрати пристрій

Для оновлення статистики в додатку порядок дій такий:

- перейти в додаток за допомогою нотифікації;
- вказати дані та натиснути зберегти;
або
- відкрити додаток;
- дочекатися повного завантаження інтерфейсу;
- натиснути на поточну дату;
- вказати дані та натиснути зберегти;

Для детального перегляду статистики в додатку порядок дій такий:

- відкрити додаток;
- дочекатися повного завантаження інтерфейсу;

- натиснути на кнопку «Статистика»;
- переглянути чітку статистику;

Для редагування профілю в додатку порядок дій такий:

- відкрити додаток;
- дочекатися повного завантаження інтерфейсу;
- натиснути на кнопку «Профіль»;
- натиснути на кнопку «Редагувати профіль»;
- після внесення даних натиснути на кнопку «Зберенти»;

Для налаштування нагадування в додатку порядок дій такий:

- відкрити додаток;
- дочекатися повного завантаження інтерфейсу;
- натиснути на кнопку «Профіль»;
- натиснути на кнопку «Налаштування нагадування»;
- після внесення даних натиснути на кнопку «Зберенти»;

ВИСНОВКИ

Ідея дипломної роботи полягає в розробці програмного забезпечення для споживачів тютюнових виробів та сумішів для куріння, що бажають звільнитися від шкідливої звички.

1. За допомогою дослідження, було виявлено недоліки аналогових програм та розроблено додаток, який буде перевершувати наявне програмне забезпечення зі схожою структурою. А саме спрощення та пришвидшення моніторингу стану курця.

2. Описано програмні засоби, які застосовувалися для розробки програмного забезпечення. Після аналізу специфікацій до програмного забезпечення було визначено, що найкращим рішенням для розробки додатку є мова програмування C# та Xamarin.Forms, які у поєднанні з середовищем розробки Visual Studio дають можливість спроектувати та створити додаток, який відповідає всім вимогам технічного завдання, а також потребам споживачів.

3. Тестування підтвердило, що для використання даного програмного забезпечення достатньо мати телефон з досить слабкими характеристиками.

4. Додаток буде корисним для споживачів тютюнових виробів та сумішів для куріння, які бажають контролювати кількість споживання або зменшити кількість споживання вище сказаних продуктів.

5. Результати досліджень бакалаврської роботи апробовані всеукраїнських науково-технічних конференціях:

1. Застосування програмного забезпечення в інфокомунікаційних технологіях // Розробка програмного забезпечення для інфокомунікацій. Збірник тез. 12.02.2020, ДУТ, м. Київ — К.: ДУТ, 2021. — С. 28 – 29.

2. II Всеукраїнська науково-технічна конференція «Сучасний стан та перспективи розвитку ІОТ». Збірник тез. 09.04.2021, ДУТ, м. Київ — К.: ДУТ, 2021.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Tobacco Fact sheet N°339". May 2014. Retrieved 13 May 2015.
2. Reitsma, Marissa B; Fullman, Nancy; Ng, Marie; Salama, Joseph S; Abajobir, Amanuel (April 2017). "Smoking prevalence and attributable disease burden in 195 countries and territories, 1990–2015: a systematic analysis from the Global Burden of Disease Study 2015". *The Lancet*. 389 (10082): 1885–906. doi:10.1016/S0140-6736(17)30819-X. PMC 5439023. PMID 28390697.
3. See Gately; Wilbert
4. Robicsek (1978), p. 30
5. P. Ram Manohar, "Smoking and Ayurvedic Medicine in India" in *Smoke*, pp. 68–75
6. Gilman & Xun 2004, pp. 20-21.
7. Phillips, pp. 303–19
8. Coe, pp. 74–81
9. Jamestown, Virginia: An Overview Archived 7 February 2009 at the Wayback Machine
10. Kulikoff, pp. 38–39.
11. Cooper, William J., *Liberty and Slavery: Southern Politics to 1860*, Univ of South Carolina Press, 2001, p. 9.
12. *The People's Chronology*, 1994 by James Trager
13. Lloyd & Mitchinson
14. Tanya Pollard, "The Pleasures and Perils of Smoking in Early Modern England" in *Smoke*, p. 38
15. Timon Screech, "Tobacco in Edo Period Japan" in *Smoke*, pp. 92–99
16. Gilman & Xun 2004, pp. 15-16.
17. Roberts 2004, pp. 53–54.
18. Burns, pp. 134–35.
19. Jos Ten Berge, "The Belle Epoque of Opium in *Smoke*, p. 114

20. Stephen R. Platt, *Imperial Twilight: the Opium War and the End of China's Last Golden Age* (NY: Knopf, 2018), 166-73. ISBN 9780307961730
21. Benjamin Rush, M.D. (1798). *Essays, Literary, Moral and Philosophical*, 2nd ed.
22. James C. Coleman, Ph.D. (1976). *Abnormal Psychology and Modern Life*, 5th ed. Scott, Foresman & Co. p. 43 and 427. ASIN B002KI5YEW. OCLC 1602234.
23. Proctor 2000, p. 178
24. Proctor 2000, p. 219
25. Proctor 2000, p. 187
26. Proctor 2000, p. 245
27. Proctor, Robert N. (1996). *Nazi Medicine and Public Health Policy. Dimensions*, Anti-Defamation League. Archived from the original on 31 May 2008. Retrieved 1 June 2008.
28. Proctor 2000, p. 228
29. White, Colin (September 1989). "Research on Smoking and Lung Cancer: A Landmark in the History of Chronic Disease Epidemiology". *The Yale Journal of Biology and Medicine*. 63 (1): 29–46. PMC 2589239. PMID 2192501.
30. Doll R, Hill AB; Hill (30 September 1950). "Smoking and carcinoma of the lung. Preliminary report". *British Medical Journal*. 2 (4682): 739–48. doi:10.1136/bmj.2.4682.739. PMC 2038856. PMID 14772469.
31. Doll R, Hill AB; Hill (26 June 1954). "The mortality of doctors in relation to their smoking habits. A preliminary report". *British Medical Journal*. 1 (4877): 1451–55. doi:10.1136/bmj.1.4877.1451. PMC 2085438. PMID 13160495.
32. Berridge, V. *Marketing Health: Smoking and the Discourse of Public Health in Britain, 1945–2000*, Oxford: Oxford University Press, 2007.
33. "Smoking and Health: Report of the Advisory Committee to the Surgeon General of the Public Health Service". 1964.

34. "Reports of the Surgeon General, U.S. Public Health Service". Archived from the original on 22 April 2015. Retrieved 17 April 2015.
35. Geyelin, Milo (23 November 1998). "Forty-Six States Agree to Accept \$206 Billion Tobacco Settlement". *Wall Street Journal*.
36. VJ Rock, MPH, A Malarcher, JW Kahende, K Asman, MSPH, C Husten, MD, R Caraballo (9 November 2007). "Cigarette Smoking Among Adults – United States, 2006". United States Centers for Disease Control and Prevention. Retrieved 1 January 2009. [...]In 2006, an estimated 20.8% (45.3 million) of U.S. adults[...]
37. Hilton, Matthew (4 May 2000). *Smoking in British Popular Culture, 1800–2000: Perfect Pleasures*. Manchester University Press. pp. 229–41. ISBN 978-0-7190-5257-6. Retrieved 22 March 2009.
38. "WHO/WPRO-Smoking Statistics". World Health Organization Regional Office for the Western Pacific. 28 May 2002. Archived from the original on 8 November 2009. Retrieved 1 January 2009.
39. Roberts 2004, pp. 46–57.
40. WHO Report on the Global Tobacco Epidemic: The MPOWER Package. World Health Organization (Report). Geneva. 2008. pp. 267–88.
41. History of the WHO Framework Convention on Tobacco Control. World Health Organization. Geneva: World Health Organization. 2009. ISBN 978-92-4-156392-5. OCLC 547193748.
42. WHO Framework Convention on Tobacco Control. World Health Organization. Geneva, Switzerland: World Health Organization. 2003. ISBN 978-92-4-159101-0. OCLC 54966940.
43. Chung-Hall, Janet; Craig, Lorraine; Gravely, Shannon; Sansone, Natalie; Fong, Geoffrey T. (17 August 2018). "Impact of the WHO FCTC over the first decade: a global evidence review prepared for the Impact Assessment Expert Group". *Tobacco Control*. 28 (Suppl 2): tobaccocontrol–2018–054389. doi:10.1136/tobaccocontrol-2018-054389. ISSN 0964-4563. PMC 6589489. PMID 29880598.

44. DoJ-DEA-History-1985-1990 Archived 25 February 2009 at the Wayback Machine
45. "Cracked up". salon.com. Archived from the original on 2 December 2008.
46. Ritchie, Hannah; Roser, Max (23 May 2013). "Smoking". Our World in Data.
47. Leslie Iverson, "Why do We Smoke?: The Physiology of Smoking" in *Smoke*, p. 320
48. Centers for Disease Control and Prevention (CDC) (2002). "Annual smoking-attributable mortality, years of potential life lost, and economic costs – United States, 1995–1999". *MMWR Morb. Mortal. Wkly. Rep.* 51 (14): 300–03. PMID 12002168.
49. Doll R, Peto R, Boreham J, Sutherland I; Peto; Boreham; Sutherland (2004). "Mortality in relation to smoking: 50 years' observations on male British doctors". *BMJ.* 328 (7455): 1519. doi:10.1136/bmj.38142.554479.AE. PMC 437139. PMID 15213107.
50. Thun MJ, Day-Lally CA, Calle EE, Flanders WD, Heath CW; Day-Lally; Calle; Flanders; Heath Jr (1995). "Excess mortality among cigarette smokers: changes in a 20-year interval". *Am J Public Health.* 85 (9): 1223–30. doi:10.2105/ajph.85.9.1223. PMC 1615570. PMID 7661229.
51. Thun MJ, Hannan LM, Adams-Campbell LL, Boffetta P, Buring JE, Feskanich D, Flanders WD, Jee SH, Katanoda K, Kolonel LN, Lee IM, Marugame T, Palmer JR, Riboli E, Sobue T, Avila-Tang E, Wilkens LR, Samet JM; Hannan; Adams-Campbell; Boffetta; Buring; Feskanich; Flanders; Jee; Katanoda; Kolonel; Lee; Marugame; Palmer; Riboli; Sobue; Avila-Tang; Wilkens; Samet (2008). "Lung cancer occurrence in never-smokers: An analysis of 13 cohorts and 22 cancer registry studies". *PLOS Med.* 5 (9): e185. doi:10.1371/journal.pmed.0050185. PMC 2531137. PMID 18788891.

52. Kenneth Johnson (24 January 2018). "Just one cigarette a day seriously elevates cardiovascular risk". *British Medical Journal*. 360: k167. doi:10.1136/bmj.k167. PMID 29367307. S2CID 46825572.
53. "Just one cigarette a day can cause serious heart problems". *New Scientist*. 3 February 2020.
54. Law MR, Morris JK, Wald NJ; Morris; Wald (1997). "Environmental tobacco smoke exposure and ischaemic heart disease: an evaluation of the evidence". *BMJ*. 315 (7114): 973–80. doi:10.1136/bmj.315.7114.973. PMC 2127675. PMID 9365294.
55. American Legacy Foundation factsheet on lung cancer Archived 2007-09-27 at the Wayback Machine; their cited source is: CDC (Centers for Disease Control) The Health Consequences of Smoking: A Report of the Surgeon General. 2004.
56. Nyboe J, Jensen G, Appleyard M, Schnohr P; Jensen; Appleyard; Schnohr (1989). "Risk factors for acute myocardial infarction in Copenhagen. I: Hereditary, educational and socioeconomic factors. Copenhagen City Heart Study". *Eur Heart J*. 10 (10): 910–16. doi:10.1093/oxfordjournals.eurheartj.a059401. PMID 2598948.
57. Devereux G (2006). "ABC of chronic obstructive pulmonary disease. Definition, epidemiology, and risk factors". *BMJ*. 332 (7550): 1142–44. doi:10.1136/bmj.332.7550.1142. PMC 1459603. PMID 16690673.
58. Braun JM, Kahn RS, Froehlich T, Auinger P, Lanphear BP; Kahn; Froehlich; Auinger; Lanphear (2006). "Exposures to environmental toxicants and attention deficit hyperactivity disorder in U.S. children". *Environ. Health Perspect*. 114 (12): 1904–09. doi:10.1289/ehp.10274. PMC 1764142. PMID 17185283.
59. Tomar, S. L.; Asma, S. (May 2000). "Smoking-attributable periodontitis in the United States: findings from NHANES III. National Health and Nutrition Examination Survey". *Journal of Periodontology*. 71 (5): 743–51. doi:10.1902/jop.2000.71.5.743. ISSN 0022-3492. PMID 10872955.

60. Ramon, Jose-Maria; Echeverria, Jose-Javier (August 2002). "Effects of smoking on periodontal tissues". *Journal of Clinical Periodontology*. 29 (8): 771–76. doi:10.1034/j.1600-051x.2002.290815.x. ISSN 0303-6979. PMID 12390575.
61. Harris, C.; Warnakulasuriya, K.A.A.S.; Gelbier, S.; Johnson, N.W.; Peters, T.J. (December 1997). "Oral and Dental Health in Alcohol Misusing Patients". *Alcoholism: Clinical and Experimental Research*. 21 (9): 1707–09. doi:10.1111/j.1530-0277.1997.tb04511.x. ISSN 0145-6008. PMID 9438534.
62. Axeix, Tony; Hedin, C. Anders (December 1982). "Epidemiologic study of excessive oral melanin pigmentation with special reference to the influence of tobacco habits". *European Journal of Oral Sciences*. 90 (6): 434–42. doi:10.1111/j.1600-0722.1982.tb00760.x. ISSN 0909-8836. PMID 6961509.
63. Brocklehurst, Paul; Kujan, Omar; O'Malley, Lucy A; Ogden, Graham; Shepherd, Simon; Glenny, Anne-Marie (19 November 2013). "Screening programmes for the early detection and prevention of oral cancer". *Cochrane Database of Systematic Reviews* (11): CD004150. doi:10.1002/14651858.CD004150.pub4. ISSN 1465-1858. PMID 24254989.
64. Monteiro-da-Silva, Filipa; Sampaio-Maia, Benedita; Pereira, Maria de Lurdes; Araujo, Ricardo (4 March 2013). "Characterization of the oral fungal microbiota in smokers and non-smokers". *European Journal of Oral Sciences*. 121 (2): 132–35. doi:10.1111/eos.12030. hdl:10216/114867. ISSN 0909-8836. PMID 23489903.
65. Reibel, Jesper (2003). "Tobacco and oral diseases. Update on the evidence, with recommendations". *Medical Principles and Practice*. 12 Suppl 1: 22–32. doi:10.1159/000069845. ISSN 1011-7571. PMID 12707498.
66. Berlinger (5 May 2016). "California raises smoking age to 21". CNN. Retrieved 5 May 2016.
67. "Which country smokes the most?". Gadling. 12 May 2008.

- 68."Cigarette Smoking Among Adults – United States, 2006". Cdc.gov. Retrieved 18 September 2008.
- 69.Cataldo JK, Prochaska JJ, Glantz SA; Prochaska; Glantz (2010). "Cigarette Smoking is a Risk Factor for Alzheimer's Disease: an Analysis Controlling for Tobacco Industry Affiliation". *Journal of Alzheimer's Disease*. 19 (2): 465–80. doi:10.3233/JAD-2010-1240. PMC 2906761. PMID 20110594.
- 70.Cosnes J, Carbonnel F, Carrat F, Beaugerie L, Cattan S, Gendre J; Carbonnel; Carrat; Beaugerie; Cattan; Gendre (1999). "Effects of current and former cigarette smoking on the clinical course of Crohn's disease". *Aliment. Pharmacol. Ther.* 13 (11): 1403–11. doi:10.1046/j.1365-2036.1999.00630.x. PMID 10571595. S2CID 6620451.
- 71.Calkins BM (1989). "A meta-analysis of the role of smoking in inflammatory bowel disease". *Dig. Dis. Sci.* 34 (12): 1841–54. doi:10.1007/BF01536701. PMID 2598752. S2CID 5775169.
- 72.Lakatos PL, Szamosi T, Lakatos L; Szamosi; Lakatos (2007). "Smoking in inflammatory bowel diseases: good, bad or ugly?". *World J. Gastroenterol.* 13 (46): 6134–39. doi:10.3748/wjg.13.6134. PMC 4171221. PMID 18069751.
- 73."Smoking and Diabetes". Centers for Disease Control and Prevention. 23 April 2018. Archived from the original on 24 August 2019. Retrieved 4 November 2019.
- 74."Share of deaths from smoking". Our World in Data. Retrieved 5 March 2020.
- 75."Death rate from smoking". Our World in Data. Retrieved 5 March 2020.
- 76.Leslie Iverson, "Why do We Smoke?: The Physiology of Smoking" in *Smoke*, p. 318
- 77.Leslie Iverson, "Why do We Smoke?: The Physiology of Smoking" in *Smoke*, pp. 320–21

78. Model D (1985). "Smoker's face: an underrated clinical sign?". *Br Med J (Clin Res Ed)*. 291 (6511): 1760–62. doi:10.1136/bmj.291.6511.1760. PMC 1419177. PMID 3936573.
79. Talhout R, Schulz T, Florek E, van Benthem J, Wester P, Opperhuizen A (2011). "Hazardous compounds in tobacco smoke". *Int J Environ Res Public Health*. 8 (2): 613–28. doi:10.3390/ijerph8020613. PMC 3084482. PMID 21556207.
80. Kastan MB (2008). "DNA damage responses: mechanisms and roles in human disease: 2007 G.H.A. Clowes Memorial Award Lecture". *Mol. Cancer Res*. 6 (4): 517–24. doi:10.1158/1541-7786.MCR-08-0020. PMID 18403632.
81. Cunningham FH, Fiebelkorn S, Johnson M, Meredith C (2011). "A novel application of the Margin of Exposure approach: segregation of tobacco smoke toxicants". *Food Chem. Toxicol*. 49 (11): 2921–33. doi:10.1016/j.fct.2011.07.019. PMID 21802474.
82. Liu XY, Zhu MX, Xie JP (2010). "Mutagenicity of acrolein and acrolein-induced DNA adducts". *Toxicol. Mech. Methods*. 20 (1): 36–44. doi:10.3109/15376510903530845. PMID 20158384. S2CID 8812192.
83. Speit G, Merk O (2002). "Evaluation of mutagenic effects of formaldehyde in vitro: detection of crosslinks and mutations in mouse lymphoma cells". *Mutagenesis*. 17 (3): 183–87. doi:10.1093/mutage/17.3.183. PMID 11971987.
84. Pu X, Kamendulis LM, Klaunig JE (2009). "Acrylonitrile-induced oxidative stress and oxidative DNA damage in male Sprague-Dawley rats". *Toxicol. Sci*. 111 (1): 64–71. doi:10.1093/toxsci/kfp133. PMC 2726299. PMID 19546159.
85. Koturbash I, Scherhag A, Sorrentino J, Sexton K, Bodnar W, Swenberg JA, Beland FA, Pardo-Manuel Devillena F, Rusyn I, Pogribny IP (2011). "Epigenetic mechanisms of mouse interstrain variability in genotoxicity of

- the environmental toxicant 1,3-butadiene". *Toxicol. Sci.* 122 (2): 448–56. doi:10.1093/toxsci/kfr133. PMC 3155089. PMID 21602187.
86. Garcia CC, Angeli JP, Freitas FP, Gomes OF, de Oliveira TF, Loureiro AP, Di Mascio P, Medeiros MH (2011). "[13C2]-Acetaldehyde promotes unequivocal formation of 1,N2-propano-2'-deoxyguanosine in human cells". *J. Am. Chem. Soc.* 133 (24): 9140–43. doi:10.1021/ja2004686. PMID 21604744.
87. Tompkins EM, McLuckie KI, Jones DJ, Farmer PB, Brown K (2009). "Mutagenicity of DNA adducts derived from ethylene oxide exposure in the pSP189 shuttle vector replicated in human Ad293 cells". *Mutat. Res.* 678 (2): 129–37. doi:10.1016/j.mrgentox.2009.05.011. PMID 19477295.
88. Fabiani R, Rosignoli P, De Bartolomeo A, Fucelli R, Morozzi G (2007). "DNA-damaging ability of isoprene and isoprene mono-epoxide (EPOX I) in human cells evaluated with the comet assay". *Mutat. Res.* 629 (1): 7–13. doi:10.1016/j.mrgentox.2006.12.007. PMID 17317274.
89. Gay, Peter (1988). *Freud: A Life for Our Time*. New York: W.W. Norton & Company. pp. 650–51. ISBN 978-0-393-32861-5.
90. Harris, J.R. (1998). *The Nurture Assumption: Why children turn out the way they do*. New York: Free Press.
91. Parrott AC (1999). "Does cigarette smoking cause stress?". *American Psychologist.* 54 (10): 817–20. doi:10.1037/0003-066X.54.10.817. PMID 10540594.
92. Eysenck, H. J. (1965). *Smoking, health and personality*. New York: Basic Books.
93. Ozga-Hess, Jenny E.; Romm, Katelyn F.; Felicione, Nicholas J.; Dino, Geri; Blank, Melissa D.; Turiano, Nicholas A. (1 September 2020). "Personality and impulsivity as predictors of tobacco use among emerging adults: A latent class analysis". *Personality and Individual Differences.* 163: 110076. doi:10.1016/j.paid.2020.110076. ISSN 0191-8869.

94. "Summaries for patients. Primary care interventions to prevent tobacco use in children and adolescents: U.S. Preventive Services Task Force recommendation statement". *Ann. Intern. Med.* 159 (8): 1–36. 2013. doi:10.7326/0003-4819-159-8-201310150-00699. PMID 23974179.
95. Chamberlain, Catherine; O'Mara-Eves, Alison; Porter, Jessie; Coleman, Tim; Perlen, Susan M.; Thomas, James; McKenzie, Joanne E. (2017). "Psychosocial interventions for supporting women to stop smoking in pregnancy". *The Cochrane Database of Systematic Reviews*. 2: CD001055. doi:10.1002/14651858.CD001055.pub5. ISSN 1469-493X. PMC 4022453. PMID 28196405.
96. Stead, Lindsay F; Koilpillai, Priya; Fanshawe, Thomas R; Lancaster, Tim (24 March 2016). "Combined pharmacotherapy and behavioural interventions for smoking cessation". *Cochrane Database of Systematic Reviews*. 3: CD008286. doi:10.1002/14651858.cd008286.pub3. ISSN 1465-1858. PMID 27009521.

Додатки

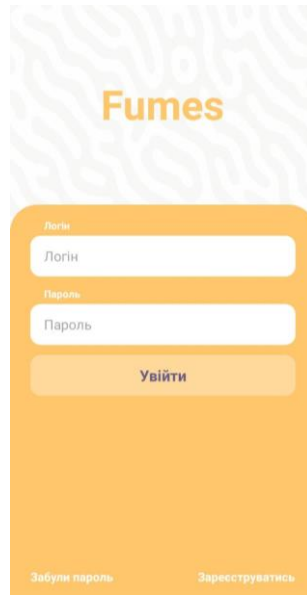


Рис. 3.8 — Екран входу

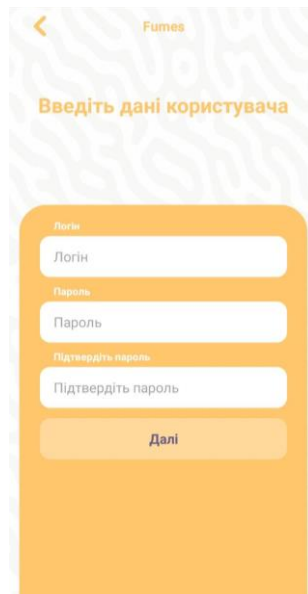


Рис. 3.9 — Екран реєстрації

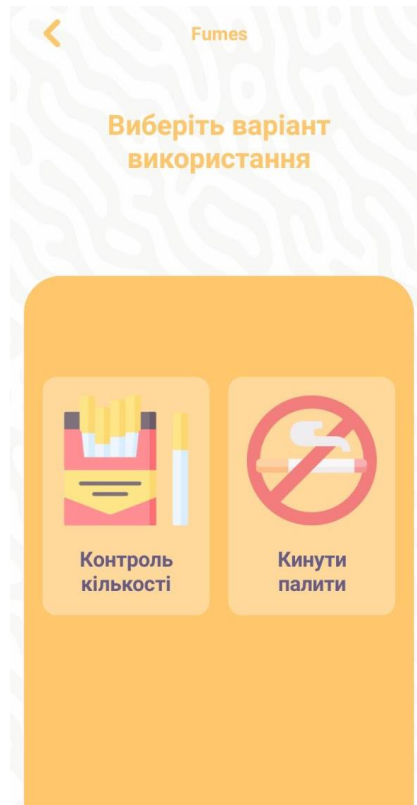


Рис. 3.10 — Екран вибору варіанту використання



Рис. 3.11 — Екран вибору пристрою



Рис. 3.12 — Экран статистики

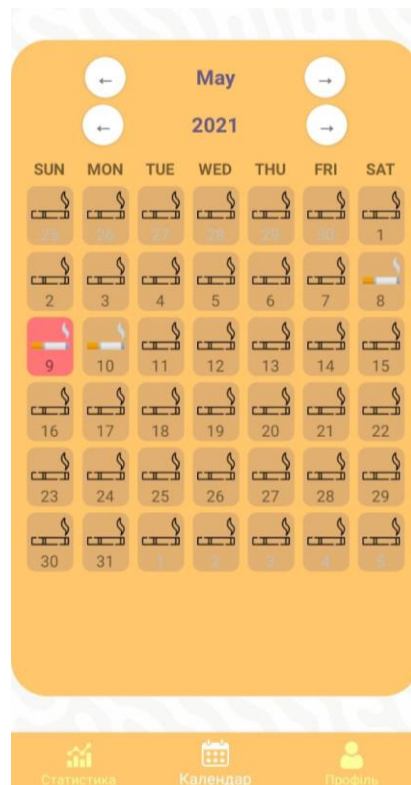


Рис. 3.13 — Экран календаря

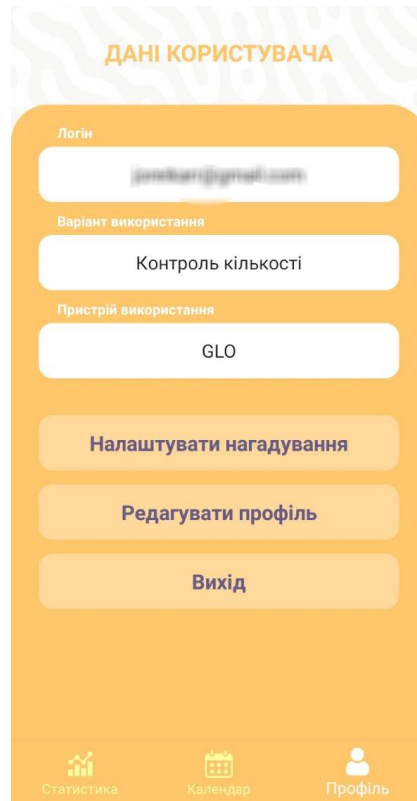


Рис. 3.14 — Екран профілю



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
 ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробки мобільного додатку для моніторингу стану курця
мовою С#.

Виконав студент 4 курсу
групи ПД-44
Читулян Вадим олегович
Керівник роботи
Негоденко Олена Василівна

Київ – 2021

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- Мета роботи - розробка програмного забезпечення для споживачів тютюнових виробів або сумішів для куріння.
- Об'єкт дослідження - покращення процесу моніторингу стану курця.
- Предмет дослідження - додаток для людей, що споживають тютюнові вироби або суміші для куріння.

АНАЛОГИ



Інтерфейс Kwit



Інтерфейс додатку Gotta Smoke?



Інтерфейс додатку «Моя остання сигарета»



Інтерфейс додатку «Не курю»

3

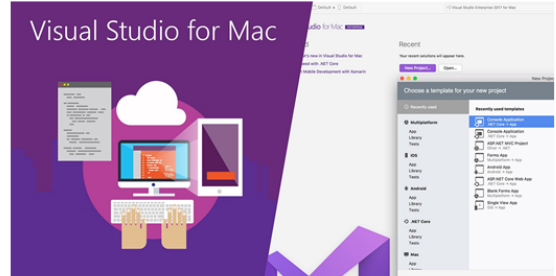
ТЕХНІЧНІ ЗАВДАННЯ

Додаток має виконувати наступні функції :

- Можливість авторизуватися в системі;
- Можливість відновлення даних аккаунта;
- Наявність нагадування в виді пуш-нотифікацій;
- Можливість редагувати частоту надсилання та кількість нотифікацій;
- Збір та збереження даних користувача;
- Конвертація даних в статистичний графік;
- Можливість зміни пароля та пристрою в профілі користувача;

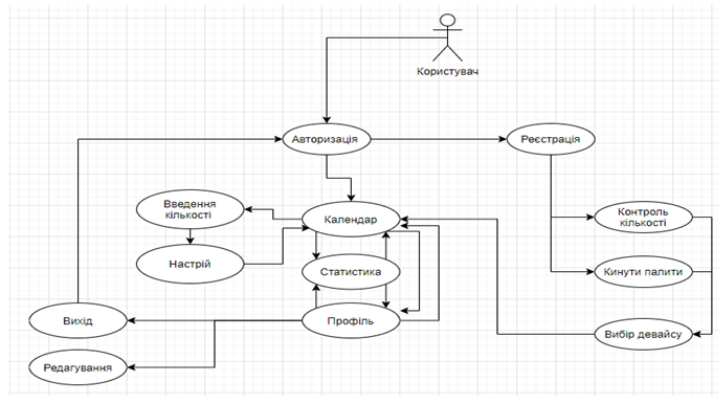
4

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



5

Діаграма користувача



6

АПРОБАЦІЇ

Результати дослідження бакалаврської роботи апробовані:

1. Застосування програмного забезпечення в інфокомунікаційних технологіях // Розробка програмного забезпечення для інфокомунікацій. Збірник тез. 12.02.2021, ДУТ, м. Київ — К.: ДУТ, 2021. — С. 123 – 124.
2. II Всеукраїнська науково-технічна конференція «Сучасний стан та перспективи розвитку ІОТ». Збірник тез. 09.04.2021, ДУТ, м. Київ — К.: ДУТ, 2021.

7

ВИСНОВКИ

Виявлено недоліки аналогових програм та розроблено додаток, який буде перевершувати наявне програмне забезпечення зі схожою структурою. А саме спрощення та пришвидшення моніторингу стану курця.

В перспективі даний додаток може очікувати наступне:

1. Оптимізація;
2. Збільшення кількості тестів спрямованих на моніторинг стану емоційного інтелекту курця.
3. Покращення безпеки, шляхом підключення смс-сервісів.
4. Розміщення додатку на google market та app store.

Технічне завдання виконане досконало

Всі поставлені вимоги виконано.

8

ДЯКУЮ ЗА УВАГУ!