

Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці

ДО ЗАХИСТУ ДОПУЩЕНО
В.о. завідувача кафедри
_____Олександр КОВАЛЬ
«_____» _____ 2024 р.

Київ – 2024

Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

Рівень вищої освіти другий (магістерський)

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр КОВАЛЬ

(підпис)

« ____ » _____ 2024 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Славському Святославу Володимировичу

(прізвище, ім'я, по батькові)

1. Тема дисертації _____

Інструментальні засоби формування поведінки на дорозі

Науковий керівник Гусєва Ірина Ігорівна, к.е.н, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «06» листопада 2023 року № 5152-с

2. Строк подання студентом дисертації 10 січня 2024 року

3. Об'єкт дослідження формування звичок та вивчення правил дорожнього руху у водіїв автомобілів

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за ОПП) програмне забезпечення для формування поведінки на дорозі на основі вивчення правил дорожнього руху та сформованої статистики поїздки

5. Перелік питань, які потрібно розробити _____

— провести дослідження та аналіз наявних засобів формування знань та навичок їзди.

— розробити модуль теоретичної підготовки для вивчення правил дорожнього руху.

— розробити модуль практичної підготовки на основі даних, отриманих під час поїздки.

— розробити алгоритм визначення рейтингу поведінки.

— протестувати розроблений програмний продукт.

6. Орієнтований перелік ілюстративного матеріалу діаграми системи, алгоритм формування рейтингу, інтерфейс програмного забезпечення

7. Орієнтований перелік публікацій

Сучасні проблеми наукового забезпечення енергетики. У 2-х т.: матеріали XX Міжнар. наук.-практ. конф. молод. вчених і студ. (присвячена 125-річчю КПІ ім. Ігоря Сікорського та 90-річчю НН ІАТЕ (ТЕФ)), м. Київ, 25–28 квіт. 2023 р. – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2023. – Т. 2. – 269 с.

8. Дата видачі завдання «01» листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	01.11.2022	виконано
2	Дослідження предметної області	01.11.2022 - 01.12.2022	виконано
3	Постановка вимог до проектування системи	01.12.2022 - 15.12.2022	виконано
4	Дослідження існуючих рішень	15.12.2022 - 01.01.2023	виконано
5	Підготовка публікацій	02.01.2023 - 08.01.2023	виконано
6	Розробка програмного продукту	06.02.2023 - 08.10.2023	виконано
7	Тестування	09.10.2023 - 22.10.2023	виконано
8	Захист програмного продукту	23.10.2023 - 27.10.2023	виконано
9	Підготовка магістерської дисертації	30.10.2023 - 20.11.2023	виконано
10	Передзахист	11.12.2023 - 15.12.2023	виконано
11	Захист	15.01.2024 - 19.01.2024	виконано

Студент

(підпис)

Славський С.В

(ім'я, прізвище)

Науковий керівник

(підпис)

Гусєва І.І

(ім'я, прізвище)

РЕФЕРАТ

Структура й обсяг дипломної роботи. Магістерська дисертація складається зі вступу, шести розділів, висновків та двох додатків. Робота містить посилання на 21 джерел та 36 ілюстрацій. Основна частина роботи викладена на 80 сторінках.

Актуальність. Зі збільшенням кількості автомобілів на дорогах збільшується кількість аварій, як у відсоткових, так і в абсолютних показниках. В Україні, щороку в дорожньо-транспортну пригоду потрапляє до 2% водіїв автомобілів, що є доволі суттєвим показником порівняно з іншими країнами Європи. Розробка сучасних інструментальних засобів формування поведінки на дорозі для покращення рівня обізнаності водіїв та зменшення кількості аварій зумовлена необхідністю зменшити кількість дорожньо-транспортних пригод за рахунок нових методів навчання та мотивації водіїв.

Метою роботи є розробка мобільного програмного застосунку формування поведінки на дорозі, для підвищення рівня обізнаності водіїв та зменшення кількості аварій та дорожньо-транспортних пригод, за рахунок кращих знань правил та надбаних звичок безпечного водіння.

Для досягнення поставленої мети необхідно реалізувати наступні **завдання**:

- провести дослідження та аналіз наявних засобів формування знань та навичок їзди;
- розробити модуль теоретичної підготовки для вивчення правил дорожнього руху;
- розробити модуль практичної підготовки на основі даних, отриманих під час поїздки;
- розробити алгоритм визначення рейтингу поведінки;
- протестувати розроблений програмний продукт в умовах реальної експлуатації.

Об’єктом дослідження є формування звичок та вивчення правил дорожнього руху у водіїв автомобілів.

Предметом дослідження є програмне забезпечення для формування поведінки на дорозі на основі вивчення правил дорожнього руху та сформованої статистики поїздки.

Методи дослідження: сукупність теоретичних, а саме аналіз та узагальнення, і емпіричних, а саме експеримент та компонентний аналіз, методів.

Практичне значення одержаних результатів дослідження полягає у тому, що розроблений програмний застосунок на базі сенсорів мобільного пристрою та розробленого алгоритму покращить рівень обізнаності правил, та замотивує водіїв дотримуватись правил дорожнього руху.

Апробація результатів дисертації.

Основні положення та висновки роботи викладені та обговорені на:

1. XX-й міжнародна науково-практична конференція молодих вчених та студентів "Сучасні проблеми наукового забезпечення енергетики" 2023.

Публікації. Зміст і результати дослідження відображено у публікації:

Інструментальні засоби формування поведінки на дорозі. Сучасні проблеми наукового забезпечення енергетики. У 2-х т.: матеріали XX Міжнар. наук.-практ. конф. молод. вчених і студ. (присвячена 125-річчю КПІ ім. Ігоря Сікорського та 90-річчю НН ІАТЕ (ТЕФ)), м. Київ, 25–28 квіт. 2023 р. – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2023. – Т. 2. – 269 с.

Ключові слова: дорожньо-транспортна пригода, інструментальні засоби, правила дорожнього руху, GPS, екстрене гальмування

ABSTRACT

Structure and scope of the thesis. The master's thesis consists of an introduction, six chapters, conclusions and two appendices. The work contains references to 21 sources and 36 illustrations. The main part of the work is set out on 80 pages.

Relevance. With the increase in the number of cars on the roads, the number of accidents increases, both in percentage and absolute terms. In Ukraine, up to 2% of car drivers are involved in a road accident every year, which is quite a significant figure compared to other European countries. The development of modern tools for shaping road behaviour to improve driver awareness and reduce the number of accidents is driven by the need to reduce the number of road accidents through new methods of training and motivation of drivers.

The aim of the work is to develop a mobile software application for shaping road behaviour to increase driver awareness and reduce the number of accidents and road accidents through better knowledge of the rules and acquired safe driving habits.

In order to achieve this goal, the following tasks need to be implemented:

- conduct research and analysis of existing means of forming knowledge and driving skills;
- to develop a theoretical training module for studying traffic rules;
- to develop a practical training module based on the data obtained during the trip;
- develop an algorithm for determining the behaviour rating;
- to test the developed software product in real-world conditions.

The object of the study is the formation of habits and learning of traffic rules among car drivers.

The subject of the study is software for shaping road behaviour based on the study of traffic rules and generated trip statistics.

Research methods: a combination of theoretical, namely analysis and generalisation, and empirical, namely experiment and component analysis, methods.

The practical significance of the research results is that the developed software application based on the sensors of a mobile device and the developed algorithm will improve the level of awareness of the rules and motivate drivers to comply with traffic rules.

Testing of the results of the dissertation.

The main provisions and conclusions of the work are presented and discussed at:

1. XX International Scientific and Practical Conference of Young Scientists and Students "Modern Problems of Scientific Support of Energy" 2023.

Publications. The content and results of the study are reflected in the publication:

Instrumental means of forming road behaviour. Modern problems of scientific support of energy. In 2 volumes: materials of the XX International Scientific and Practical Conf. of Young Scientists and Students (dedicated to the 125th anniversary of Igor Sikorsky Kyiv Polytechnic Institute and the 90th anniversary of the ER IAE (TEF)), Kyiv, 25-28 April 2023 - Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, Polytechnic Publishing House, 2023. - Vol. 2. - 269 p.

Keywords: road traffic accident, tools, traffic rules, GPS, emergency braking

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
1 ПОСТАНОВКА ЗАВДАННЯ ФОРМУВАННЯ ПОВЕДІНКИ НА ДОРОЗІ	12
1.1 Мета роботи та формулювання завдань.....	12
1.2 Призначення системи.....	13
1.3 Реалізація алгоритму визначення рейтингу поведінки	13
1.4 Програмна реалізація системи	14
Висновки до розділу 1	14
2 ДОСЛІДЖЕННЯ АНАЛОГІЧНИХ РІШЕНЬ.....	16
2.1 Мобільні застосунки для вивчення правил дорожнього руху.....	16
2.2 Мобільні застосунки для визначення емпіричних даних поїздки	19
2.3 Існуючі системи мотивування користувачів	21
Висновки до розділу 2	23
3 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	24
3.1 Обґрунтування вибору мови програмування	24
3.2 Обґрунтування використання середовища розробки	26
3.3 Обґрунтування використання бази даних	28
3.4 Обґрунтування використання сторонніх бібліотек	30
Висновки до розділу 3	32
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ЗАСТОСУНКУ	33
4.1 Діаграма прецедентів	33
4.2 Функціональна декомпозиція	34

4.3 Алгоритм формування якості поведінки водія	35
4.4 Структура бази даних	38
Висновки до розділу 4	49
5 ВИКОРИСТАННЯ ЗАСТОСУНКУ КОРИСТУВАЧЕМ	50
5.1 Системні вимоги застосунку	50
5.2 Встановлення мобільного застосунку	50
5.3 Запуск і користування мобільним застосунком	50
5.4 Система мотивації користувачів	60
Висновки до розділу 5	61
6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	62
6.1 Опис ідеї проєкту	62
6.2 Технологічний аудит ідеї проєкту	65
6.3 Аналіз ринкових можливостей запуску стартап-проєкту	66
6.4 Розроблення ринкової стратегії проєкту	73
Висновки до розділу 6	75
ВИСНОВКИ	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
ДОДАТКИ	81
ДОДАТОК А Лістинг розробленої програми	81
ДОДАТОК Б Апробація	99
ДОДАТОК В Презентація	103

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

ДТП — Дорожньо-транспортна пригода;

GPS — Система глобального позиціювання;

ПДР — Правила дорожнього руху;

JVM — Java Virtual Machine, віртуальна машина для виконання байт-коду Java;

IDE — Integrated Development Environment, інтегроване середовище розробки;

API — Application programming interface, інтерфейс прикладного програмування.

ВСТУП

З активним розвитком технологій людство все більше починає покладатись на працю машин та техніки, що зменшує людський фактор впливу, проте збільшує негативні наслідки можливих помилок системи. З початку 20 століття автомобілі стрімко увійшли в повсякденне життя і зараз не можливо уявити собі велике місто, по якому не проїжджають тисячі, або навіть мільйони транспортних засобів, зі швидкістю, що набагато перевищує людські можливості. Автомобільний трафік спростив життя людей, але разом з тим приніс немало небезпеку, як для самих водіїв, так і для звичайних громадян, що можливо ніколи в своєму житті не сиділи за кермом, цього транспорту.

Правила дорожнього руху, які з'являлися з розвитком і автомобілів, і інфраструктури для них, часто були написані внаслідок, дорожньо-транспортних пригод, а саме тому їх дотримання є таким важливим для людей.

Дорожньо-транспортна пригода (ДТП) — подія, що сталася під час руху транспортного засобу, внаслідок якої загинули або поранені люди чи завдані матеріальні збитки.

Офіційна статистика дорожньо-транспортних пригод в Україні зазначає, що кожного року відбувається близько 150000 ДТП, які були задокументовані та офіційно оформленні, кількість неоформлених невідома і може перевищувати відому кількість в декілька разів [1].

Якщо врахувати кількість автомобілів в Україні, а це трохи більше 10 млн. можна дізнатись, що кожного року кожен 50 водій потрапляє в ДТП. Переважна більшість дорожніх пригод відбувається через недотримання правил дорожнього руху, через їх незнання або самовпевненість водіїв.

Основною метою розробки інструментальних засобів формування поведінки на дорозі є створення мобільного застосунка для щоденного використання, яке покращить теоретичні знання правил, а також практичні навички водіння за рахунок накопичення даних поведінки водія на дорозі та системи заохочення

вивчення та повторення базових, а також розширених правил дорожнього руху та поведінки за кермом авто [2].

Поведінка на дорозі це сукупність роботи кожного її учасника, а отже, формувати цю поведінку необхідно у кожній людині, не зважаючи на те чи це досвідчений водій, чи новачок, який тільки отримав права, чи пішохід, який переходить дорогу.

Отже, за рахунок інструментальних засобів, що розробляються, можливо досягти покращення знань правил дорожнього руху, їх щоденне дотримання та в перспективі сформувати принципи та засади поведінки на дорозі, що унеможливить майбутні нещасні випадки та смерті.

1 ПОСТАНОВКА ЗАВДАННЯ ФОРМУВАННЯ ПОВЕДІНКИ НА ДОРОЗІ

1.1 Мета роботи та формулювання завдань

Метою роботи є розробка мобільного програмного застосунку формування поведінки на дорозі для підвищення рівня обізнаності водіїв та зменшення кількості аварій та дорожньо-транспортних пригод за рахунок кращих знань правил та надбаних звичок безпечного водіння.

Об'єкт дослідження: формування звичок та вивчення правил дорожнього руху у водіїв автомобілів.

Предмет дослідження: програмне забезпечення для формування поведінки на дорозі на основі вивчення правил дорожнього руху та сформованої статистики поїздки.

Для досягнення поставленої мети необхідно реалізувати наступні завдання.

1. Провести дослідження та аналіз наявних засобів формування знань та навичок їзди.
2. Розробити модуль теоретичної підготовки для вивчення правил дорожнього руху.
3. Розробити модуль практичної підготовки на основі даних, отриманих під час поїздки.
4. Розробити алгоритм визначення рейтингу поведінки.
5. Розробити базу даних системи.
6. Протестувати розроблений програмний продукт в умовах реальної експлуатації.

Вхідні дані: дані сенсора GPS в мобільному пристрої, часові дані, правила дорожнього руху, відповідно до Закону України «Про дорожній рух» [3].

Вихідні дані: якісна та кількісна статистика проходження тестів ПДР, сформована статистика кожної поїздки, сформована статистика якості останніх поїздок.

Користувачі: водії транспортних засобів, що бажають оновити знання про правила дорожнього руху та покращити стиль своєї їзди, слухачі курсів водіння.

1.2 Призначення системи

Розроблена програмна система призначена для комплексного формування поведінки усіх учасників дорожнього руху за рахунок кращих знань правил та надбаних звичок безпечного водіння.

Додатковим призначенням мобільного забезпечення є механізми додаткової стимуляції користувачів до використання програмного забезпечення для підтримання високого рівня практичних та теоретичних знань.

Програмне забезпечення буде корисне водіям транспортних засобів з будь-яким рівнем знань та навичок, вільним слухачам курсів водіння, а також пасажирам транспортних засобів, які хочуть переконатись в тому, що їх поїздка проходить безпечно та без порушень правил дорожнього руху.

1.3 Реалізація алгоритму визначення рейтингу поведінки

Ідеєю алгоритму є створення збалансованої рейтингової системи, яка буде показувати рівень теоретичних знань та практичних навичок водіння користувача. Алгоритм зміни рейтингу повинен віддавати перевагу даним зібраним під час поїздки, оскільки саме практичні навички сильніше впливають на ймовірність потрапити в дорожньо-транспортну пригоду.

Додатково розроблений алгоритм повинен стимулювати користувачів покращувати свої знання та вміння за рахунок гейміфікації процесу користування програмним забезпеченням.

1.4 Програмна реалізація системи

Програмний продукт повинен складатись з мобільного застосунку, який містить вбудовану базу даних.

Функції програмного забезпечення, які повинні бути реалізовані.

1. Збереження та відображення правил дорожнього руху.
2. Збереження та відображення тестів на знання правил дорожнього руху.
3. Отримання поточної швидкості та прискорення за допомогою GPS.
4. Розрахунок середньої, а також максимальної швидкості під час руху та за результатами поїздки.
5. Розрахунок відстані, кількості перевищень швидкості та екстрених гальмувань за результатами кожної поїздки.
6. Визначення поточного рейтингу водіння на основі пройдених тестів та поїздок.
7. Фіксування поточного відрізка користування програмним забезпеченням.
8. Візуалізація статистики поїздок та проходження тестування.

Висновки до розділу 1

За результатами дослідження проблем формування поведінки на дорозі було визначено мету роботи, сформовано об'єкт та предмет дослідження.

Поставлено завдання, які необхідно реалізувати.

1. Провести дослідження та аналіз наявних засобів формування знань та навичок їзди.
2. Розробити модуль теоретичної підготовки для вивчення правил дорожнього руху.
3. Розробити модуль практичної підготовки на основі даних, отриманих під час поїздки.

4. Розробити алгоритм визначення рейтингу поведінки.
5. Розробити базу даних системи.
6. Протестувати розроблений програмний продукт в умовах реальної експлуатації.

Метою роботи є розробка мобільного програмного застосунку формування поведінки на дорозі для підвищення рівня обізнаності водіїв

Отже, необхідно розробити мобільний застосунок, який допоможе покращити знання правил дорожнього руху водіїв, а також сформує статистику їх якості їзди. Розроблений програмний продукт повинен бути простим у користуванні, мати інтуїтивно зрозумілий інтерфейс та заохочувати користувачів користуватись ним на повсякденній основі.

2 ДОСЛІДЖЕННЯ АНАЛОГІЧНИХ РІШЕНЬ

2.1 Мобільні застосунки для вивчення правил дорожнього руху

На цей момент існує велика кількість різноманітних мобільних застосунків, вебпорталів, десктопних програмних продуктів, які допомагають вивчити правила дорожнього руху, як за рахунок представлення переліку правил, так і за рахунок тестів на перевірку знань цих правил.

Електронні джерела разом з програмним забезпеченням для комп'ютерів, зазвичай, мають і перелік правил дорожнього руху, і вагому базу тестів, які допоможуть підготуватись до складання іспиту, але часто не дають достатньої, навіть теоретичної підготовки для повноцінного керування транспортним засобом в умовах сучасного урбаністичного світу. Беззаперечно, що повнота інформації представлена на різних ресурсах може відрізнятись, а також не завжди є зручною для користування у повсякденному житті.

Найбільш популярними ресурсами для вивчення правил та підготовки до складання іспитів ПДР є такі ресурси:

- 1) vodiу.ua (рис. 2.1) [4];
- 2) green way (рис. 2.2) [5].

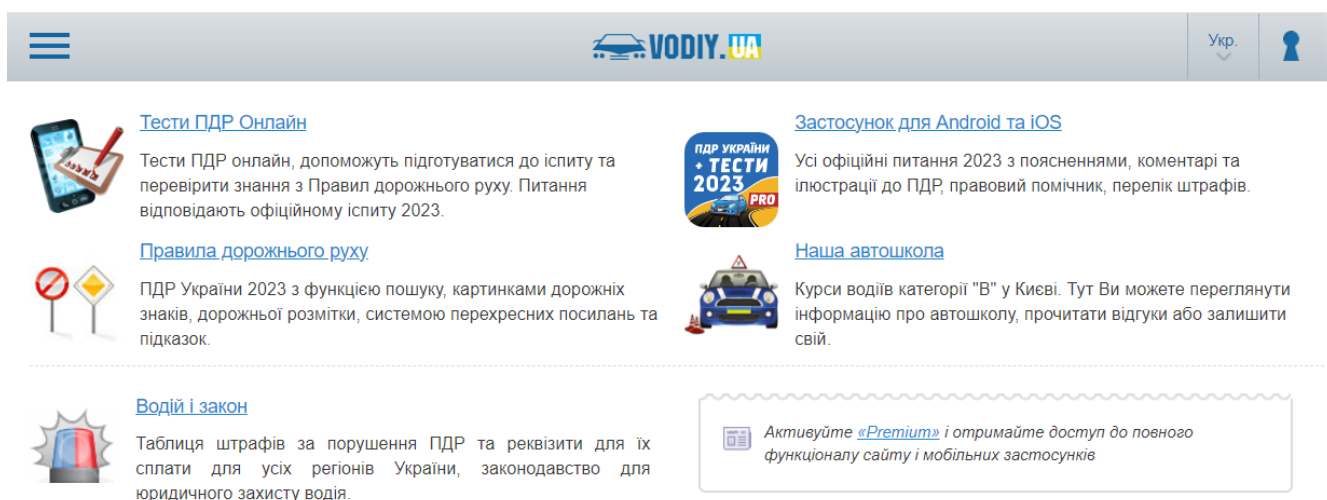


Рисунок 2.1 – Сайт vodiу.ua

Вебзастосунок «vodi.ua» пропонує онлайн підготовку та тестування водіїв. Надає можливість скористатися тестами ПДР онлайн, щоб підготуватися до іспиту та перевірити знання з правил дорожнього руху. Сайт також містить правовий помічник, перелік штрафів, а також правила дорожнього руху України зразка 2023 року з функцією пошуку.

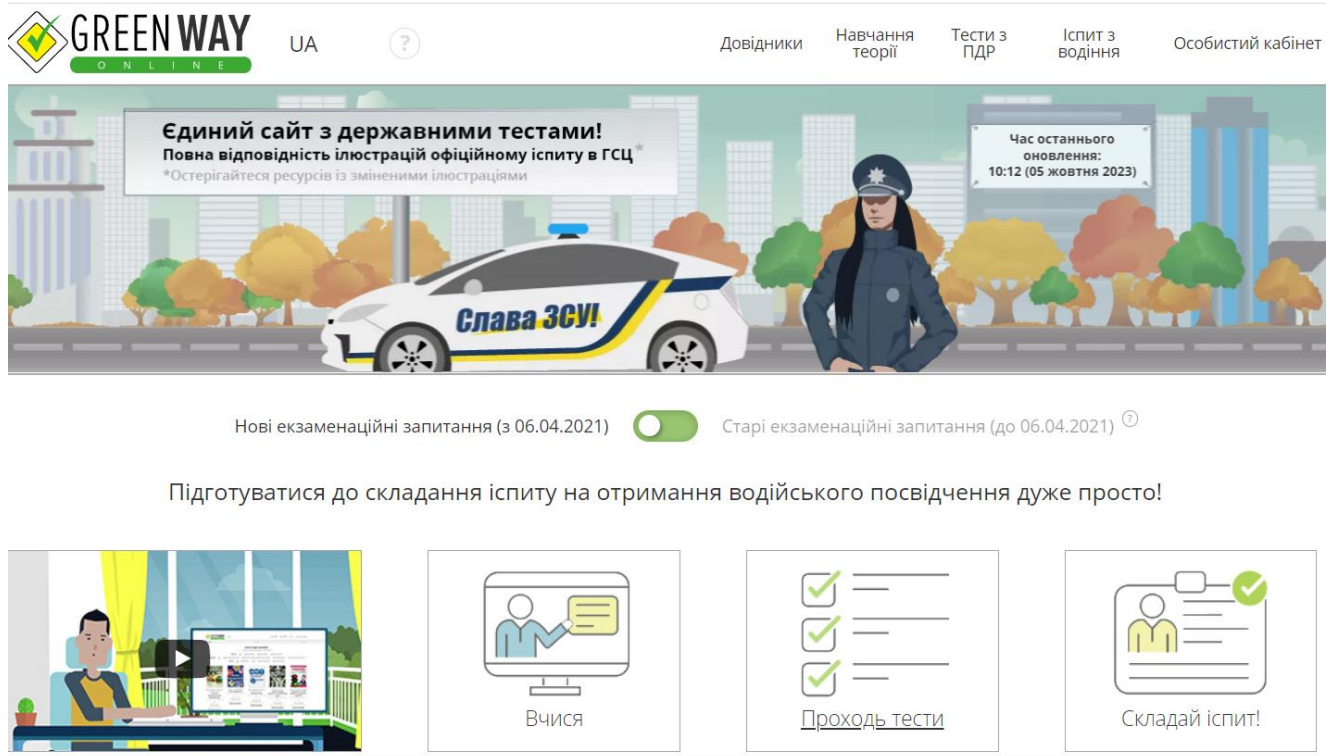


Рисунок 2.2 – Сайт green-way.com.ua

Вебзастосунок «green-way.com.ua» містить онлайн-платформу з тестами Правил дорожнього руху України, які на 100% відповідні офіційним вимогам Головного сервісного центру МВС України за 2023 рік. Використовуючи матеріали, що пропонуються, можна підготуватися до теоретичної частини іспиту на отримання права керування транспортними засобами.

У більшості електронних джерел з вивчення ПДР є мобільні застосунки, які частково чи повністю відтворюють функціонал сайтів. Окрім застосунків, що дублюють офіційні сайти, також є безліч інших мобільних застосунків, які допомагають з вивченням загальних правил чи поглибленню у певні теми.

Найбільш популярні мобільні застосунки по вивченню ПДР:

- 1) «Тести ПДР» від green-way (рис. 2.3);
- 2) «Знаки ПДР» (рис. 2.4).

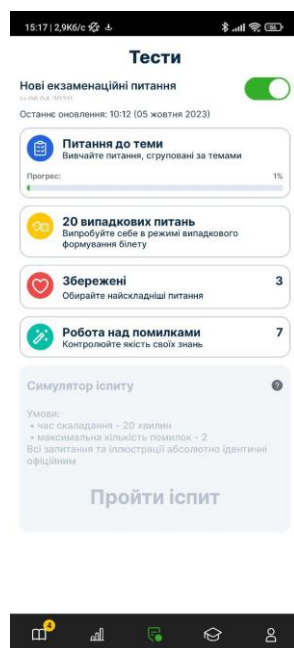


Рисунок 2.3 – Застосунок «Тести ПДР»

Застосунок «Тести ПДР» має базу даних тестів правил дорожнього руху аналогічну сайту green-way.com.ua, проте функціонал обмежений тільки тестами й не має повного переліку правил дорожнього руху



Рисунок 2.4 – Застосунок «Знаки ПДР»

Застосунок «Знаки ПДР» направлений на більш детальне вивчення знаків дорожнього руху, має покращені тести на дану тему, що дозволяє більш якісно вивчити її, проте він є вузькопрофільним, а отже, не має тестів до інших тем, та загального переліку правил дорожнього руху.

Основним недоліком розглянутих програмних рішень є відсутність системи мотивування користуватись ними після складання іспитів для отримання прав керування транспортним засобом.

2.2 Мобільні застосунки для визначення емпіричних даних поїздки

Оскільки поведінка на дорозі характеризується не тільки знанням правил та їх дотриманням, а ще і якісними та кількісними показниками кожної поїздки, для формування звичок та навичок водіння необхідно досліджувати поїздки водіїв, щоб знаходити певні патерни поведінки, які впливають на якість водіння.

Найбільш популярні пристрої для визначення емпіричних даних є спідометр, одометр та GPS пристрій.

Одометр — пристрій для вимірювання відстані, що подолав транспортний засіб.

Спідометр — пристрій для вимірювання миттєвої швидкості руху транспортного засобу.

GPS — сукупність радіоелектронних засобів, що дозволяє визначати положення та швидкість руху об'єкта на поверхні Землі

Серед безлічі аналогічних мобільних застосунків характерною популярністю користуються такі програми:

- 1) «GPS спідометр: одометр» (рис. 2.5) [6];
- 2) «Спідометр: GPS Speed Tracker» (рис. 2.6) [7].

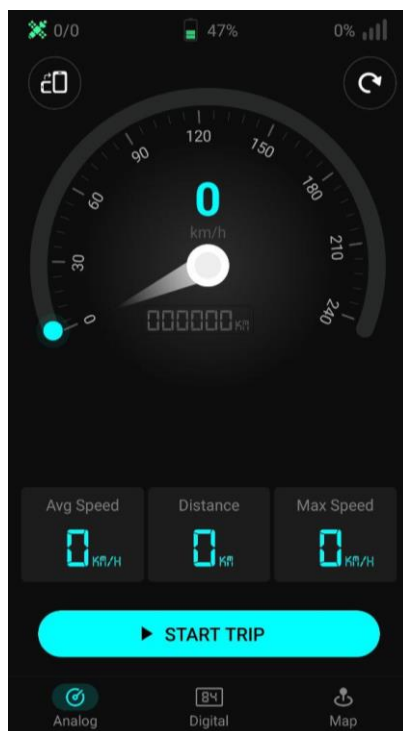


Рисунок 2.5 – Застосунок «GPS спідометр: одометр»

Застосунок «GPS спідометр: одометр» характерний можливістю зміни режиму транспортного засобу, такі як: велосипед, автомобіль, ходьба. Вираховує дистанцію, максимальну швидкість їзди та середню.



Рисунок 2.6 – Застосунок «Спідометр: GPS Speed Tracker»

Мобільний застосунок «Спідометр: GPS Speed Tracker» характеризується покращеною статистикою поїздки, до якої входять такі показники: тривалість поїздки, дистанція, максимальна швидкість, середня швидкість. А також інтегрована карта від Google для побудови маршруту та автоматичного визначення точок початку та кінця поїздки.

Основним недоліком існуючих застосунків є відсутність засобів, які необхідні для визначення якості водіння, а саме кількість перевищень швидкості руху, та кількість екстрених гальмувань, що відбулися за час поїздки.

2.3 Існуючі системи мотивування користувачів

Одним з головних завдань формування поведінки є створення системи мотивації користувачів щоденно використовувати програмне забезпечення для покращення своїх знань, а також недопущення нехтувань цими правилами під час щоденних поїздок за кермом.

Як показує світова практика, мотивацією користувачів до повторного використання програмного продукту при умові відсутності критичної потреби, є гейміфікація процесу користування програмним продуктом.

Гейміфікація представляє собою використання ігрових методів у контекстах, які не є ігровими, з метою мотивації та залучення користувачів. З точки зору програмного дизайну, гейміфікація включає ігрові елементи у неігрові програми для поліпшення користувацького досвіду та привертання уваги користувачів [8]. Цей підхід надає інтерактивний досвід навчання, який може стимулювати користувачів до досягнення оптимальних результатів

У водінні, особливо в контексті гейміфікації, методи зворотного зв'язку є ефективними для стимулювання молодих водіїв до прийняття безпечних практик за кермом.

В контексті водіння гейміфікація може створити «потік», що визначається як внутрішній стан мотивації та задоволення, збільшуючи рівень виклику в рутинних

завданнях водіння. Повторювана практика, яка використовує ігрові елементи, вважається ключовим фактором успішності гейміфікації. З цієї причини деякі дослідження, зазвичай оцінюють ефективність гейміфікації в різні моменти часу, запрошуючи учасника пройти однаковий маршрут кілька разів і оцінювати його ефективність та водійську поведінку кожного разу [9].

Зазвичай процес гейміфікації в багатьох програмних продуктах полягає у створенні штучної конкуренції між усіма користувачами програмних продуктів. Інколи це приводить до очікуваного результату і люди починають більше часу приділяти корисному застосунку, для отримання ефемерного визнання серед інших користувачів, проте не всі користувачі отримують мотивацію під час змагання з іншими людьми, яких він не знає.

З іншого боку можна розглянути процес отримання певних нагород, які будуть доступні тільки користувачу програмного забезпечення. Цей підхід побудований на бажанні людей покращувати свої особисті досягнення, не зважаючи на те в якій області ці досягнення відбуваються. Даний підхід може бути стимулом для людини, яка знає достатню інформацію покращувати свої знання та навички, не маючи на це критичної необхідності.

Одним з прикладом гейміфікації процесу навчання, який не відносить до описаних вище, є система створення нескінченної послідовності занять, завдань чи іншого, для постійного повернення користувача до програного забезпечення. Прикладом такого підходу є застосунок по вивченню мов Duolingo. Він створює послідовність з неперервної кількості днів, коли людина вивчає мову, чим більша ця послідовність, тим більша мотивація у людини позайматись, щоб не втратити її. Реалізація еталонної послідовності від Duolingo продемонстрована на рисунку 2.7

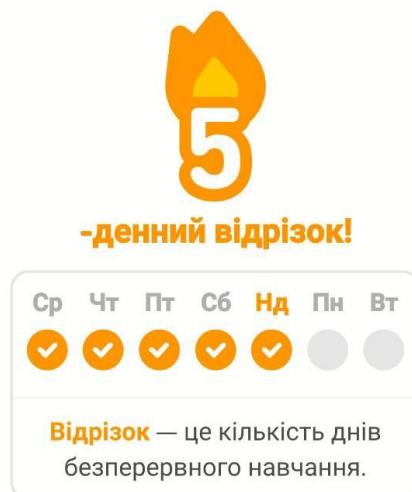


Рисунок 2.7 – Нескінченна послідовність Duolingo

Система мотивації Duolingo разом з якісним продуктом показують результати, понад 60 млн. активних користувачів щоденно [10].

Висновки до розділу 2

Враховуючи проведений аналіз програмних продуктів зі схожими функціями, можна зробити висновки, що на даний момент часу кількість готових рішень, які б підвищували рівень поведінки на дорозі за рахунок теоретичних та практичних даних, є недостатнім. Всі розглянуті програмні продукти виконують поставлене завдання частково, спираючись тільки на одну з гілок покращення навичок, а отже, створення програмного продукту, який зможе поєднати кращі практики обох методів та запропонувати поглиблений підхід до дотримання правил, є актуальною темою, що потребує реалізації, оскільки має попит на ринку мобільних програм. Процес гейміфікації програмного продукту є актуальною тенденцією на даний момент і має свої переваги та недоліки, але він безпосередньо впливає на залучення людей до програмних продуктів.

3 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування вибору мови програмування

Для реалізації поставленої мети було проведено порівняння найбільш актуальних мов програмування мобільних застосунків. Серед таких мов як: Java, Swift, Flutter та Kotlin, для розробки застосунка було обрано останню.

Kotlin є однією з найбільш динамічних і затребуваних мов програмування. Kotlin поєднує в собі найкращі можливості об'єктно-орієнтованого та функціонального програмування. В останні роки розробники програмного забезпечення в усьому світі віддають перевагу Kotlin для різноманітних проєктів [11].

Kotlin - це статично типізована мова програмування з відкритим вихідним кодом, яка підтримує як об'єктно-орієнтоване, так і функціональне програмування. Kotlin має схожий синтаксис та концепції з іншими мовами, включаючи C#, Java та Scala, серед багатьох інших. Kotlin збирає кращі практики з інших мов програмування. Вона існує у варіантах, орієнтованих на JVM (Kotlin/JVM), JavaScript (Kotlin/JS) та нативний код (Kotlin/Native).

Мова Kotlin знаходиться під управлінням Kotlin Foundation, групою, створеною JetBrains і Google, яка займається просуванням і постійним розвитком мови. Kotlin офіційно підтримується Google для розробки Android, що означає, що документація та інструментарій для Android розроблені з урахуванням особливостей Kotlin.

У травні 2017 року Google оголосив Kotlin офіційною мовою розробки застосунків для Android.

Kotlin сумісна з Java, що означає, що обидві мови можуть обмінюватися інформацією та використовувати її. Як і компілятор Java, компілятор Kotlin генерує байт-код, який можна виконати на JVM.

Kotlin широко використовується для розробки серверних застосунків, застосунків для Android, кросплатформних мобільних застосунків, тощо. Бібліотеки Java можна використовувати в проєкті Kotlin. Незважаючи на те, що Java більш розповсюджена мова програмування, розробники Kotlin більше зосереджуються на спрощенні коду та підвищенні прозорості [12].

Існує багато особливостей мови Kotlin, через яку її вважають однією з найкращих мов для розробки мобільних програм, деякі з яких наведено нижче.

1. Мова Kotlin може використовуватися на кількох платформах, зменшуючи час, витрачений на написання та підтримку коду на різних платформах. Це не залежить від віртуальної машини на цільовій платформі.

2. Kotlin має такі сучасні функції, як нульова безпека, функція, яка допомагає пом'якшити проблеми, пов'язані з нульовими посиланнями. Kotlin підтримує такі функції, як лямбда-функції (анонімна функція) та інтелектуальне переведення (використовується для додавання умов у операторах if).

3. Kotlin має багато розширень, які дозволяють додавати функціональність без успадкування класу за допомогою шаблону декоратора. Ця функція робить код легким для підтримки та читання.

4. У Kotlin функції можна передати як параметри функції, тобто функція може діяти як змінна, і, як і змінна, функція також може бути передана. Вона також відома, як функція верхнього рівня.

5. Kotlin повністю сумісний із Java, має можливість конвертувати файли Java у файли Kotlin лише за допомогою сценарію. Він також працює з бібліотеками JVM і Java, які добре працюють у Kotlin.

6. Kotlin має клас даних. Рівень даних — це рівень, який не виконує жодних операцій і лише зберігає стан. Перевагою цього класу є те, що він забезпечує самогенерований код і уникає реалізації методів усередині класу [13].

Як і більшість сучасних мов, Kotlin взяв найкращі практики зі старих мов та додав до них нові можливості, щоб покращити процес розробки програмного продукту.

Підсумовуючи, можна сказати, що ця мова програмування найкраще підходить для вирішення поставлених завдань, завдяки своїй сумісності з Java, а також тому, що типи даних за замовчуванням обов'язково повинні бути не nullable, тобто вони не можуть приймати значення null. Мова є захищено від NullPointerException, а ця помилка є найпоширенішою причиною збоїв мобільних програм. Крім того, код написаний на мові Kotlin, набагато компактніше за Java код – на 30-40%, що дозволяє збільшити швидкість розробки програмного продукту.

3.2 Обґрунтування використання середовища розробки

Для досягнення поставленої мети було вирішено використовувати середовище розробки Android Studio, як одне з найкращих рішень, призначених для розробки програмного забезпечення для мобільних пристроїв.

Android Studio – це офіційне інтегроване середовище розробки (IDE) для розробки застосунків для Android [14]. Засноване на потужному редакторі коду та інструментах розробки IntelliJ IDEA, Android Studio пропонує ще більше можливостей для підвищення продуктивності при створенні Android-застосунків, які наведені далі.

1. Гнучка система збірки на основі Gradle.
2. Уніфіковане середовище розробки для всіх пристроїв Android.
3. Редагування компонентів з оновленням в емуляторах та фізичних пристроях в режимі реального часу.
4. Комплексні інструменти та фреймворки для тестування.
5. Інструменти Lint для відстеження продуктивності, зручності користування, сумісності версій та інших проблем [15].
6. Інтелектуальний редактор коду, що забезпечує просунуте завершення коду, рефакторинг та аналіз коду під час виконання.
7. Підтримка шаблонів коду

8. Вбудований емулятор Android, який підтримує тестування програмного забезпечення на багатьох пристроях, включаючи телефони, планшети, телевізори, годинники та автомобілі
9. Інтеграція з Git, GitHub та Bitbucket
10. Підтримка Kotlin, Java, NDK та C++
11. Функціональність може бути розширена за допомогою плагінів - наприклад, підтримка Javascript
12. Інтеграція з Firebase та Google Cloud Platform
13. Візуальний редактор для XML та Jetpack Compose макетів
14. Розширені інструменти налагодження для кожного рівня додатку
15. Vector Asset Studio для створення, редагування та керування векторною графікою та зображеннями
16. Редактор перекладів для редагування файлів локалізацій
17. Аналізатор APK-файлів
18. Інтеграція з Android Debug Bridge (ADB)
19. Інтерфейс для перегляду веб-сторінок
20. Керування підключеними пристроями
21. Apply Changes дозволяє розробникам вносити зміни у запущений застосунок без перезапуску ітр, а в деяких випадках навіть без перезапуску поточної активності.

Android Studio прискорює створення програмного продукту за рахунок вбудованих шаблонів, автоматичної перевірки написаного коду до моменту його компіляції на орфографічні та синтаксичні помилки. При створенні готового застосунку середовище автоматично збудує його з усіма необхідними залежностями та налаштуваннями, що зменшує імовірність помилок та прискорює розробку застосунків.

3.3 Обґрунтування використання бази даних

Щоб гарантувати збереження даних, згенерованих під час роботи програми, було вирішено використовувати базу даних SQLite через бібліотеку Room.

SQLite - це вбудована, безсерверна система керування реляційними базами даних. Це бібліотека з відкритим вихідним кодом з нульовою конфігурацією в пам'яті, яка не потребує встановлення. Крім того, вона дуже зручна, оскільки має розмір менше 500 кБ, що значно менше, ніж інші системи управління базами даних.

Особливості вибору SQLite:

- це програмне забезпечення з відкритим вихідним кодом. Після встановлення програмне забезпечення не потребує жодної ліцензії.
- є безсерверним, оскільки для роботи йому не потрібен інший серверний процес або система.
- дозволяє працювати з декількома базами даних в одному сеансі одночасно, що робить його гнучким.
- це кросплатформенна СУБД, яка може працювати на всіх платформах, включаючи macOS, Windows тощо.
- не потребує конфігурації. Вона не потребує налаштування або адміністрування.

Room — це компонент архітектури Android, представлений на Google I/O у 2017 році як заміна Realm, ORMLite та GreenDao.

Ця бібліотека спрощує роботу з об'єктами SQLiteDatabase у програмах, що розробляються, значно зменшуючи кількість шаблонного коду та виконуючи перевірку SQL-запитів під час компіляції [16].

Room — це високорівневий інтерфейс для низькорівневих функцій SQLite, вбудованих в Android.

Більшість його функціональних можливостей виконується під час компіляції, що дозволяє створювати API на основі вбудованого API SQLite, що значно спрощує взаємодію з базами даних Android.

На відміну від багатьох ORM, Room використовує обробники операторів для зберігання даних. Це означає, що на відміну від більшості інших ORM, включаючи Realm, класи застосунків і моделей не потребують розширень Room.

База даних Room побудована на трьох головних компонентах, таких як:

- database;
- dao;
- entity.

Сутність (entity) — представляє собою таблицю в базі даних. Room автоматично створює таблицю для кожного класу, який має анотацію "сутність", де поля в класі відповідають стовпцям у таблиці. Таким чином, класи "сутність" зазвичай є невеликими модельними класами, які не містять логіки.

База даних (database) — для створення бази даних використовуючи Room, необхідно створити клас та анотувати його з базою даних. Для створення бази даних, необхідно вказувати, які таблиці повинні бути в цій базі даних і версії баз даних в анотації.

Дао (dao) — щоб мати повноцінний доступ до бази даних, необхідно створювати об'єкт доступу до даних використовуючи інтерфейс з анотацією Dao. Всередині інтерфейсу створюються методи, які необхідні для операції з базою даних.

Структура побудови бази даних представлена на рисунку 3.1 [17].

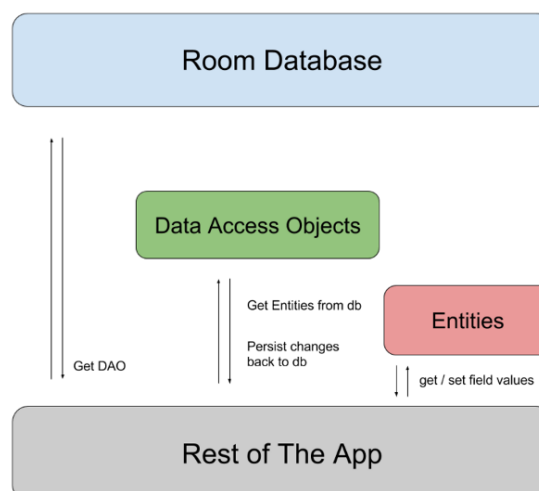


Рисунок 3.1 – Структура бази даних Room

База даних Room є зручним інструментом, який прискорює та спрощує створення та взаємодію з базою даних в мобільних застосунках, як під час розробки, так і під час користування.

Важливою особливістю обраної бази даних є необхідність перезавантаження програмного забезпечення, яке побудоване з використання даної технології, при зміні структури бази даних під час розробки, для коректної роботи мобільного застосунку.

3.4 Обґрунтування використання сторонніх бібліотек

Під час розробки мобільного застосунку були використанні бібліотеки, які наведені нижче.

1. Material.
2. Navigation.
3. Play-services-location.
4. Picasso.

Material – це адаптивна система рекомендацій, компонентів та інструментів, які підтримують найкращі практики в дизайні користувацьких інтерфейсів. Заснований на відкритому вихідному коді, Material спрощує реалізацію дизайну під час розробки програмного продукту.

Material допомагає прискорити розробку інтерфейсу програмного продукту, за рахунок готових модулів. Для користування нею достатньо під'єднати необхідні залежності та обрати необхідні компоненти зі списку стандартних чи розширених прикладів [18].

При використанні бібліотеки Material час розробки інтерфейсу суттєво скорочується, а якість готового продукту наближається до провідних застосунків Play Маркету.

Navigation – це фреймворк для навігації між «пунктами призначення» в Android-застосунку, що забезпечує узгоджений API, незалежно від того, чи реалізовані ці пункти призначення, як фрагменти, дії або інші компоненти [19].

Navigation – це взаємодія, яка дозволяє користувачам переходити, входити та виходити з різного вмісту програми.

Навігаційний компонент Android Jetpack допомагає реалізувати навігацію, від простих натискань до більш складних конструкцій, таких як панелі застосунків і навігаційні панелі.

Компонент навігації також забезпечує узгоджену та передбачувану взаємодію з користувачем, дотримуючись встановленого набору вказівок щодо «призначення» в програмах Android, забезпечуючи послідовний API незалежно від місця призначення, розгорнутого як фрагмент чи інший компонент.

Play-services-location – це фреймворк розроблений компанією Google, є складовою частиною її навігації у застосунку «Карти», та вважається еталонною бібліотекою для роботи з координатами та швидкістю руху у мобільних застосунках різної направленості. Дана бібліотека допомагає уникнути дублювання функції по визначенню усіх навігаційних даних [20].

Даний фреймворк необхідний для реалізації логіки поїздок, оскільки саме ця бібліотека допомагає точно визначати місце знаходження пристрою, швидкість їзди та прискорення.

Picasso — потужна бібліотека для завантаження та кешування зображень для Android. Зображення додають необхідний контекст і візуальну привабливість програм Android. Picasso спрощує завантаження зображень у програму – часто за допомогою лише одного рядка коду, що значно спрощує розробку [21].

Picasso автоматично виправляє багато типових проблем із завантаженням зображень на Android, які описані далі.

1. Перетворення ImageView та обробка скасування завантаження в адаптері.

2. Комплексне перетворення зображень з мінімальним використанням пам'яті.

3. Автоматичне кешування пам'яті та диска.

Оскільки Picasso це бібліотека з відкритим вихідним кодом, вона часто оновлюється, що дає можливість швидко вирішувати проблеми, що виникають, навіть під час створення програмного продукту.

Висновки до розділу 3

Провівши аналіз вимог, які поставлені до програмного продукту, що розробляється, та шляхів реалізації, було вирішено використовувати для вирішення поставлених завдань мову програмування Kotlin, оскільки вона дозволяє розробляти якісно, швидко та використовує найкращі рішення для створення мобільних застосунків, а також є основною мовою для розробки мобільних застосунків за рекомендаціями від Google.

В проєкті використані готові програмні модулі з відкритим вихідним кодом, для створення дизайну, та функцій отримання швидкості від сенсорів телефону. Розробка програмного продукту проводилась в середовищі Android Studio, для прискорення розробки, а також мінімізації помилок компілювання чи залежностей різних бібліотек.

Було вирішено обрати базу даних Room, як швидке та надійне рішення для реалізації збереження необхідних даних під час роботи програмного забезпечення.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ЗАСТОСУНКУ

4.1 Діаграма прецедентів

Розроблена діаграма прецедентів є важливим інструментом для аналізу, проектування та комунікації функціональності системи та описує можливі взаємодії користувача з програмою (рис. 4.1).

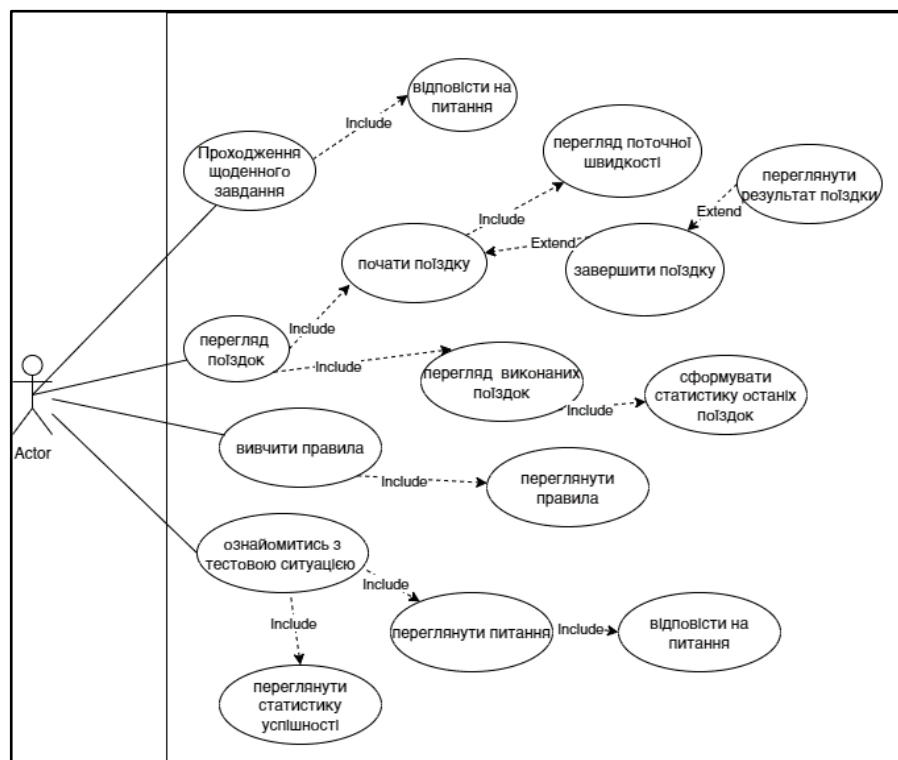


Рисунок 4.1 – Діаграма прецедентів системи

Розроблена діаграма описує шляхи взаємодії з програмним продуктом. Користувач має можливість пройти щоденне завдання для перевірки своїх теоретичних знань, а також покращення своєї статистики користувача.

При перегляді поїздок користувач має можливість створити нову поїздку та зібрати за її результатами певну інформацію, для подальшого створення статистики, як поїздки окремо, так поїздок загалом, також він має можливість переглянути статистику за минулими поїздками й кожну окремо, й загально сформувати дані.

Під час вивчення правил користувач отримує доступ до усіх правил дорожнього руху у зручному для вивчення форматі, з інтерактивними підказками та офіційними поясненнями.

У момент ознайомлення з тестовою ситуацією користувач має можливість перевірити свої теоретичні знання та вдосконалити їх.

4.2 Функціональна декомпозиція

Під час розробки програмного продукту було зроблено функціональну декомпозицію проєкту для полегшення процесу розробки програмного продукту, а також з метою покращеної оцінки затрачених ресурсів і для покращення реалізації продукту, що розробляється (рис. 4.2).

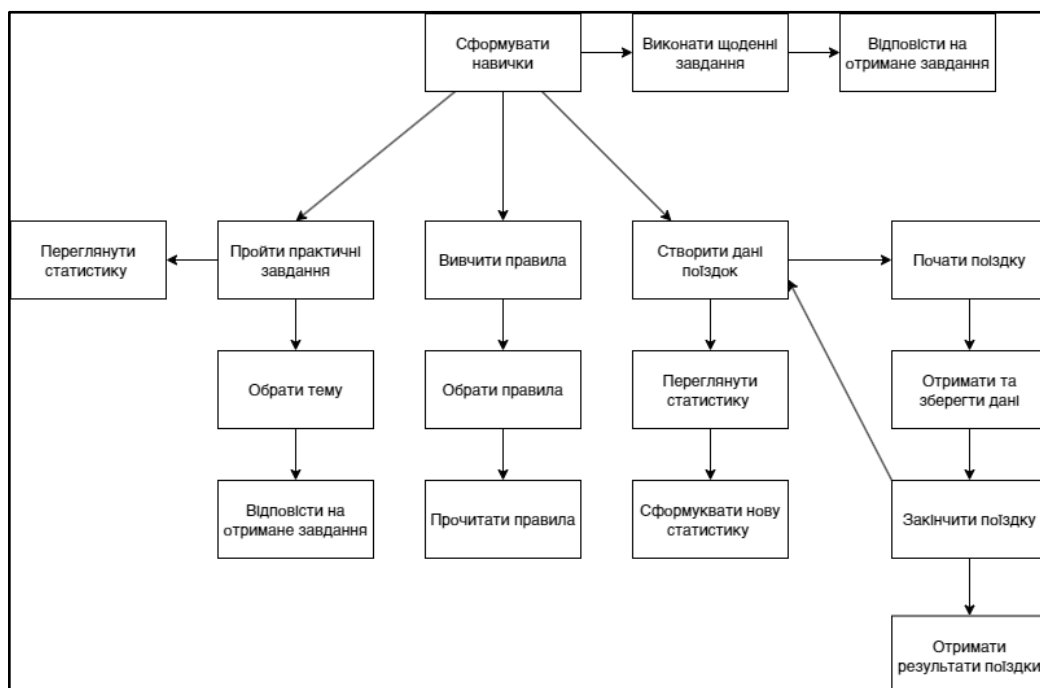


Рисунок 4.2 – Діаграма функціональної декомпозиції

Дана декомпозиція показує взаємодію різних функцій в програмі, при цьому головною функцією програми виступає формування навичок водіння.

4.3 Алгоритм формування якості поведінки водія

Застосунок розроблений з метою покращення теоретичних та практичних навичок водіння повинен чітко відображати прогрес користувача у його знаннях та вміннях. З метою реалізації цього задуму було створено алгоритм, який змінює рейтинг якості водіння на основі отриманих даних поїздок та вивчених правил дорожнього руху.

Рейтинг водіння складається з двох показників, які визначаються окремо, але не рівнозначно впливають на зміну даного показника, оскільки саме під час реального водіння транспортного засобу людина застосовує на практиці набуті знання. Під час водіння неможливо визначити, порушення відбулося через незнання правил дорожнього руху, чи навмисного нехтування певними правилами.

Таким чином, за результатами поїздки рейтинг збільшується максимум на чотири показники при дотриманні всіх умов безпечного пересування, чи зменшується на шість показників при нехтуванні усіх правил дорожнього руху, що відстежуються. За результатами проходження тестових завдань на знання ПДР рейтинг може змінюватись на два пункти максимально.

Зміна рейтингу за результатами поїздки

Показники які впливають на рейтинг кожної окремої поїздки:

- 1) середня медіальна швидкість;
- 2) кількість перевищень швидкості;
- 3) кількість екстремальних гальмувань;
- 4) кількість різких прискорень.

За створеним алгоритмом зміни рейтингу під час поїздки, рейтинг змінюється тільки при виконанні певних умов цієї поїздки, а саме: тривалість поїздки повинна бути не менша ніж 10 хвилин, а відстань повинна бути більшою за

3 кілометри, для запобігання зловживанням поїздок для штучного покращення свого рейтингу. Розроблений алгоритм представлений на рисунку 4.3.

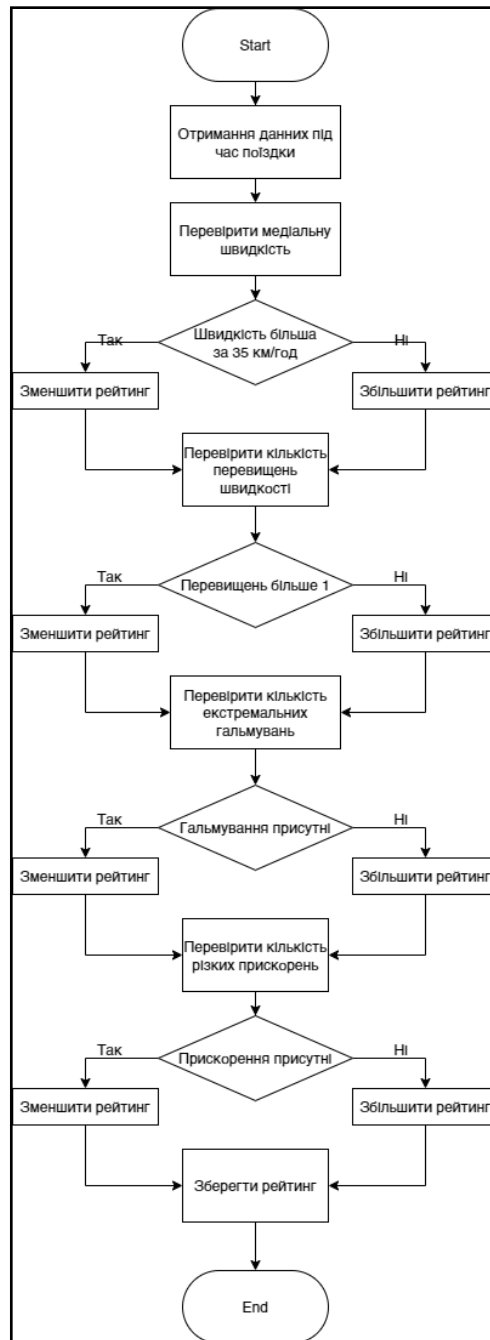


Рисунок 4.3 – Алгоритм зміни рейтингу за результатами поїздки

Для міського пересування середня медіальна швидкість поїздки складає від 20 до 30 км на годину в залежності від завантаженості міста та кількості учасників дорожнього руху. Таким чином було вирішено взяти за відмітку зміни рейтингу швидкість, яка дорівнює 35 км/год.

Якщо швидкість за результатами поїздки більша за 35 – рейтинг понижується на 1 бал, а якщо нижча то збільшується на таку ж кількість балів.

При кількості перевищень швидкості від 0 до 1, рейтинг підвищується на 1 бали, при кількості перевищень від 2 до 4, рейтинг знижується на 1 бал, а якщо перевищень більш чи дорівнює 5 – рейтинг знижується на 2 бали відповідно.

Під час руху транспорту будь-які екстрені дії можуть призводити до фатальних наслідків, саме тому при різкому прискоренні чи екстремальному гальмуванні потрібно змінювати рейтинг максимально сильно, як за грубе порушення комплексу правил дорожнього руху.

Якщо під час руху не було різких прискорень чи гальмувань, рейтинг збільшується на 1 бал, при кількості 1 для кожного з показників рейтинг зменшується на 1 бал, а при кількості гальмувань більше за 2 рейтинг понижується на 2 бали відповідно.

Зміна рейтингу за результатами тестування

На зміну рейтингу під час тестування впливає тільки кількість правильних та неправильних відповідей, на рейтинг не впливає час проходження тестування, оскільки даний програмний продукт націлений саме на вивчення правил дорожнього руху, а не на складання тестів за правилами сервісних центрів.

Під час проходження тестування користувач може подивитись поточний час проходження тесту, а в подальшому дізнатись скільки він витратив часу на всі свої тестування.

Рейтинг за тестуванням змінюється тільки при проходженні завдання «20 випадкових питань», інші види тестів, такі як, обрані питання, питання дня та всі тести, на рейтинг не впливають, а слугують лише механізмом кращого вивчення правил дорожнього руху.

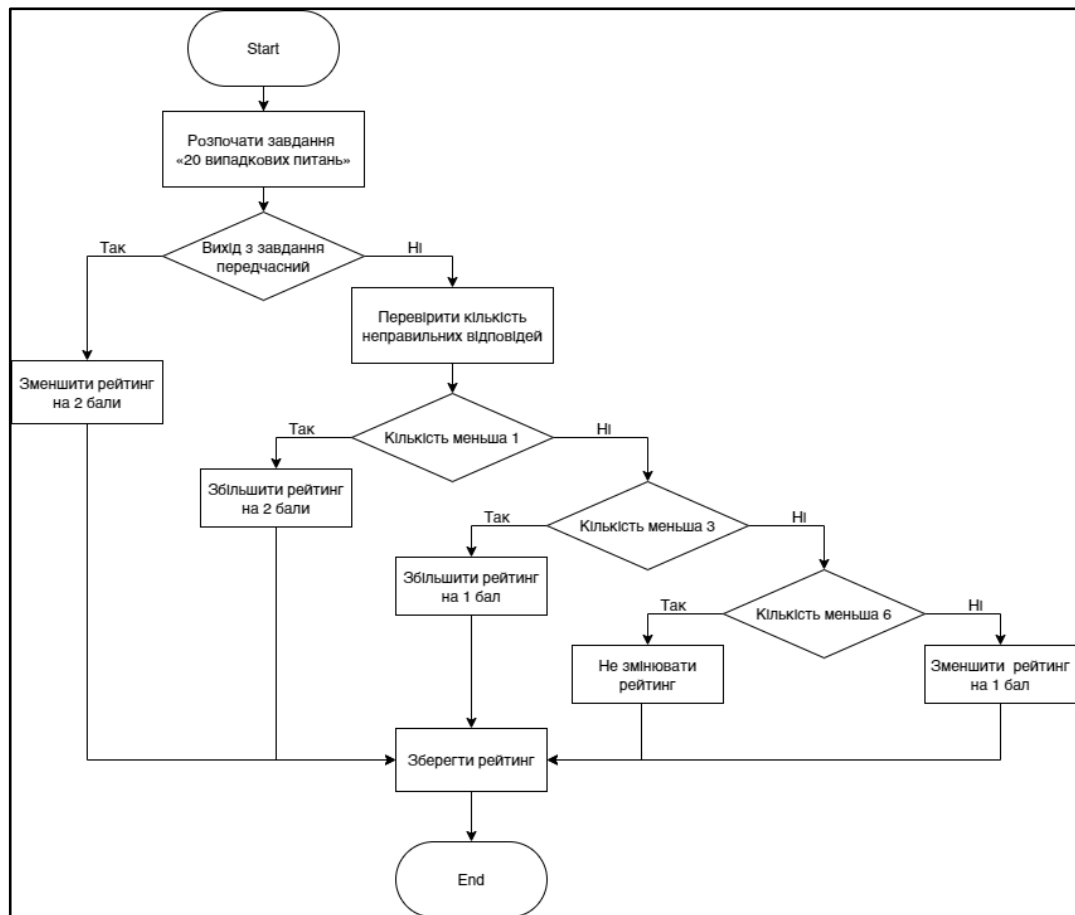


Рисунок 4.4 – Алгоритм зміни рейтингу за пройденим тестуванням

При несвоєчасному завершенню завдання знімається 2 бали, при повному проходженню тестування з кількістю помилок понад 5 рейтинг зменшується на 1 бал, якщо допустити від 3 до 5 помилок, рейтинг не зміниться, а при успішному виконанні та припущені до 2 помилок – підвищиться на 1 бал. Якщо ж пройти тестування без жодної помилки, рейтинг підвищується на максимальних 2 бали.

4.4 Структура бази даних

Збереження та робота з даними є одним з головних завдань, які необхідно реалізувати при створенні будь-якого мобільного застосунку. Збереження та відтворення цілісних даних – це запорука успішного програмного продукту.

Програмний продукт функціонально поділений на дві рівнозначні частини: проходження тестів правил дорожнього руху та визначення якості водіння під час поїздки транспортним засобом. Саме через структуру програмного продукту база даних структурно поділена на дві частини, кожна з яких відповідає за збереження та відтворення даних, що відносяться до неї. Структура бази даних представлена на рисунку 4.5.

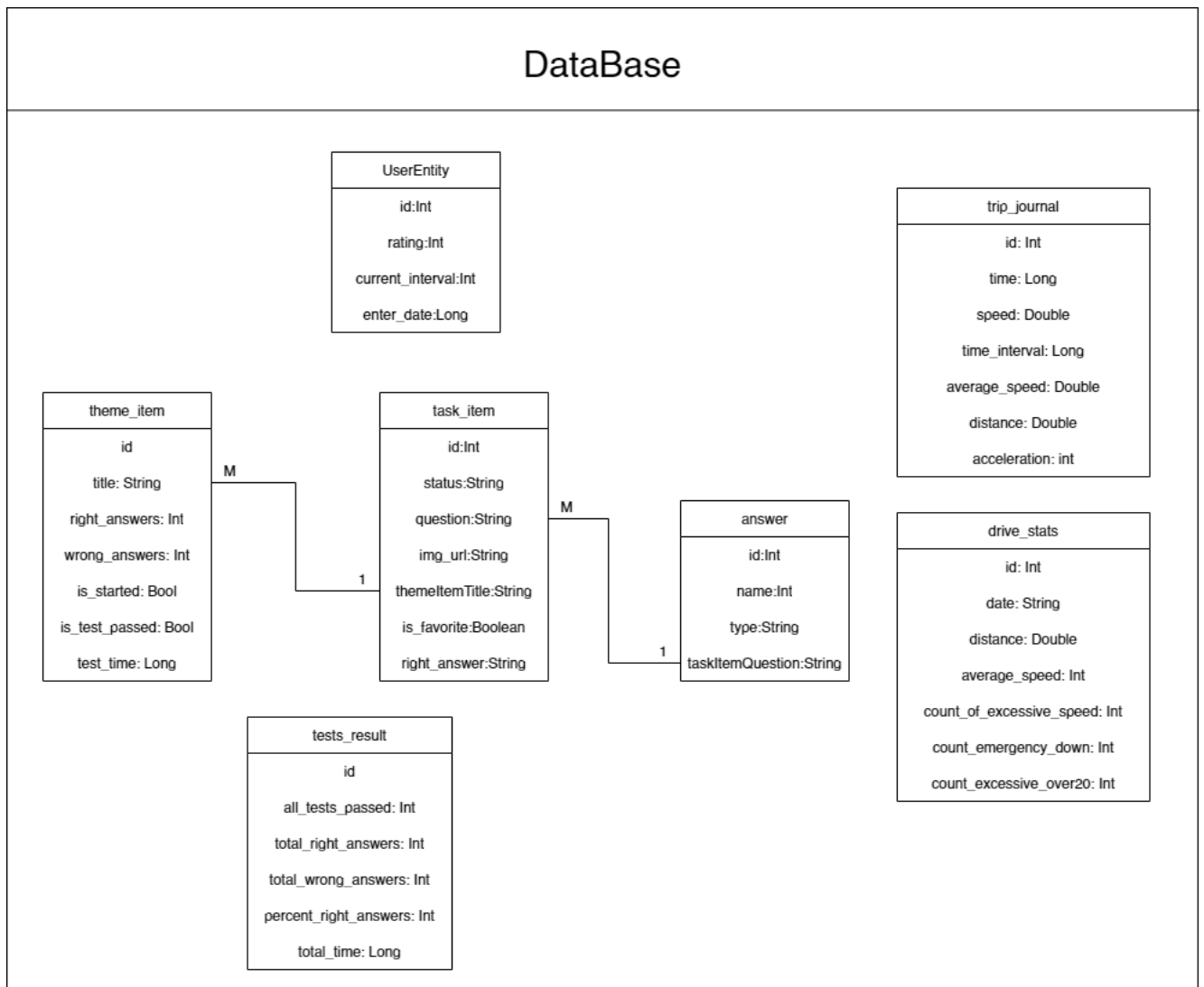


Рисунок 4.5 – Структура бази даних

База даних містить 7 таблиць: UserEntity, theme_item, task_item, answer, tests_result, а також trip_journal та drive_stats. Як зображено на рисунку, певні таблиці мають відношення на рівні бази даних, а деякі пов'язані між собою лише бізнес логікою програми.

Навколо основної логіки мобільного застосунку створена надбудова стимуляції користувачів до щоденного використання мобільного застосунку. Дана надбудова має свою таблицю в базі даних і представлена на рисунку 4.6.

UserEntity
id: Int
rating: Int
current_interval: Int
enter_date: Long

Рисунок 4.6 – Таблиця сутності користувача

Таблиця сутності користувача відповідає за збереження та відтворення персонального рейтингу користувача, а також його неперервний відрізок користування мобільним застосунком.

Структурно складається з наступних полів:

- id: Int;
- rating: Int;
- current_interval: Int
- enter_date: Long.

Поле id відповідає за унікальний номер користувача, і є приватним ключем для даної таблиці

Поле rating зберігає дані рейтингу користувача, який він отримав під час поїздки чи проходження тестів.

Поле current_interval зберігає дані неперервного користування програмним забезпеченням, дані обнуляються, якщо користувач не використовував мобільний застосунок протягом 24 годин.

Поле `enter_interval` зберігає дату та час останнього користування.

Таблиці теоретичного тестування

Оскільки логіка роботи програмного забезпечення передбачає проходження теоретичного тестування на знання правил дорожнього руху, необхідно зберігати та відтворювати велику кількість питань, які розподілені по різних темах та мають декілька варіантів відповідей. Структурно на верхньому рівні бази даних знаходиться розподіл питань за темами, таблиця розподілу питань за темами представлена на рисунку 4.7

theme_item
id
title: String
right_answers: Int
wrong_answers: Int
is_started: Bool
is_test_passed: Bool
test_time: Long

Рисунок 4.7 – Таблиця тематичного розподілу

Таблиця тематичного розподілу відповідає за збереження та відтворення інформації про якісну на кількісну характеристику пройдених тестів.

Структурно складається з наступних полів:

- `id: Int;`
- `title: String`
- `right_answers: Int`
- `wrong_answers: Int`

- is_started: Bool
- is_test_passed: Bool
- test_time: Long

Поле id відповідає за унікальний номер теми і є приватним ключем для даної таблиці.

Поле title зберігає назву теми.

Поле right_answers відповідає за збереження кількості правильних відповідей кожної теми.

Поле wrong_answers відповідає за збереження кількості неправильних відповідей кожної теми.

Поле is_started відповідає чи початий тест, це потрібно, щоб знати, чи тест з першого питання, якщо тест ще не був відкритий, чи з останнього, де зупинились, якщо тест був початий.

Поле is_test_passed зберігає інформацію про те, чи пройдено тест, це необхідно для лічильника пройдених тестів, щоб відрізнити від тестів, які були ще не відкриті.

Поле test_time зберігає кількість часу витраченого на кожну тему.

В кожній темі є перелік питань, які до неї відносяться. Кожне питання має свою структуру, яка представлена на рисунку 4.8.

task_item
id: Int
status: String
question: String
img_url: String
themeItemTitle: String
is_favorite: Boolean
right_answer: String

Рисунок 4.8 – Таблиця структури питання

Таблиця структури питання відповідає за збереження та відтворення кожного питання, яке необхідне для складання теоретичної частини іспиту в сервісному центрі.

Структурно складається з наступних полів:

- id: Int;
- status: String;
- question: String
- img_url: String
- themeItemTitle: String
- is_favorite: Boolean
- right_answer: String

Поле id відповідає за унікальний номер питання і є приватним ключем для даної таблиці.

Поле status зберігає статус кожного питання, питання може мати такі статуси: FAIL, RIGHT, SELECTED, CLOSE.

Поле question зберігає текстовий зміст самого питання, на яке необхідно дати відповідь.

Поле img_url зберігає посилання на зображення до кожного питання, може бути пустим, якщо питання не передбачає ілюстрації.

Поле themeItemTitle зберігає відомості про тему, до якої відноситься дане питання.

Поле is_favorite зберігає стан питання, чи є воно обраним, для окремого відтворення обраних питань.

Поле right_answer зберігає правильну відповідь на кожне питання.

До кожного питання є список з варіантами відповідей на нього. Кожна відповідь зберігається в таблиці відповідь, яка продемонстрована на рисунку 4.9.

answer
id: Int
name: Int
type: String
taskItemQuestion: String

Рисунок 4.9 – Таблиця відповіді

Таблиця відповіді спрямована на збереження самої відповіді та інформації, до якого питання вона відноситься.

Структурно складається з наступних полів:

- id: Int;
- name: Int
- type: String
- taskItemQuestion: String

Поле id відповідає за унікальний номер відповіді і є приватним ключем для даної таблиці.

Поле name зберігає відповідь на питання.

Поле type зберігає стан відповідей, який необхідний для відображення правильних та неправильних відповідей після проходження тесту. Відповідь може набувати таких станів: RIGHT, WRONG, DEFAULT. За замовчуванням всі відповіді мають стан default, якщо відповідь обрана правильно вона набуває статусу right і тим самим змінює відображення обраного варіанту. Якщо було обрано неправильну відповідь, вона набуває статусу wrong, а правильна відповідь статусу right.

Поле taskItemQuestion зберігає відношення відповіді до певного питання.

Таблиці теми, питання та відповіді мають певні відношення між собою, які продемонстровані на рисунку 4.10.

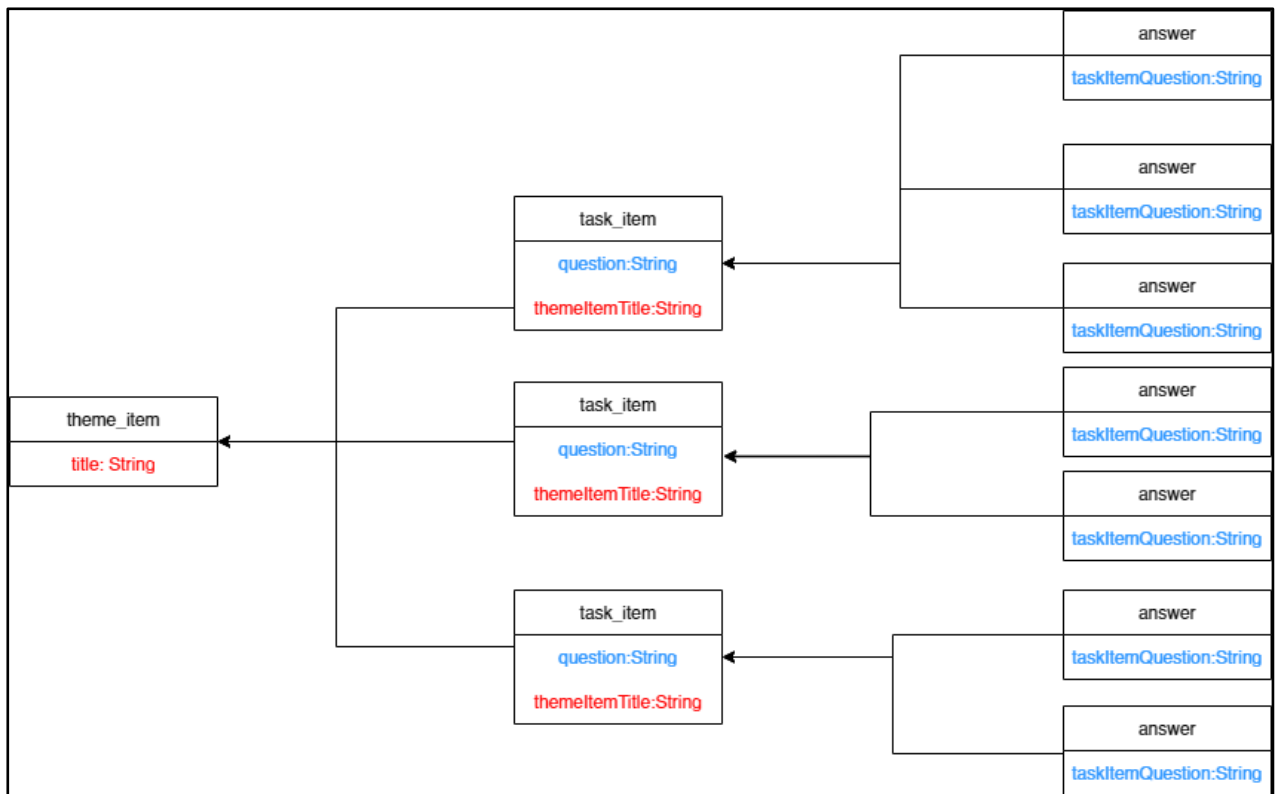


Рисунок 4.10 – Реляційні відносини таблиць

Таблиця тематичного розподілу відноситься до таблиці питань відношенням один до багатьох, оскільки в кожній темі може бути багато відповідей. У таблиці питань є поле `themeItemTitle`, яке є ключем до поля `title` в таблиці теми, для позначення, до якої теми відноситься завдання.

Таблиця структури питання відноситься до таблиці відповіді відношенням один до багатьох, оскільки кожного питання є як мінімум дві відповіді. У таблиці відповіді параметр `taskItemQuestion` є ключем до поля `question` в таблиці питань, щоб позначити, до якого питання відноситься відповідь.

Під час проходження теоретичного тестування збирається та зберігається загальна статистика успішності пройдених тестів (рис 4.11).

tests_result
id: Int
all_tests_passed: Int
total_right_answers: Int
total_wrong_answers: Int
percent_right_answers: Int
total_time: Long

Рисунок 4.11 – Таблиця статистики тестування

Таблиця статистики зберігає всю необхідну інформацію для представлення повної статистики за результатами пройдених тестів.

Структурно складається з наступних полів:

- id: Int;
- all_tests_passed: Int
- total_right_answers: Int
- total_wrong_answers: Int
- percent_right_answers: Int
- total_time: Long

Поле id відповідає за унікальний номер запису і є приватним ключем.

Поле all_tests_passed зберігає кількість пройдених тестів, незважаючи на правильність чи неправильність відповіді.

Поле total_right_answers зберігає кількість правильних відповідей, які були обрані під час проходження тестів.

Поле total_wrong_answers зберігає кількість неправильних відповідей, які були обрані під час проходження тестів.

Поле percent_right_answers зберігає відношення кількості правильних відповідей до загальної кількості пройдених тестів.

Поле `total_time` зберігає загальний час, який був витрачений на проходження всіх тестів за весь час користування програмою.

Таблиці практичних поїздок

Для повноцінної роботи програмного забезпечення в базі даних також створено таблицю, яка забезпечує збереження та відтворення даних, які збираються безпосередньо під час поїздки. Журнал поїздки представлений на рисунку 4.12

trip_journal
id: Int
time: Long
speed: Double
time_interval: Long
average_speed: Double
distance: Double
acceleration: int

Рисунок 4.12 – Таблиця журнал поїздки

Таблиця зберігає дані, які надходять від GPS системи безпосередньо під час поїздки.

Структурно складається з наступних полів:

- `id: Int;`
- `time: Long`
- `speed: Double`
- `time_interval: Long`
- `average_speed: Double`
- `distance: Double`
- `acceleration: int`

Поле `id` відповідає за унікальний номер запису і є приватним ключем.

Поле time зберігає час отримання даних від GPS.

Поле speed зберігає миттєву швидкість транспортного засобу.

Поле time_interval зберігає інтервал часу між отриманням даних.

Поле average_speed зберігає середню швидкість за інтервал часу.

Поле acceleration зберігає прискорення транспортного засобу.

Після завершення кожної поїздки дані, збережені в таблиці журнал поїздки, опрацьовуються та записуються до таблиці статистика поїздки, яка продемонстрована на рисунку 4.13

drive_stats
id: Int
date: String
distance: Double
average_speed: Int
count_of_excessive_speed: Int
count_emergency_down: Int
count_excessive_over20: Int

Рисунок 4.13 – Таблиця статистики поїздок

Дана таблиця відповідає за збереження і відображення даних, які були отримані під час поїздки,

Структурно складається з наступних полів:

- id: Int;
- date: String
- distance: Double
- average_speed: Int
- count_of_excessive_speed: Int
- count_emergency_down: Int

— `count_excessive_over20`: Int

Поле `id` відповідає за унікальний номер поїздки і є приватним ключем для даної таблиці.

Поле `date` зберігає дату та точний час, коли відбулась поїздка.

Поле `distance` відповідає за збереження на надання загальної відстані, яка була пройдена під час поїздки.

Поле `average_speed` зберігає середню швидкість, яка була вирахована протягом всієї поїздки.

Поле `count_of_excessive_speed` зберігає кількість екстрених гальмувань.

Поле `count_emergency_down` зберігає кількість різких прискорень.

Поле `count_excessive_over20` зберігає кількість порушень правил дорожнього руху, а саме кількість перевищень швидкості на понад 20 кілометрів.

Висновки до розділу 4

Даний розділ докладно описує та обґрунтовує створення діаграм, їх необхідність для подальшої реалізації проєкту. Також було докладно описано алгоритм формування рейтингу водія за результатами його теоретичної підготовки, та фактичною якістю їзди. Представлений опис розробленої бази даних, відношення таблиць між собою, та описано кожну таблицю, яка задовольняє повне функціонування програмного забезпечення.

5 ВИКОРИСТАННЯ ЗАСТОСУНКУ КОРИСТУВАЧЕМ

5.1 Системні вимоги застосунку

Для коректної роботи створеного програмного продукту, необхідно використовувати його на пристрої з характеристиками, які описані далі.

1. Android – як операційна система, версії 8.0 Oreo та вище.
2. GPS модуль на пристрої.
3. Доступ до мережі інтернет.
4. Вільний обсяг пам'яті не менше 30 МБ.

Під час використання програми необхідно слідувати підказкам, які надаються програмою, для коректної її роботи.

5.2 Встановлення мобільного застосунку

Перед безпосереднім використанням розробленого програмного продукту, його необхідно завантажити та встановити на пристрій, на якому він буде використовуватись. Алгоритм дій:

- 1) завантажити apk файл програми на пристрій;
- 2) натиснути на завантажений файл;
- 3) дотриматись інструкції встановлення;
- 4) надати необхідні дозволи у налаштуваннях;
- 5) увімкнути місцеперебування та передачу даних.

5.3 Запуск і користування мобільним застосунком

При першому запуску програмного продукту, якщо були надані усі дозволи для коректної роботи мобільного застосунку, відобразиться перший екран «Правила дорожнього руху», який представлений на рисунку 5.1.

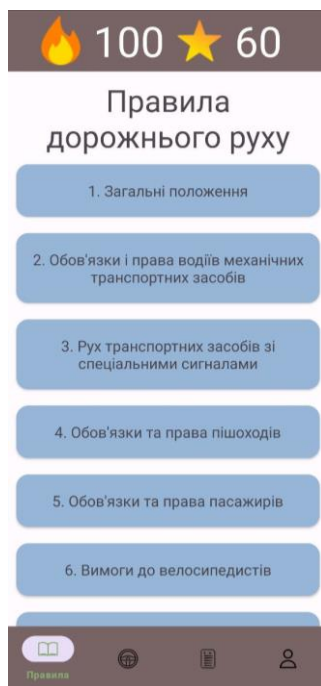


Рисунок 5.1 – Правила дорожнього руху

На кожному екрані знизу є панель доступу до усіх основних розділів застосунку, таких як:

- 1) правила;
- 2) водіння;
- 3) тести;
- 4) профіль.

Також на верхній панелі кожного екрану відображається поточний рейтинг користувача, при першому запуску продукту, рейтинг автоматично встановлюється на 60, надалі він може понизитись до 0 чи піднятись до відмітки у 100, в залежності від якості водіння та знання правил дорожнього руху користувача.

Екран правил відповідає за надання користувачу повного переліку тем правил дорожнього руху, при натисканні на кнопку з назвою теми відкривається новий екран, в який завантажуються усі правила даної теми. Екран з правилами першої теми продемонстрований на рисунку 5.2.



Рисунок 5.2 – Загальні положення

Даний екран надає повну інформацію про правила дорожнього руху теми Загальні положення, інші теми будуть відображатись аналогічно.

Перейдемо до розділу водіння, який представлений на рисунку 5.3, за допомогою нижньої панелі, для цього необхідно натиснути на відповідну іконку.

Даний екран надає можливість розпочати нову поїздку, або переглянути результати попередніх поїздок натиснувши на кнопку «Статистика».

Під час поїздки на екрані відображаються дані миттєвої швидкості, пройденого шляху та максимальної швидкості, яка була розвинута, що обраховуються в режимі реального часу.



Рисунок 5.3 – Екран водіння

При натисканні кнопки «Старт» спрацьовує логіка поїздки й програма починає отримувати дані про поточну швидкість, та точний час отриманих даних від GPS модуля, ця інформація отримується кожні 2 секунди.

Під час поїздки екран може знаходитись у двох станах: безпосередня поїздка, яка показана на рисунку 5.4, та пауза поїздки, що показана на рисунку 5.5

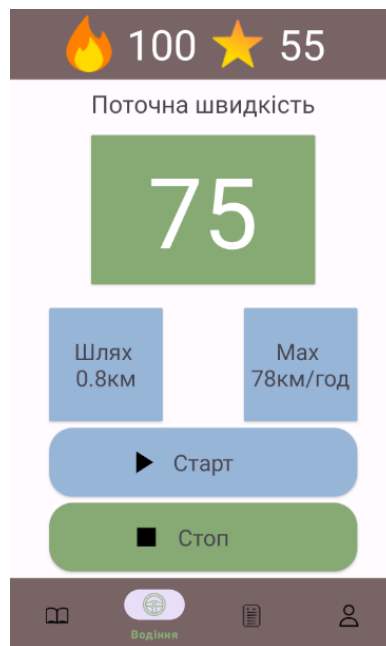


Рисунок 5.4 – Поїздка



Рисунок 5.5 – Пауза

Під час паузи дані від GPS не отримуються, при поновленні поїздки з паузи дані знову отримуються в режимі реального часу.

При натисканні на кнопку «Стоп» – поїздка закінчується і з’являється екран результатів поїздки, який продемонстровано на рисунку 5.6, також завершити поїздку можна з паузи.



Рисунок 5.6 – Статистика поїздки

Даний екран відображає кінцеву статистику однієї поїздки, а саме:

- 1) відстань однієї поїздки;
- 2) середню швидкість руху;
- 3) кількість перевищень швидкості;
- 4) кількість екстрених гальмувань;
- 5) кількість різких прискорень.

При натисканні на кнопку «Далі», відкриється новий екран, що продемонстровано на рисунку 5.7, в якому зберігається статистика за усі минулі поїздки і відображається в порядку від останньої до першої.

Під час кожної поїздки фіксується дата, а також час, коли вона розпочалась і коли завершилась.

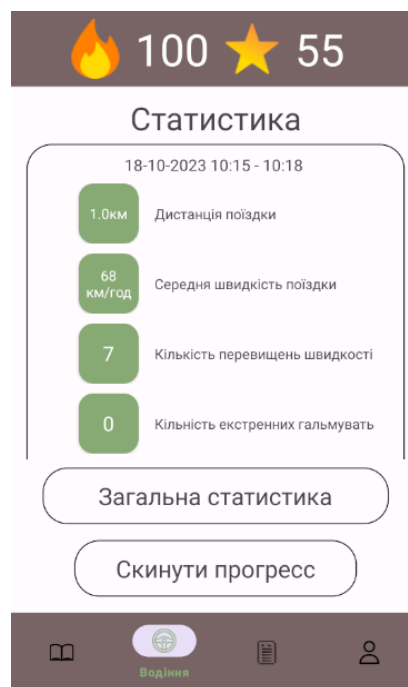


Рисунок 5.7 – Екран статистики поїздок

Також є можливість сформувати загальну статистику по всіх поїздках натиснувши на кнопку «Загальна статистика», вона сформує та відобразить середні значення отримані за результатами останніх поїздок.

Перейшовши до розділу «Тести» на нижній панелі, натиснувши на іконку, відкривається екран з таким функціоналом:

- 1) питання дня;
- 2) всі теми;
- 3) 20 випадкових питань;
- 4) обрані питання;
- 5) статистика.

Кнопка «Питання дня» відкриває 5 випадкових тестових питань, які можна пройти для оновлення своїх знань та повторення ПДР, при проходженні питання дня один раз за день, збільшується кількість неперервних днів навчання.

Кнопка «Всі теми» надає доступ до всіх питань правил дорожнього руху, які розподілені за темами, в даному розділі можна перевірити свої знання конкретної теми чи повторно пройти питання, на які раніше вже була дана відповідь. Даний екран представлено на рисунку 5.8.



Рисунок 5.8 – Тести

Кнопка «20 випадкових питань» надає доступ до проходження повноцінного тестувального білета, емулюючи іспит на водійські права у сервісному центрі. Екран з тестами продемонстровано на рисунку 5.9.



Рисунок 5.9 – Випадкові питання

Після відповіді на питання є можливість повернутись назад та дізнатись правильну відповідь, якщо було допущено помилку, що представлено на рисунку 5.10 та рисунку 5.11.



Рисунок 5.10 – Правильна відповідь



Рисунок 5.11 – Неправильна відповідь

Під час будь-якого проходження тестів можна додати певне питання до обраних, натиснувши в лівому верхньому куті на зірку.

За результатами тестування можна подивитись статистику, натиснувши на кнопку «Статистика», що показано на рисунку 5.12.

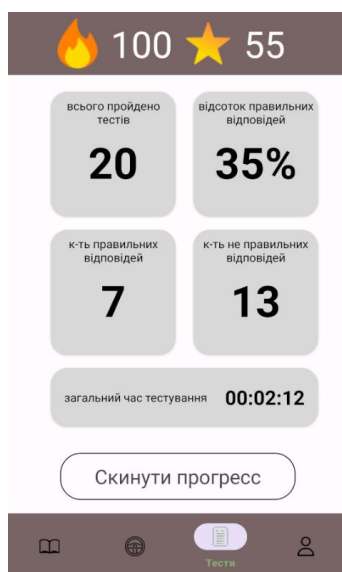


Рисунок 5.12 – Статистика тестувань

Дана статистика показує кількість пройдених тестів, кількість правильних та неправильних відповідей, а також автоматично вираховує відсоток правильних відповідей. При необхідності дані статистики завжди можна видалити, натиснувши кнопку «Скинути прогрес».

Перейдемо до розділу «Профіль», натиснувши на відповідну іконку на нижній панелі. Даний розділ, який представлений на рисунку 5.13, показує рейтинг користувача, та за допомогою кольорової підсвітки показує, в якій групі знаходиться його рейтинг, червоній, жовтій чи зеленій.



Рисунок 5.13 – Профіль

При першому запуску програмного продукту, поточний відрізок користування буде складати 0, а поточний рейтинг 60. Це стартові дані, які закладені в систему.

На верхній панелі кожного екрану дублюється поточний відрізок і рейтинг користувача, для постійної мотивації, покращувати свої знання та навички, а також не забувати користуватись програмним забезпеченням щодня для формування кращої поведінки.

Профіль надає можливість не тільки переглянути свій рейтинг, а і подивитись статистику поїздок та статистику тестування, винагороди та часті запитання.

5.4 Система мотивації користувачів

Робота над програмним забезпеченням, показала необхідність зацікавлення користувачів до постійного використання його. Це є важливим фактором формування якісної поведінки автомобілістів, оскільки саме постійне повторення правил дорожнього руху і дотримання їх при реальних поїздках, впливає на кількість дорожньо-транспортних пригод.

За результатами дослідження різних систем мотивації, було прийнято рішення, що система повинна бути гейміфікована, для покращення експлуатації і залучення користувачів до постійного використання системи.

Однією з реалізованих систем мотивування користувачів став, нескінченний відрізок користування, він повинен мотивувати користувачів щоденно використовувати систему тим самим повторювати та покращувати правила та навички водіння. Система нескінченного відрізка користування продемонстровано на рисунку 5.14



Рисунок 5.14 – Система нескінченного відрізка користування

Дана система працює на основі лічильника з збільшення відрізки при щоденному використанні на 1. Для збільшення відрізка користувачу необхідно хоча б один раз на день пройти тестове завдання, це може бути «Питання дня» або «20 випадкових питань», якість проходження не впливає на відрізок, але впливає на статистику проходження тестів. Іншим шляхом збільшення відрізка є поїздка, якщо людина здійснить хоча б одну поїздку, яка буде записана до статистики, відрізок збільшиться. Однак якщо, користувач не здійснює жодних в вище

перерахованих дій протягом дня, відрізок автоматично понижується до позначки 0, а отже, новий відрізок потрібно набирати з самого початку.

Окрім системи нескінченного відрізка було прийнято рішення ввести систему індивідуального рейтингу користувача. Алгоритм формування рейтингу за результатами тестувань та поїздок було описано в пункті 4.3.

Для додаткової мотивації користувачів покращувати свої знання та навички, рейтинг користувача та нескінченний відрізок демонструється на кожному екрані програмного забезпечення (рис. 5.15).



Рисунок 5.15 – Рейтинг користувача та нескінченний відрізок

Рейтинг користувача позначається зірочкою біля відповідного показника рейтингу. Показник рейтингу також відображається в окремому кольоровому квадраті, що змінює колір відповідно до рейтингу.

Висновки до розділу 5

У цьому розділі описуються можливі сценарії розробленої системи, а також надаються необхідні пояснення щодо роботи програми. Тут також описані інструкції з використання програми та надані знімки екранів готового продукту. Продемонстровано роботу з кожним розділом розробленого мобільного застосунку. Описано розроблену систему мотивувань користувачів до щоденної взаємодії з програмним забезпеченням.

6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

6.1 Опис ідеї проєкту

Ідеєю проєкту є розробка та реалізація комплексного програмного рішення для формування поведінки на дорозі та підвищення рівня обізнаності водіїв, а також зменшення кількості аварій та дорожньо-транспортних пригод за рахунок кращих знань правил та надбаних звичок безпечного водіння.

Проаналізовано зміст ідеї проєкту, сфери його застосування та основні переваги, які користувачі можуть отримати, використовуючи цей продукт. Результати наведені в таблиці 6.1.

Таблиця 6.1 — Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення програмного забезпечення для формування поведінки на дорозі, підвищення рівня обізнаності водіїв, та зменшення кількості аварій та дорожньо-транспортних пригод	1. Використання системи водіями транспортних засобів	Повторення вивчених правил дорожнього руху, покращення стилю водіння.
	2. Водії початківці та люди, що хочуть знати ПДР	Вивчення правил дорожнього руху з системою мотивації
	3. Пасажири транспортних засобів	Відслідковування швидкості та якості поїздки

У роботі були проаналізовані аналогічні рішення, які представлені на даний момент часу, зроблені висновки, що система має численні переваги для формування поведінки, оскільки спрямована на різні аспекти, які впливають на якість їзди, натомість аналогії переважно сконцентровані на одному певних завдань.

Проект має перспективну цінність, оскільки поєднує кращі напрацювання програмних продуктів різних сфер, що забезпечує надання якісної теоретичної підготовки водіїв разом з практичними навичками керування.

У якості аналогів було розглянуто програмні продукти, які направлені на вивчення правил дорожнього руху та складання іспитів в сервісному центр, а саме «Тести ПДР» від green-way – рішення для мобільних телефонів, та сайт vodiу.ua – універсальне рішення. Оскільки прямого аналога розробленої системи немає, також було зроблено порівняння з програмним забезпеченням для отримання емпіричних даних поїздки, а саме «GPS спідометр: одометр» та «Спідометр: GPS Speed Tracker» – мобільні застосунки. Аналіз систем наведено в таблиці 6.2 та таблиці 6.3

Таблиця 6.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту для вивчення правил дорожнього руху.

№ п/п	Техніко- економічні характери- тики ідеї	(потенційні) товари/концепції конкурентів			Слабкі (W), нейтральні (N) та сильні (S) сторони		
		Мій проект	Тести ПДР	vodiу.ua	W	N	S
1	2	3	4	5	6	7	8
1.	Доступність	Всюди, де є смартфон	Всюди, де є смартфон	Всюди, де є інтернет			+

Продовження таблиці 6.2

1	2	3	4	5	6	7	8
2.	Основні технології	Інтегровані бази даних	Комбіновані бази даних	Хмарні бази даних		+	
3.	Повнота функціоналу	Правила Тести Рейтинг	Тести Рейтинг	Правила Тести			+
4.	Надійність	Висока	Висока	Висока			+
5.	Залежність від інтернету	Часткова	Не залежить	Залежить			+
6.	Кількість проведених тестувань	Дуже мала	Дуже велика	Дуже велика	+		

Таблиця 6.3 — Визначення сильних, слабких та нейтральних характеристик ідеї проєкту для визначення емпіричних даних.

№ п/п	Техніко-економічні характеристики ідеї	товари/концепції конкурентів			Слабкі (W), нейтральні (N) та сильні (S) сторони		
		Мій проєкт	GPS спідометр: одометр	Спідометр: GPS Speed Tracker	W	N	S
1	2	3	4	5	6	7	8
1.	Доступність	Всюди, де є смартфон	Всюди, де є смартфон	Всюди, де є смартфон			+

Продовження таблиці 6.3

1	2	3	4	5	6	7	8
2.	Основні технології	GPS засоби	GPS засоби	GPS засоби			+
3.	Вихідні данні	Швидкість, дистанція, перевищення швидкості екстрені дії	Швидкість, дистанція, шлях на карті	Швидкість, дистанція, шлях на карті			+
4.	Надійність	Висока	Висока	Висока			+
5.	Залежність від інтернету	Не залежить	Залежить	Залежить			+
6.	Кількість проведених тестувань	Дуже мала	Дуже велика	Дуже велика	+		

6.2 Технологічний аудит ідеї проєкту

Був проведений аудит технологій, за допомогою яких можна поєднати кращі практики систем конкурентів в один багатофункціональний застосунок, що реалізує ідею проєкту, було визначено доступність технологій, а також чи потребують вони подальшого доопрацювання для реалізації ідеї. Технологічний аудит представлений в таблиці 6.4.

Таблиця 6.4 — Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Розробка програмного забезпечення для формування поведінки на дорозі, підвищення рівня обізнаності водіїв, та зменшення кількості аварій та дорожньо-транспортних пригод	- використання Kotlin для розробки програмного забезпечення.	Наявна	Доступна
		- використання material та navigation для створення зручного інтерфейсу програмної системи.	Наявна	Доступна
		- використання play-services-location для отримання даних за допомогою GPS системи.	Наявна	Доступна
		- використання бібліотеки Picasso для відображення картинок під час роботи мобільного застосунку.	Наявна	Доступна
Обрані технології реалізації проєкту: мова програмування Kotlin, розробка велась у середовищі Android Studio, бібліотеки з відкритим програмним кодом: material, navigation, play-services-location, picasso.				

6.3 Аналіз ринкових можливостей запуску стартап-проєкту

Аналіз наявних ринкових можливостей для ринкової реалізації проєкту та ринкових загроз, які можуть перешкоджати реалізації проєкту, дозволяє спланувати напрям розвитку продукту з урахуванням стану ринкового середовища

та потреб. Оцінити пропозиції, які надають конкуренти та дізнатись потреби потенційних клієнтів.

Потенційні користувачі та їхня характеристика, а також сформований перелік орієнтовних вимог до пропозиції представлено в таблиці 6.5.

Таблиця 6.5 — Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності в поведінці різних цільових груп	Вимоги споживачів до товару
1.	Універсальність при вивченні та повторенні правил дорожнього руху	Водії початківці та люди, що хочуть знати ПДР	Використання в якості навчальних матеріалів	Наявність усіх теоретичних матеріалів
2.	Комплексність рішення для практичного покращення водіння	Водії транспортних засобів	Використання як система покращення навичок	Наявність системи мотивації

Після проведеного аналізу різних груп клієнтів було проаналізовано поточний ринок. За результатами даних досліджень створено таблицю факторів, які сприяють або перешкоджають впровадженню проєкту на ринок. Фактори загроз та фактори можливостей представлені в таблиці 6.6 та таблиці 6.7 відповідно.

Таблиця 6.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Інертність суспільства	Небажання переходити на новий комплексний продукт	Проведення маркетингової кампанії.
2.	Конкуренція	Поява конкурентів зі схожим функціоналом	Введення нового функціоналу. Піар компанія.
3.	Неточність даних продукту	Залежність системи від якості вхідних даних	Удосконалення системи шляхом оптимізації роботи програмного забезпечення.

Таблиця 6.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Висока якість продукту	Вихід на міжнародний ринок	Розширення підтримки програмного продукту в країнах світу
2.	Технологічний розвиток	Розвиток автомобільних технологій	Впровадження нових параметрів поїздки та створення нового алгоритму формування рейтингу
3.	Велика кількість користувачів	Створення фізичного бізнесу, розробка нових систем	Відкриття автошколи з використанням розробленої системи. Створення нових програмних продуктів для покращення якості водіння

Далі проведено аналіз пропозицій, за яким визначаються загальні риси конкуренції ринку (табл. 6.8).

Таблиця 6.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоби бути конкурентоспроможною)
1	2	3
1. Тип конкуренції: - чиста	На ринку присутні багато різних компаній, але деякі з них мають більшу популярність за інших	Створення продукту високої якості та постійне вдосконалення
2. Галузева ознака: - міжгалузева	Використання продукту для покращення поведінки на дорозі	Покращення функціоналу та популяризація продукту
3. Рівень конкурентної боротьби: - національний	Компанії-конкуренти націлені на український ринок	Розвиток продукту та компанії з подальшим виходом на міжнародний ринок
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між програмним забезпеченням одного виду	Покращення програмного продукту за рахунок випуску новіших версій
5. Характер конкурентних переваг: - нецінова	Функціональні можливості програмних засобів	Збільшення функціональних можливостей

Продовження таблиці 6.8

1	2	3
6. Інтенсивність: - не марочна	Важливим є якість продукту, а не бренд	Покращення продукту, реалізація нових алгоритмів

Після вивчення конкурентного середовища, проведено глибокий аналіз умов конкуренції у галузі за допомогою моделі п'яти сил М. Портера (табл. 6.9).

Таблиця 6.9 — Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	GPS спідометр: одометр», «Спідометр: GPS Speed Tracker», Тести ПДР, vodiу.ua	Відносно високий бар'єр через популярність конкурентів, але додаткові дозволи на використання технологій не потрібні	Залежність від постачальників відсутня, система є незалежною	Споживачі є вагомим фактором, оскільки продукти взаємозамінні аналогічними	Відсутні. Наявні лише конкуренти зі схожими розробками деякого функціоналу
Висновки	Активна конкурентна боротьба вимагає систематичної роботи над вдосконаленням застосунку.	Вхід на ринок є невисоким, що свідчить про те, що конкуренти можуть з'явитися на ринку протягом року	Незалежна від постачальників	Сильний вплив на продукт мають споживачі, отже, подальший розвиток продукту залежить від їхніх вподобань та вимог.	Для утримання позицій на ринку необхідно підтримувати високу якість продукту.

На підставі розгляду конкурентного середовища згідно з моделлю М. Портера, який був проведений у таблиці 6.9, а також з урахуванням попередніх таблиць, визначено та обґрунтовано перелік факторів конкурентоспроможності у таблиці 6.10.

Таблиця 6.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
1.	Потреби споживачів	Необхідність розробки проєкту обумовлена потребами споживачів.
2.	Повнота функціоналу	Програмний продукт надає розширений обсяг користувацьких послуг
3.	Надійність	Програмний продукт працює коректно, непередбачувані випадки відсутні
4.	Не залежить від транспортного засобу	Продукт можна використовувати у всіх видах транспортних засобів.
5.	Ціна продукту	Не завищена, оптимальна на ринку
6.	Технічне обслуговування	Постійне оновлення програмного забезпечення та впровадження нових версій

З врахуванням визначених факторів конкурентоспроможності, які представлені у таблиці 6.10, був здійснений аналіз сильних та слабких сторін стартап-проєкту (табл. 6.11).

Таблиця 6.11 — Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваним проектом						
			-3	-2	-1	0	+1	+2	+3
1.	Потреби споживачів	20				+			
2.	Повнота функціоналу	3		+					
3.	Надійність	10				+			
4.	Не залежить від транспортного засобу	6			+				
5.	Ціна продукту	4		+					
6.	Технічне обслуговування	9				+			

За результатами порівняльного аналізу, можна зробити висновки, що проект, який створюється, має сильні переваги в порівнянні з конкурентами.

Техніко-економічну доцільність реалізації проекту було проаналізовано шляхом створення матриці SWOT-аналізу (матриця, що визначає сильні (Strength) та слабкі (Weak) сторони, ризики (Troubles) та можливості (Opportunities)). SWOT-аналіз проекту наведено в таблиці 6.12.

Таблиця 6.12 — SWOT-аналіз стартап-проекту

Сильні сторони (S): – Повнота функціоналу – Доступність – Незалежність від транспортного засобу – Використання гейміфікації	Слабкі сторони (W): – Обмежений ринковий вплив на початковому етапі через рекламні ініціативи конкурентів. – Низька репутація в період запуску проекту.
Можливості (O): – Вихід на міжнародний ринок – Розширення функціоналу програми – Покращення роботи системи	Загрози (T): – Недостатня оптимізованість продукту – Невелика кількість користувачів

На основі проведеного SWOT-аналізу розроблено альтернативну стратегію ринкової поведінки для введення стартап-проєкту на ринок. Також визначено приблизний оптимальний час для їхньої реалізації, враховуючи потенційні проєкти конкурентів, які можуть бути представлені на ринку.

Альтернативи для введення на ринок визначено і представлено у таблиці 6.13, яка містить інформацію щодо ринкового впровадження стартап-проєкту.

Таблиця 6.13 — Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Організація демонстраційної конференції для міжнародного ринку.	60%	6-9 місяців
2.	Безкоштовне розповсюдження створеного продукту	40%	9-12 місяців
3.	Розповсюдження продукту за підпискою	70%	12-18 місяців

В результаті проведеного аналізу було вирішено передусім впровадити систему на основний ринок збуту, оскільки цей підхід надає високу ймовірність отримання ресурсів за короткий період реалізації.

6.4 Розроблення ринкової стратегії проєкту

Розробка ринкової стратегії починається з визначення стратегії охоплення ринку. Визначено групи потенційних користувачів розробленої системи для діяльності вибраних сегментів ринку, які показано в таблиці 6.14.

Була обрана стратегія охоплення ринку вибіркового розподілу, що застосовується до товарів, що користуються високим попитом у споживачів. Диференціація була обрана як стратегія розвитку, щоб представити новий підхід і отримати лояльних клієнтів через демонстрацію продукту.

Таблиця 6.14 — Опис потенційних груп

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Водії транспортних засобів	середня готовність	середній	невелика	середня
2.	Водії початківці	готові	високий	невелика	легка
3.	Люди що хочуть знати ПДР	готові	високий	велика	середня
4.	Пасажири транспортних засобів	неготові	низький	мала	важка
Які цільові групи обрано: водії транспортних засобів та пасажири					

Для ведення діяльності в обраному сегменті ринку була розроблена основна стратегія розвитку, яку представлено у таблиці 6.15.

Таблиця 6.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Розробка системи для формування поведінки на дорозі	Вибірковий розподіл	Розширення функціоналу, цінова політика, нововведення	Стратегія диференціації

За результатами проведеної роботи було обрано стратегію диференціації як базову стратегію конкурентної поведінки. Дана стратегія була вибрана, оскільки дана ніша потребує нового погляду на старі завдання, а також з метою охоплення нової аудиторії.

Основними вимогами до програмного продукту цільовою аудиторією я вивчення чи покращення правил дорожнього руху, а також формування індивідуальної якості водіння, саме тому головною позицією було обрано впровадження нових систем покращення поведінки на дорозі. Ключові асоціації – вивчення правил дорожнього руху, покращення поведінки на дорозі, формування рейтингу водія.

Висновки до розділу 6

В результаті аналізу ринкових можливостей розробленої системи було створено концепційний опис проєкту, ідентифіковані слабкі та сильні сторони. Технічний аудит, який включав вибір мови програмування Kotlin та використання

відкритих бібліотек (material, navigation, play-services-location, picasso) у середовищі розробки Android Studio, був проведений для реалізації проєкту.

Даний аналіз ринкових можливостей дозволив визначити конкурентоспроможність системи та визначити цільову аудиторію, якою є водії транспортних засобів та пасажери. Базуючись на цьому аналізі, була обрана стратегія диференціації, як основна стратегія розвитку. Для розробки була обрана стратегія охоплення ринку вибіркового розподілу, що застосовується до товарів із високим попитом.

Ключовими факторами успіху проєкту є якість вхідних даних та використання передових технологій GPS-відслідковування. Також визначено, що важливо слідкувати за розробками конкурентів і систематично удосконалювати функціонал проєкту.

ВИСНОВКИ

Під час виконання роботи було розроблено інструментальні засоби формування поведінки на дорозі, які направлені на підвищення рівня обізнаності правил дорожнього руху та покращення якості водіння учасників дорожнього руху задля зменшення кількості дорожньо транспортних пригод.

Було проведено аналіз програмних рішень, які реалізують схожий функціонал, проаналізовано їх особливості та інструментарії. Наявні рішення реалізують лише частковий функціонал запропонованої системи, вони або пропонують вивчити правила, часто без надання можливості перевірити свої знання, або навпаки, пройти тестування без можливості вивчити правила. Програми, які допомагають водію під час поїздки, збирають тільки часткову інформацію і не спонукають водія до покращення свого навичку. Таким чином, на даний момент не існує комплексного рішення, для теоретичного та практичного покращення знань та навичок водіння, хоча тема є актуальною, а тому розроблений програмний продукт повинен конкурувати з наявними застосунками й повністю задовольняти потреби користувача. Також було проаналізовано системи мотивування користувачів до використання програмних засобів. Було вирішено використати процес гейміфікації навчання в якості додаткового стимула покращувати знання та навички.

Створено модуль для вивчення правил дорожнього руху за рахунок правил та тестів правил дорожнього руху. Розроблений модуль надає повний спектр теоретичної підготовки користувачів, як до складання іспитів в сервісному центрі, так і до бездоганного знання та розуміння правил дорожнього руху

Створено модуль відслідковування якості водіння під час поїздки, що базується на показниках швидкості, прискорень, гальмувань та кількості перевищень швидкості під час руху. Даний модуль працює на базі GPS системи мобільного пристрою, він відслідковує емпіричні показники водіння і зберігає повну статистику поїздок користувача.

Реалізовано алгоритм, який вираховує рейтинг водіння користувача на базі статистики, зібраної під час поїздок, та якості проходження теоретичного тестування. Даний алгоритм більш направлений на зміну рейтингу під час реальних поїздок, ніж теоретичного тестування, оскільки саме від якості водіння залежить імовірність потрапляння у дорожньо транспортну пригоду.

Реалізовано базу даних для системи, яка складається з семи таблиць, які містять інформацію про теоретичні питання до тестів разом з відповідями, інформацію про водія разом з його досягненнями, а також статистичні дані отримані під час користування програмним забезпеченням.

Було проведено комплексне тестування програмного продукту під час симуляції поїздок за допомогою програмного забезпечення. Також були проведені тестування системи на реальних даних.

Дисертація виконана в межах проєкту Driver's Behavior Cognition – Розпізнавання поведінки водія на основі сенсорів мобільного телефону (спільно з Політехнічним інститутом м. Томар, Португалія).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Статистика. Патрульна поліція України. URL: <https://patrolpolice.gov.ua/statystyka/>
2. Stavros Nousias, Christos Tselios, Dimitris Bitzas, Dimitris Amaxilatis, Javier Montesa, Aris S. Lalos, Konstantinos Moustakas, Ioannis Chatzigiannakis. Exploiting Gamification to Improve Eco-driving Behaviour: The GamECAR Approach. 2019. 14 с.
3. Про дорожній рух : Закон України від 30.06.1993 р. № 3353-XII : станом на 1 січ. 2024 р. URL: <https://zakon.rada.gov.ua/laws/show/3353-12#Text>
4. Vodiу.ua. Онлайн підготовка і тестування водіїв. Vodiу.ua. Онлайн підготовка і тестування водіїв. URL: <https://vodiу.ua/>
5. Правила дорожнього руху України (ПДР 2023) онлайн на сайті green-way.com.ua. Green-way. URL: <https://green-way.com.ua/uk>
6. Digital Zone. Digital Zone. URL: <https://digitalzone.today/#apps>
7. KTW Apps. KTW Apps. URL: <https://ktwapps-ad94a.firebaseio.com/>
8. Contributors to Wikimedia projects. Gamification - Wikipedia. Wikipedia, the free encyclopedia. URL: <https://en.wikipedia.org/wiki/Gamification>
9. The Impact of Gamifications and Serious Games on Driving under Unfamiliar Traffic Regulations / H. Alyamani et al. Applied Sciences. 2023. Vol. 13, no. 5. P. 3262. URL: <https://doi.org/10.3390/app13053262>
10. Contributors to Wikimedia projects. Duolingo - Wikipedia. Wikipedia, the free encyclopedia. URL: <https://en.wikipedia.org/wiki/Duolingo>
11. John H. Android Programming with Kotlin for Beginners: Packt Publishing, 2019. 698 с.
12. Michael F. Kotlin and Android Development featuring Jetpack. Build Better, Safer Android Apps: Pragmatic Bookshelf, 2021. 446 с.

13. Alexey S., Anton A. Kotlin Design Patterns and Best Practices: Build scalable applications using traditional, reactive, and concurrent design patterns in Kotlin: Packt Publishing, 2022. 356 c.
14. Download Android Studio & App Tools - Android Developers. Android Developers. URL: <https://developer.android.com/studio/>
15. Neil S. Android Studio Development Essentials: CreateSpace Independent Publishing Platform, 2014. 816 c.
16. Room | Jetpack | Android Developers. Android Developers. URL: <https://developer.android.com/jetpack/androidx/releases/room>
17. Save data in a local database using Room | Android Developers. Android Developers. URL: <https://developer.android.com/training/data-storage/room>
18. Material Design. URL: <https://m3.material.io/>
19. Navigation | Jetpack | Android Developers. URL: <https://developer.android.com/jetpack/androidx/releases/navigation>
20. Build location-aware apps | Android Developers. Android Developers. URL: <https://developer.android.com/training/location>
21. Picasso. Square Open Source. URL: <https://square.github.io/picasso/>

ДОДАТКИ

ДОДАТОК А Лістинг розробленої програми

MainActivity.kt

```
package com.diplom.diplom_pdr.presentation

import android.Manifest
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import androidx.navigation.fragment.NavHostFragment
import com.diplom.diplom_pdr.R
import com.diplom.diplom_pdr.databinding.ActivityMainBinding
import com.diplom.diplom_pdr.models.UserEntity
import com.diplom.diplom_pdr.presentation.utils.viewmodels.MainViewModel
import org.koin.androidx.viewmodel.ext.android.viewModel
import java.util.Calendar

class MainActivity : AppCompatActivity() {
    private lateinit var _binding: ActivityMainBinding
```

```

private val viewModel: MainViewModel by viewModel()

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    _binding = ActivityMainBinding.inflate(layoutInflater).also {
setContentView(it.root) }
    requestPermission()
    viewModel.fillData()
    loadBottomNav()

    viewModel.userData.observe(this) {
        _binding.tvFlame.text = it.currentInterval.toString()
        _binding.tvStar.text = it.rating.toString()
    }

    _binding.root.postDelayed({
        viewModel.userData.value?.let {
            if (it.enterDate == null) {
                it.enterDate = System.currentTimeMillis()
                viewModel.updateUser(it)
            } else {
                val result = checkDate(it)
                if (!result) {
                    it.currentInterval = 0
                    viewModel.updateUser(it)
                }
            }
        }
    }, 1000)
}

```

```

    }, 1500)

}

private val requestPermissionLauncher = registerForActivityResult(
    ActivityResultContracts.RequestMultiplePermissions()
) { permissions ->
    Log.e("PERMISSIONS", "${permissions.entries}")
    permissions.entries.forEach { permission ->
        with(permission) {
            if (!value) {
                Toast.makeText(
                    this@MainActivity,
                    "Надайте дозвіл для використання програми",
                    Toast.LENGTH_SHORT
                ).show()
                this@MainActivity.finish()
                return@registerForActivityResult
            }
        }
    }
}
}

```

```

private fun checkDate(userEntity: UserEntity): Boolean {
    val systemTime = Calendar.getInstance()
    systemTime.timeInMillis = System.currentTimeMillis()
}

```

```

systemTime.clear(Calendar.HOUR_OF_DAY)
systemTime.clear(Calendar.MINUTE)
systemTime.clear(Calendar.SECOND)
systemTime.clear(Calendar.MILLISECOND)

val userTime = Calendar.getInstance()
userTime.timeInMillis = userEntity.enterDate!!

userTime.clear(Calendar.HOUR_OF_DAY)
userTime.clear(Calendar.MINUTE)
userTime.clear(Calendar.SECOND)
userTime.clear(Calendar.MILLISECOND)

val comparisonResult = systemTime.compareTo(userTime)
return when {
    comparisonResult == 0 -> true
    comparisonResult < 0 -> true
    else -> false
}
}

private fun requestPermission() {
    val permissions = mutableListOf<String>()

    if (ContextCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_FINE_LOCATION

```

```

        ) == PackageManager.PERMISSION_DENIED
    ) {
        permissions.add(Manifest.permission.ACCESS_FINE_LOCATION)
    }

    if (ContextCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_COARSE_LOCATION
    ) == PackageManager.PERMISSION_DENIED
    ) {
        permissions.add(Manifest.permission.ACCESS_COARSE_LOCATION)
    }

    Log.e("PERMISSIONS", "$permissions")
    if (permissions.isNotEmpty()) {
        requestPermissionLauncher.launch(permissions.toTypedArray())
    }
}

private fun loadBottomNav() {
    val navHost =
        supportFragmentManager.findFragmentById(R.id.main_container) as
        NavHostFragment
    val navController = navHost.navController

    _binding.bottomNavigation.setOnItemSelectedListener { item ->
        when (item.itemId) {

```

```
R.id.rules_page -> {  
    navController.navigate(R.id.rulesScreen)  
    true  
}
```

```
R.id.drive_page -> {  
    navController.navigate(R.id.driveScreen)  
    true  
}
```

```
R.id.test_page -> {  
    navController.navigate(R.id.testsScreen)  
    true  
}
```

```
R.id.profile_page -> {  
    navController.navigate(R.id.profileScreen)  
    true  
}
```

```
else -> false
```

```
}
```

```
}
```

```
}
```

```
}
```

Dao.kt

```
package com.diplom.diplom_pdr.db

import androidx.room.Dao
import androidx.room.Insert
import androidx.room.OnConflictStrategy
import androidx.room.Query
import androidx.room.Transaction
import com.diplom.diplom_pdr.models.Answer
import com.diplom.diplom_pdr.models.DriveStatsModel
import com.diplom.diplom_pdr.models.TaskItem
import com.diplom.diplom_pdr.models.TestsResultEntity
import com.diplom.diplom_pdr.models.ThemeItem
import com.diplom.diplom_pdr.models.ThemeItemWithTasks
import com.diplom.diplom_pdr.models.UserEntity

@Dao
interface Dao {

    @Query("SELECT * FROM UserEntity")
    suspend fun getUser(): UserEntity

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insertUser(userEntity: UserEntity)

    @Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insertAnswer(answer: Answer)
```

```
@Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insertAnswer(answers: List<Answer>)
```

```
@Query("SELECT * FROM answer WHERE taskItemQuestion =:question")
```

```
suspend fun getAnswerList(question: String): List<Answer>
```

```
@Query("SELECT * FROM task_item WHERE is_favorite = :isFavorite")
```

```
suspend fun getFavoriteTasks(isFavorite: Boolean): List<TaskItem>
```

```
@Query("DELETE FROM drive_stats")
```

```
suspend fun deleteDriveStats()
```

```
@Query("SELECT * FROM drive_stats")
```

```
suspend fun getDriveStats(): List<DriveStatsModel>
```

```
@Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insertDriveStats(driveStatsModel: DriveStatsModel)
```

```
@Query("UPDATE theme_item SET test_time = :totalTestTime WHERE id = :id")
```

```
suspend fun updateTotalTestTime(id: Int, totalTestTime: Long)
```

```
@Query("UPDATE theme_item SET is_test_passed = :isTestPassed WHERE id = :id")
```

```
suspend fun updateIsTestPassed(id: Int, isTestPassed: Boolean)
```

```
@Query("UPDATE theme_item SET wrong_answers = :wrongAnswers WHERE id = :id")
```

```
suspend fun updateWrongAnswers(id: Int, wrongAnswers: Int)
```

```
@Query("UPDATE theme_item SET right_answers = :rightAnswers WHERE id = :id")
```

```
suspend fun updateRightAnswers(id: Int, rightAnswers: Int)
```

```
@Query("SELECT * FROM tests_result")
```

```
fun getTestsResult(): TestsResultEntity
```

```
@Query("DELETE FROM tests_result")
```

```
suspend fun deleteTestResult()
```

```
@Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insertTestResult(testsResultEntity: TestsResultEntity)
```

```
@Query("SELECT * FROM task_item WHERE themeItemTitle =:title")
```

```
fun getAllTasks(title: String): List<TaskItem>
```

```
@Query("DELETE FROM theme_item")
```

```
suspend fun deleteAllThemes()
```

```
@Query("SELECT * FROM theme_item")
```

```
fun getAllThemes(): List<ThemeItem>
```

```
@Transaction
```

```
@Query("SELECT * FROM theme_item")
```

```
suspend fun getThemeItemWithTasks(): List<ThemeItemWithTasks>
```

```
@Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insertThemeItem(themeItem: ThemeItem)
```

```
@Query("SELECT * FROM theme_item where title = :title")
```

```
suspend fun getThemeItem(title: String): ThemeItem
```

```
@Query("DELETE FROM task_item")
```

```
suspend fun deleteTaskItem()
```

```
@Query("SELECT * FROM task_item where id=:id")
```

```
suspend fun getTaskItem(id: Int): TaskItem
```

```
@Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insertTaskItem(taskItem: TaskItem)
```

```
@Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insertListTaskItem(taskItem: List<TaskItem>)
```

```
@Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insertListThemeItem(themeItem: List<ThemeItem>)
```

```
@Query("UPDATE task_item SET is_favorite = :isFavorite WHERE id = :id")
```

```
fun updateIsFavoriteTaskItem(id: Int, isFavorite: Boolean)
```

```
}
```

LocalStorage.kt

```
package com.diplom.diplom_pdr.controller

import android.content.Context
import com.diplom.diplom_pdr.R
import com.diplom.diplom_pdr.db.Dao
import com.diplom.diplom_pdr.models.Answer
import com.diplom.diplom_pdr.models.DriveStatsModel
import com.diplom.diplom_pdr.models.TaskItem
import com.diplom.diplom_pdr.models.TestsResultEntity
import com.diplom.diplom_pdr.models.ThemeItem
import com.diplom.diplom_pdr.models.UserEntity
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.coroutineScope
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext
import java.nio.charset.StandardCharsets

class LocalStorage(private val context: Context, private val dao: Dao) {

    suspend fun insertUser(userEntity: UserEntity) = dao.insertUser(userEntity)

    suspend fun getUser() = dao.getUser()

    suspend fun insertAnswer(answer: Answer) = dao.insertAnswer(answer)
```

```
suspend fun getAnswerList(question: String) = dao.getAnswerList(question)
```

```
suspend fun getRandQuestions(count: Int): List<TaskItem> {
```

```
    return try {
```

```
        val themes = dao.getThemeItemWithTasks()
```

```
        val allTasks = themes.flatMap { it.taskItem }
```

```
        val randomNums = mutableSetOf<Int>()
```

```
        while (randomNums.size < count) {
```

```
            val randNum = (allTasks.indices).random()
```

```
            randomNums.add(randNum)
```

```
        }
```

```
        val randTasks = mutableListOf<TaskItem>()
```

```
        randomNums.forEachIndexed { index, i ->
```

```
            randTasks.add(allTasks[i])
```

```
        }
```

```
        randTasks
```

```
    } catch (e: Exception) {
```

```
        emptyList<TaskItem>()
```

```
    }
```

```
}
```

```
suspend fun getFavoriteTasks() = dao.getFavoriteTasks(true)
```

```
suspend fun deleteTaskItem() = dao.deleteTaskItem()
```

```
suspend fun deleteThemes() = dao.deleteAllThemes()
```

```
suspend fun deleteTestResult() = dao.deleteTestResult()
```

```
suspend fun deleteDriveStats() = dao.deleteDriveStats()
```

```
suspend fun getDriveStats() = dao.getDriveStats()
```

```
suspend fun saveResultDrive(driveStatsModel: DriveStatsModel) =  
    dao.insertDriveStats(driveStatsModel)
```

```
fun getAllThemes() = dao.getAllThemes()
```

```
suspend fun updateTotalTestTime(title: String, totalTestTime: Long) {  
    val theme = getThemeItem(title)  
    dao.updateTotalTestTime(theme.id.toInt(), totalTestTime)  
}
```

```
suspend fun updateIsTestPassed(title: String, isTestPassed: Boolean) {  
    val theme = getThemeItem(title)  
    dao.updateIsTestPassed(theme.id.toInt(), isTestPassed)  
}
```

```
suspend fun updateWrongAnswers(title: String, wrongAnswers: Int) {  
    val theme = getThemeItem(title)
```

```
        dao.updateWrongAnswers(theme.id.toInt(), wrongAnswers)
    }
```

```
suspend fun updateRightAnswers(title: String, rightAnswers: Int) {
    val theme = getThemeItem(title)
    dao.updateRightAnswers(theme.id.toInt(), rightAnswers)
}
```

```
fun getTestsResult() = dao.getTestsResult()
```

```
suspend fun insertTestResult(testsResultEntity: TestsResultEntity) =
    dao.insertTestResult(testsResultEntity)
```

```
fun getAllTasks(title: String): List<TaskItem> = dao.getAllTasks(title)
```

```
suspend fun updateTask(taskItem: TaskItem) {
    dao.insertTaskItem(taskItem)
}
```

```
fun setFavorite(id: Int, isFavorite: Boolean) {
    dao.updateIsFavoriteTaskItem(id, isFavorite)
}
```

```
suspend fun getThemeItem(title: String) = dao.getThemeItem(title)
```

```
suspend fun getCurrentTaskItem(id: Int) = dao.getTaskItem(id)
```

```
suspend fun getAllData() = dao.getThemeItemWithTasks()
```

```
suspend fun loadNewDBTasks() {  
    putListThemes()  
    val listThemes = dao.getThemeItemWithTasks()  
    val assetsManager = context.assets  
    for (i in 1..33) {  
        coroutineScope {  
            launch {  
                withContext(Dispatchers.IO) {  
                    val input = assetsManager.open("test$i.txt")  
                    val bytes = input.readBytes()  
                    val text = String(bytes, StandardCharsets.UTF_8)  
                    input.close()  
  
                    val parse = text.split("<newline>")  
  
                    for (task in 0 until parse.size - 1) {  
                        val parseTask = parse[task].split("|")  
                            .filter { it.isNotBlank() }  
                            .map { it }  
  
                        val img = parseTask[0]  
                        val question = parseTask[1]  
                        val answers =  
                            parseTask
```

```

        .slice(2 until parseTask.size - 1)
        .map { "$it|" }
        .map {
            Answer(
                name = it,
                type = Answer.TYPE.DEFAULT,
                taskItemQuestion = question
            )
        }

dao.insertAnswer(answers)

val rightAnswer = parseTask[parseTask.size - 1]

dao.insertTaskItem(
    TaskItem(
        question = question,
        rightAnswer = rightAnswer,
        imgURL = img,
        themeItemTitle = listThemes[i - 1].themeItem.title,
    )
)
}
}
}
}
}
}
}
}
}
}

```

```
}
```

```
suspend fun checkIsDBEmpty(): Boolean =  
dao.getThemeItemWithTasks().isEmpty()
```

```
private suspend fun putListThemes() = with(context) {  
    val list = listOf(  
        ThemeItem(title = getString(R.string.general_terms)),  
        ThemeItem(title = getString(R.string.right_drivers)),  
        ThemeItem(title =  
getString(R.string.movement_of_vehicles_with_special_signals)),  
        ThemeItem(title =  
getString(R.string.responsibilities_and_rights_of_pedestrians)),  
        ThemeItem(title =  
getString(R.string.responsibilities_and_rights_of_passengers)),  
        ThemeItem(title = getString(R.string.requirements_for_cyclists)),  
        ThemeItem(title =  
getString(R.string.requirements_for_persons_driving_horse_drawn)),  
        ThemeItem(title = getString(R.string.traffic_regulation)),  
        ThemeItem(title = getString(R.string.warning_signals)),  
        ThemeItem(title =  
getString(R.string.starting_a_movement_and_changing_its_direction)),  
        ThemeItem(title = getString(R.string.location_of_vehicles_on_the_road)),  
        ThemeItem(title = getString(R.string.speed_of_movement)),  
        ThemeItem(title = getString(R.string.distance_interval_oncoming_traffic)),  
        ThemeItem(title = getString(R.string.overtaking)),  
        ThemeItem(title = getString(R.string.stopping_and_parking)),  
        ThemeItem(title = getString(R.string.passing_intersections)),  
        ThemeItem(title = getString(R.string.advantages_of_route_vehicles)),
```

```

        ThemeItem(title =
getString(R.string.passing_pedestrian_crossings_and_vehicle_stops)),
        ThemeItem(title = getString(R.string.use_of_external_lighting_devices)),
        ThemeItem(title = getString(R.string.moving_through_railroad_crossings)),
        ThemeItem(title = getString(R.string.passenger_transportation)),
        ThemeItem(title = getString(R.string.cargo_transportation)),
        ThemeItem(title =
getString(R.string.towing_and_operation_of_transport_trains)),
        ThemeItem(title = getString(R.string.training_ride)),
        ThemeItem(title = getString(R.string.movement_of_vehicles_in_columns)),
        ThemeItem(title =
getString(R.string.traffic_in_residential_and_pedestrian_areas)),
        ThemeItem(title =
getString(R.string.driving_on_highways_and_roads_for_cars)),
        ThemeItem(title =
getString(R.string.driving_on_mountain_roads_and_steep_descents)),
        ThemeItem(title = getString(R.string.international_movement)),
        ThemeItem(title =
getString(R.string.number_plates_identification_marks_inscriptions_and_designations))
    ,
        ThemeItem(title =
getString(R.string.technical_condition_of_vehicles_and_their_equipment)),
        ThemeItem(title =
getString(R.string.some_issues_of_road_traffic_organization_that_need_to_be_coordin
ated)),
        ThemeItem(title = getString(R.string.road_signs)),
    )

    dao.insertListThemeItem(list)
}
}

```

ДОДАТОК Б Апробація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XX Міжнародної
науково-практичної конференції
молодих вчених і студентів
(присвячена 125-річчю КПІ ім. Ігоря Сікорського та 90-
річчю НН ІАТЕ (ТЕФ))
м. Київ, 25–28 квітня 2023 року

ТОМ 2



Київ- 2023

<i>Керівник - доц., д.т.н. Коваль О.В.</i>	
Інструментальні засоби навігації в транспортних системах.	120
<i>ЄЗГОР В.С., магістрант гр. ТВ-21мн</i>	
<i>Керівник - доц., к.т.н. Гусєва І.І.</i>	
Програмний застосунок формування навчальних планів.	122
<i>КОВТУН А.С., магістрант гр. ТВ-22мн</i>	
<i>Керівник - доц., к.е.н. Гусєва І.І.</i>	
Інструментальні засоби формування поведінки на дорозі.	124
<i>СЛАВСЬКИЙ С.В., магістрант гр. ТВ-21мн</i>	
<i>Керівник - доц., к.е.н. Гусєва І.І.</i>	
Аналіз засобів пошуку інформації для побудови класифікаторів оцінювання результатів моделювання складних систем.	126
<i>ЛОГВІНЕНКО Т.С., магістрант гр. ТВ-91мн</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
Нейромережеві підходи до генерації акустичних сигналів водного середовища.	128
<i>ОЛЕКСІЙ А.О., аспірант</i>	
<i>Керівник - проф., к.ф.-м.н. Верлань А.А.</i>	
Моделі та методи опису та проведення тестування програмних продуктів на прикладі тестування веб-додатку кабінету аспіранта кафедри.	130
<i>ПОЛОВІНКИН П.О., магістрант гр. ТВ-14мн</i>	
<i>Керівник - доц., д.т.н. Недашківський О.Л.</i>	
Модуль обміну даними з банками інформаційної системи ODOO.	132
<i>ЗАСТУПАЙЛО М.І., студент гр. ТМ-92</i>	
<i>Керівник - доц., к.т.н. Шушура О.М.</i>	
Інструментальні засоби розпізнавання маневрів транспортних засобів.	134
<i>ЛОЛА Н.О., студент гр. ТІ-91</i>	
<i>Керівник - доц., к.е.н. Гусєва І.І.</i>	
Інструментальні засоби формування керованих даними стратегій водіння.	136
<i>ЛУКІНСЬКИЙ Д.Д., студент гр. ТВ-91</i>	
<i>Керівник - доц., к.е.н. Гусєва І.І.</i>	
Інструментальні засоби експертизи транспортних засобів після дорожньо-транспортних пригод.	138
<i>МАКСИМЕНКО П.О., студент гр. ТІ-92</i>	
<i>Керівник - доц., к.е.н. Гусєва І.І.</i>	
Розробка порталів для інтеграції геоінформаційних WEB сервісів з хмарним середовищем.	140
<i>РЕГЕДА М.Ю., студент гр. ТР-93</i>	
<i>Керівник - ст.викл. Гурін А.Л.</i>	
Інструментальні засоби краудсорсингу в транспортних системах.	142
<i>ТЮТЮННИК О.Г., студент гр. ТІ-92</i>	
<i>Керівник - доц., к.е.н. Гусєва І.І.</i>	
Інструментальні засоби автоматичного контролю кондиціонера для максимальної енергоефективності використання палива.	144
<i>ЯРИНИЧ В.П., студент гр. ТІ-92</i>	
<i>Керівник - доц., к.е.н. Гусєва І.І.</i>	
СЕКЦІЯ №10 КОМП'ЮТЕРНИЙ ЕКОЛОГО-ЕКОНОМІЧНИЙ МОНІТОРИНГ ТА ГЕОМЕТРИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ І СИСТЕМ	146

УДК 004.78

Магістрант 1 курсу, гр. ТВ-21мп Славський С.В.
Доц., к.е.н. Гусєва І.І.

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ФОРМУВАННЯ ПОВЕДІНКИ НА ДОРОЗІ

Метою дослідження є розробка та впровадження мобільного програмного застосунку формування покращеної поведінки на дорозі для збільшення рівня обізнаності населення та зменшення кількості аварій та дорожньо-транспортних пригод за рахунок кращих знань правил та надбаних звичок ввічливого та безпечного водіння.

Дорожньо-транспортна пригода (ДТП) — подія, що сталася під час руху транспортного засобу, внаслідок якої загинули або поранені люди чи завдані матеріальні збитки.

Офіційна статистика дорожньо-транспортних пригод в Україні продемонстрована на рис. 1 зазначає, що кожного року відбувається близько 150000 ДТП, які були задокументовані та офіційно оформленні, кількість неоформлених невідома і може перевищувати загальну кількість в декілька разів [1].

Якщо врахувати кількість автомобілів українського автопарку, а це трохи більше 10 млн. можна дізнатись, що кожного року кожен 50 водіїв потрапляє в ДТП. Переважна більшість дорожніх пригод відбувається через недотримання правил дорожнього руху, через їх незнання або самовпевненість водіїв.



Рисунок 1 – Статистика ДТП в Україні

Основною метою розробки інструментальних засобів формування поведінки на дорозі є створення мобільного застосунку для щоденного використання, яке змотивує та покращить теоретичні знання правил, а також практичні навички водіння за рахунок накопичення даних поведінки водія на дорозі та системи заохочення вивчення та повторення базових, а також розширених правил дорожнього руху та поведінки за кермом автомобіля [2].

Для досягнення поставлених цілей слід використати кросплатформну мову програмування мобільних додатків Kotlin, яка забезпечить оперативну розробку додатку, а також дозволить масштабувати її на різні платформи для збільшення кількості користувачів [3].

Для збереження даних, які накопичує програма під час роботи, було прийнято рішення використовувати швидку та надійну базу даних SQLite за допомогою бібліотеки Room. Room — це високорівневий інтерфейс для вбудованих низькорівневих прив'язок SQLite в Android. Він виконує більшу частину роботи під час компіляції, створюючи API над вбудованим API SQLite, що значно полегшує роботу з базами даних Android [4].

Отже за рахунок інструментальних засобів, що розробляються, необхідно досягти

покращення знань правил дорожнього руху, їх щоденне дотримання та в перспективі сформулювати принципи та засади поведінки на дорозі, що унеможливить майбутні нещасні випадки та смерті.

В результаті проведеної роботи буде представлений зручний та інтуїтивно зрозумілий мобільний додаток, що виконує відразу декілька важливих функцій, а саме: накопичує та опрацьовує кількісні та якісні показники водія під час руху, на основі яких сповіщає про порушення та недоліки водіння; надає інструмент для вивчення та покращення теоретичних і практичних знань правил дорожнього руху; пропонує механізм мотивації навчання правил та утримання зацікавленості користувачів програмним продуктом протягом довгого періоду часу [5].

Поведінка на дорозі це сукупність роботи кожного її учасника, а отже формувати цю поведінку необхідно у кожній особистості, не зважаючи на те чи це досвідчений водій, чи новачок, який тільки отримав права, чи пішохід, який переходить дорогу.

Перелік посилань:

1. Статистика. Патрульна поліція України.URL:
<https://patrolpolice.gov.ua/statystyka/>.
2. Stavros Nousias, Christos Tselios, Dimitris Bitzas, Dimitris Amaxilatis, Javier Montes, Aris S. Lalos, Konstantinos Moustakas, Ioannis Chatzigiannakis. Exploiting Gamification to Improve Eco-driving Behaviour: The GamECAR Approach. 2019. 14 с.
3. Michael F. Kotlin and Android Development featuring Jetpack. Build Better, Safer Android Apps: Pragmatic Bookshelf, 2021. 446 с.
4. SQLite Tutorial - An Easy Way to Master SQLite Fast. SQLite Tutorial. URL:
<https://www.sqlitetutorial.net/>.
5. Jiawei WU. Analysis of taxi drivers'driving behavior based on a driving simulator experiment. 2014. 112 с.

ДОДАТОК В Презентація



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Кафедра Інженерії програмного забезпечення в енергетиці.



Інструментальні засоби формування поведінки на дорозі

Виконав: студент магістерського рівня 2-го року навчання, групи ТВ-21мп

Славський Святослав Володимирович

Керівник: к.е.н., доцент Гусєва Ірина Ігорівна



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АКТУАЛЬНІСТЬ РОБОТИ

Поведінка на дорозі – це сукупність формальних та неформальних правил, які регулюють взаємодію учасників дорожнього руху



Статистика ДТП в Україні



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПОСТАНОВКА ЗАВДАННЯ

Об'єкт дослідження: формування звичок та вивчення правил дорожнього руху у водіїв автомобілів.

Предмет дослідження: програмне забезпечення для формування поведінки на дорозі на основі вивчення правил дорожнього руху та сформованої статистики поїздки.

Метою дослідження є розробка мобільного програмного застосунку формування поведінки на дорозі.

Для досягнення поставленої мети необхідно реалізувати наступні завдання:

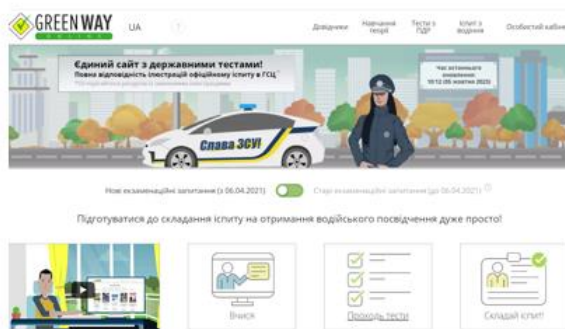
1. Провести дослідження та аналіз наявних засобів формування знань та навичок їзди.
2. Розробити модуль теоретичної підготовки для вивчення правил дорожнього руху.
3. Розробити модуль практичної підготовки на основі даних, отриманих під час поїздки.
4. Розробити алгоритм визначення рейтингу поведінки.
5. Розробити базу даних системи.
6. Протестувати розроблений програмний продукт в умовах реальної експлуатації.



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

3

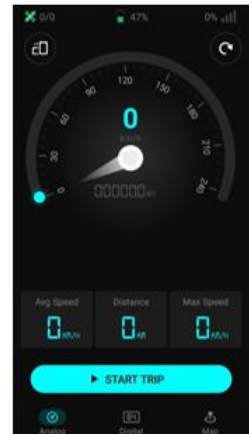
АНАЛОГИ СИСТЕМИ



Сайт green-way.com.ua



Знаки ПДР



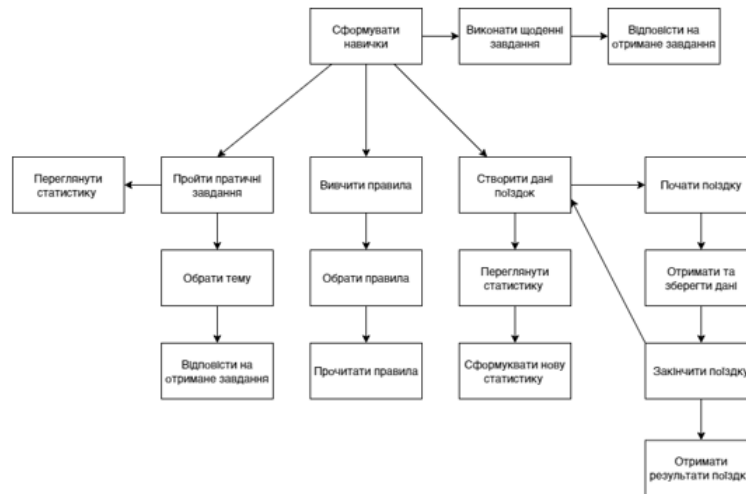
GPS спідометр



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

4

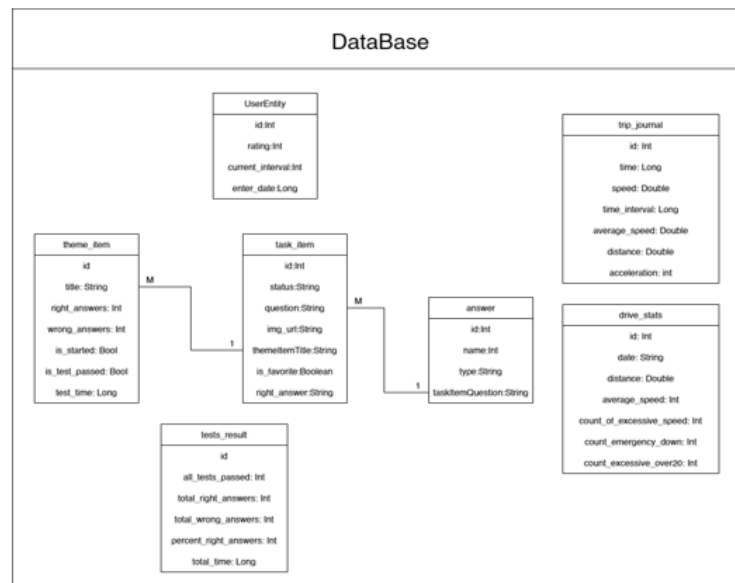
ФУНКЦІОНАЛЬНА ДЕКОМПОЗИЦІЯ



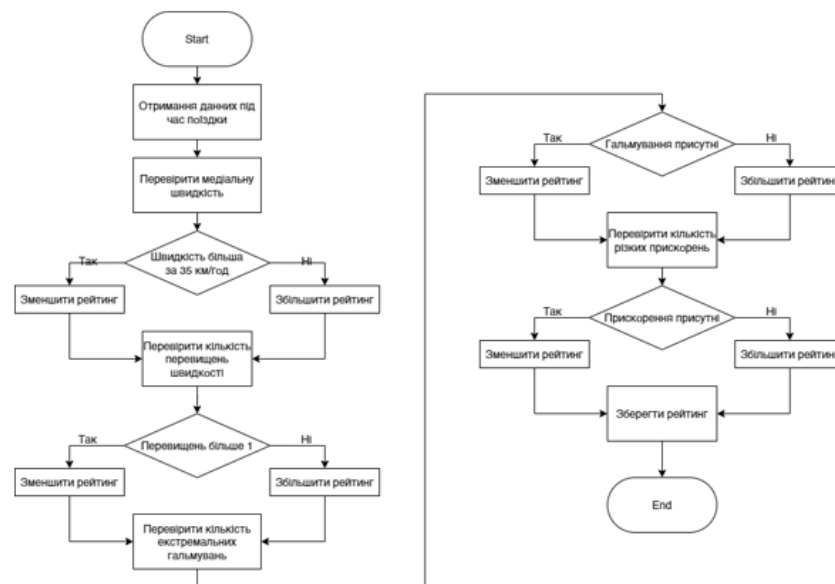
ДІАГРАМА ПРЕЦЕДЕНТІВ



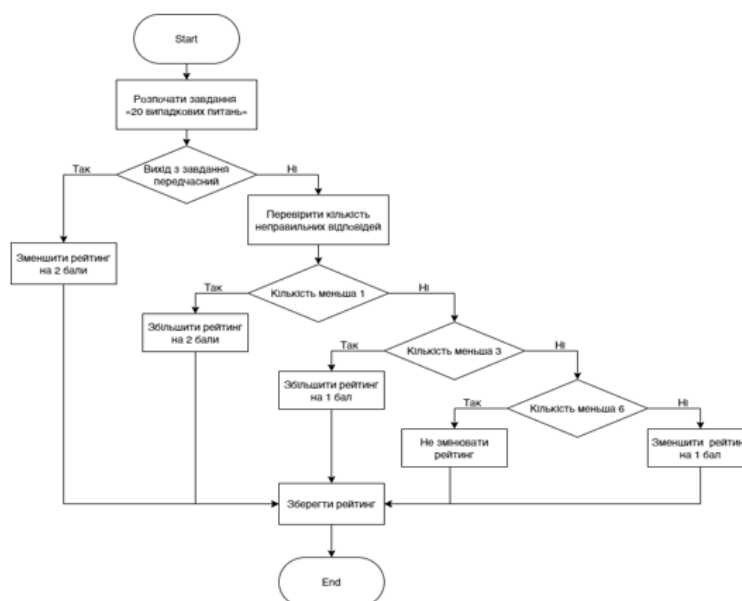
СТРУКТУРА БАЗИ ДАНИХ



АЛГОРИТМ ЗМІНИ РЕЙТИНГУ ЗА РЕЗУЛЬТАТАМИ ПОЇЗДКИ



АЛГОРИТМ ЗМІНИ РЕЙТИНГУ ЗА ПРОЙДЕНИМ ТЕСТУВАННЯМ



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

9

ЗАСОБИ РОЗРОБКИ



Kotlin – мова програмування
мобільних додатків



Room – база даних
побудована на основі SQLite



Android Studio – середовище
розробки мобільних пристроїв



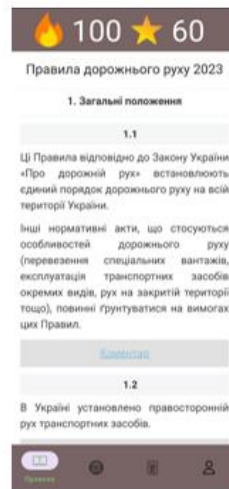
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

10

ПРАВИЛА ДОРОЖНЬОГО РУХУ ТА ПРОФІЛЬ



Екран з темами ПДР



Екран ПДЗ за темою



Екран профілю користувача



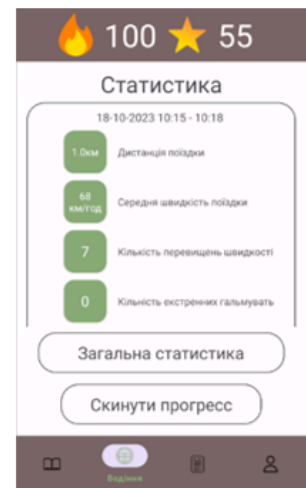
ЕКРАНИ ПОЇЗДКИ



Екран під час поїздки



Екран після поїздки



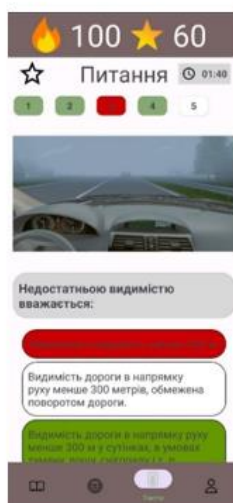
Екран статистики поїздки



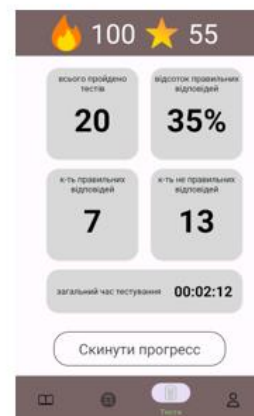
ЕКРАНИ ПРОХОДЖЕННЯ ТЕСТІВ



Екран з вибором тестів



Екран проходження тесту



Екран статистики тестів



ВИСНОВКИ

Під час виконання роботи було реалізовано інструментальні засоби формування поведінки на дорозі, які направлені на підвищення рівня обізнаності правил дорожнього руху, та покращення рівня їзди учасників дорожнього руху.

- Було проведено аналіз програмних рішень, які реалізують схожий функціонал. Наявні рішення реалізують лише частковий функціонал системи що розробляється, одні пропонують тільки вивчити правила, інші тільки пройти тестування. Програми які допомагають водію під час поїздки, збирають тільки часткову інформацію і не спонукають водія до покращення свого навичку. Таким чином на даний момент не існує комплексного рішення, а тому розроблений програмний продукт повинен конкурувати з наявними застосунками й повністю задовольняти потреби користувача
- Створено модуль для вивчення правил дорожнього руху, за рахунок правил та тестів правил дорожнього руху.
- Створено модуль відслідковування якості водіння під час поїздки, що базується на показниках швидкості, прискорень, гальмувань та кількості перевищень швидкості під час руху.
- Реалізовано алгоритм, який вираховує рейтинг водіння користувача, на базі статистики зібраної під час поїздок, та якості проходження теоретичного тестування.
- Реалізовано базу даних для системи, яке складається з семи таблиць.
- Було проведено комплексне тестування програмного продукту на реальних даних.





Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"