ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА

Пояснювальна записка
до кваліфікаційної роботи бакалавра

на тему:
Розробка мобільного додатку на платформі iOS для «Таємного покупця»

Виконала:
Здобувач вищої освіти, групи КН 2022-1
спеціальності
122 «Комп'ютерні науки»
(шифр і назва спеціальності)

Катерина БЄЛОВА
(прізвище та ініціали)

Керівник:

к.т.н., доц. Володимир БРЕДІХІН
(прізвище та ініціали)
Рецензент:

д.т.н., проф. Анатолій ЛИТВИНОВ
(прізвище та ініціали)

м. Харків – 2026 рік

Харківський національний університет міського господарства імені О. М. Бекетова
(повне найменування закладу вищої освіти)
Навчально-науковий Інститут енергетичної, інформаційної
та транспортної інфраструктури
Кафедра комп'ютерних наук та інформаційних технологій
Рівень вищої освіти перший (бакалаврський)
Спеціальність 122 «Комп'ютерні науки»
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри КНтаІТ
Марина НОВОЖИЛОВА
« 22 » червня 2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Бєловій Катерині Олегівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка мобільного додатку на платформі iOS для «Таємного покупця»
керівник роботи к.т.н., доцент Бредіхін В.М.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «22» травня 2026 р. № 440-03
2. Термін подання студентом роботи 20.06.2026р.
3. Вихідні дані до роботи Рекомендації для розробки мобільного додатку на платформі iOS для «Таємного покупця»
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
проаналізувати предметну область та прототипи системи; обґрунтувати вибір інструментального середовища та технічної платформи; виконати функціональний аналіз додатку; розробити програмну реалізацію та інтерфейс взаємодії з користувачем; описати архітектуру, інтерфейс розробки мобільного додатку для роботи таємного покупцята та перевірити функціонування програмного забезпечення.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентація – 15 аркушів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Володимир БРЕДІХІН	11.05.2026	10.05.2026
Розділ II	Володимир БРЕДІХІН	17.05.2026	15.05.2026
Розділ III	Володимир БРЕДІХІН	21.05.2026	30.05.2026
Розділ IV	Вікторія МАЛИШЕВА	27.05.2026	15.06.2026

7. Дата видачі завдання 10.04.2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми дипломної роботи	03.05.2026	Викон.
2	Затвердження тем, наукових керівників, завдань та календарного плану підготовки дипломної роботи	05.05.2026	Викон.
3	Написання I розділу	10.05.2026	Викон.
4	Написання II розділу	15.05.2026	Викон.
5	Написання III розділу	20.05.2026	Викон.
6	Написання IV розділу	30.05.2026	Викон.
7	Подання дипломної роботи керівнику	05.06.2026	Викон.
8	Робота по усуненню зауважень керівника, уточнення і доповнення практичного матеріалу, оформлення додатків до роботи	10.06.2026	Викон.
9	Подання доопрацьованого варіанту роботи керівнику	15.06.2026	Викон.
10	Захист матеріалів дипломної роботи на засіданні кафедри	18.06.2026	Викон.
11	Офіційний захист матеріалів дипломної роботи на засіданні екзаменаційної комісії	22.06.2026	Викон.

Студент _____ (підпис) Катерина БЄЛОВА (прізвище та ініціали)

Керівник роботи _____ (підпис) Володимир БРЕДІХІН (прізвище та ініціали)

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка кваліфікаційної роботи здобувача вищої освіти групи КН 2022-1 спеціальності 122 «Комп’ютерні науки» Бєлової Катерини Олегівни за темою «Розробка мобільного додатку на платформі iOS для «Таємного покупця» складається з 4 розділів, містить 64 сторінок тексту, 15 рисунків, 7 таблиць, 39 джерела.

Дана робота присвячена актуальній темі, а саме, створенню мобільного додатку на платформі iOS для «Таємного покупця».

Мета дослідження – розробка мобільного додатка на платформі iOS для автоматизації діяльності «таємного покупця» та підвищення ефективності контролю якості обслуговування.

Об'єктом дослідження є процес функціонування мобільного додатка для збору, обробки та передачі даних про результати перевірок.

Предмет дослідження – методи, моделі та програмні засоби створення мобільного додатка для підтримки роботи «таємного покупця».

У першому розділі виконано аналіз предметної області, розглянуто сутність методу «таємного покупця», досліджено існуючі програмні аналоги та визначено їхні переваги й недоліки, а також обґрунтовано необхідність розробки власного програмного рішення.

У другому розділі здійснено проєктування системи, визначено функціональні та нефункціональні вимоги до мобільного додатка, побудовано UML-діаграми, описано архітектуру системи та обґрунтовано вибір технологій реалізації.

У третьому розділі описано процес програмної реалізації мобільного додатка, включаючи розробку моделей даних, контролерів та інтерфейсів користувача, організацію взаємодії з сервером, а також проведення тестування для перевірки працездатності системи.

У четвертому розділі розглядаються питання охорони праці.

Ключові слова: МОБІЛЬНИЙ ДОДАТОК, АВТОРИЗАЦІЯ, КОРИСТУВАЧ, БАЗА ДАНИХ, ANDROID ДОДАТОК

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ 10
1.1 Постановка проблеми 10
1.2 Мобільний додаток для «Тайного покупця» 14
1.3 Ананаліз подібних проектів 17
РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ 21
2.1 Аналіз вимог та опис функціонального призначення системи 21
2.1.1 Функціональні вимоги 21
2.1.2 Нефункціональні вимоги 22
2.2 Варіанти використання системи 22
2.3 Компоненти системи 24
2.4 Діаграми послідовності 26
2.5 Ананаліз існуючих рішень для реалізації додатку 29
Висновки до другого розділу 32
РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ 33
3.1 Моделювання взаємодії програмних компонентів 33
3.2 Реалізація контролерів 35
3.3 Реалізація інтерфейсів 38
3.4 Реалізація взаємодії з сервером 39
3.5 Функціональне тестування 43
3.5.1 Unit-тестування 45
3.5.2 Інтеграційне тестування 46
Висновок до третього розділу 47
РОЗДІЛ 4 ОХОРОНА ПРАЦІ 48
4.1 Організаційно-правові основи забезпечення безпеки праці 48
4.2 Характеристика об’єкта та виявлення потенційних небезпек 49
4.3 Дослідження ризику реалізації потенційних небезпек на об’єкті проектування та розробка заходів щодо їх попередження 52

Висновок до четвертого розділу 56
ЗАГАЛЬНІ ВИСНОВКИ 58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 60

ВСТУП

У наші дні власникам роздрібних мереж під час розширення бізнесу стає все важче контролювати кожен елемент мережі, а саме відстежувати рівень відповідності стандартам компанії. Для вирішення цієї проблеми існує метод «таємний покупець», який застосовується в рамках маркетингового дослідження, спрямованого на оцінку споживчого досвіду, отриманого клієнтом у процесі придбання товару чи послуги, так і з метою вирішення організаційних завдань, а саме завдань торговельного аудиту, контролю торговельного обладнання та встановлених цін. Контроль торгових точок необхідний для залучення більшої кількості споживачів, особливо важливо в умовах високої ринкової конкуренції.

На сьогодні мобільні пристрої є найбільш популярними і повсюдно використовуваними девайсами, а також починають перевершувати інші пристрої за часом використання. За даними на кінець 2015 року, 75% інтернет-користувачів виходили в онлайн за допомогою мобільних пристроїв, також відомо, що користувачі від 16 до 24 років витрачають у середньому 3.5 години на день за смартфоном [1]. В останні роки ринок мобільних додатків суттєво зростає у всіх його областях. У 2019 році ринок мобільних додатків оцінювався в 53 мільярди доларів, а вже в 2024 році він наблизився до оцінної вартості у 100 мільярдів доларів [2].

Виходячи з наведених даних, можна зробити висновок, що актуально буде розробити мобільний додаток на платформі iOS для «таємного покупця», завдяки якому власники різних підприємств зможуть здійснювати контроль торгових точок та проводити маркетингові дослідження для оцінки якості та покращення вироблених товарів та послуг.

Метою цієї роботи є розробка мобільного додатка на платформі iOS для «таємного покупця». Для досягнення поставленої мети необхідно виконати такі завдання:

1) проаналізувати предметну область;

2) спроектувати мобільний додаток;
3) реалізувати мобільний додаток;
4) протестувати мобільний додаток.

РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Постановка проблеми

Таємний покупець – це метод досліджень, який компанії використовують, щоб оцінити купівельний досвід та роботу співробітників у торговій точці [3].

Рисунок 1.1 – Таємний покупець

Метод дослідження «таємний покупець» використовується власниками бізнесу для таких цілей:

Моніторити якість обслуговування клієнтів. Якщо використовувати цю техніку постійно, наприклад, кілька разів на місяць, можна зрозуміти, наскільки якість обслуговування відповідає вашим вимогам і чи змінюється вона після того, як ви вказуєте співробітником на можливості для покращення;

Виявляти професійний рівень працівників. Важливо розуміти, наскільки співробітники є компетентними і можуть допомогти клієнту з вибором. Для цього продавці повинні добре знати асортимент, швидко перевіряти його наявність на складі, знати, з яких продуктів готуються страви в їхньому

ресторані та інше [4].

Аналізувати якість обслуговування конкурентів. Іноді цей метод використовують, щоб порівняти обслуговування у своєму магазині з магазинами конкурентів. При цьому таємний покупець не повинен робити нічого, що може завдати шкоди репутації магазину або закладу, тому що тоді це перетворюється на неетичний метод конкуренції.

Цей метод не замінює інші методи оцінки якості роботи персоналу, такі як атестація, тестування, спостереження, а доповнює їх.

Таємний покупець – це методологія спостережливого дослідження, яка гарантує, що бренди послідовно виконують обіцянки, дані клієнтам. Вона використовує навчених «покупців», які перевіряють взаємодію з продуктами або послугами по всьому підприємству та звітують про свій досвід, надаючи детальний та об’єктивний звіт. Таємний покупець допомагає компаніям зрозуміти, з чим стикається їхній типовий клієнт, яку поведінку персоналу слід визнавати або винагороджувати, а також які частини процесу продажів або обслуговування клієнтів можна покращити [5].

Сьогодні програму таємного покупця можна розгорнути в будь-якому каналі зв'язку з клієнтами: у магазині, по телефону (роздрібний магазин або контакт-центр), а також в онлайн- та мобільних точках контакту (вебсайт, додаток, електронна пошта тощо).

Результати агрегуються та використовуються для виявлення та усунення системних збоїв у бажаному та запланованому досвіді клієнтів. За умови ефективного використання, дослідження методом таємного покупця може бути потужним інструментом управління ефективністю, який підвищує обізнаність про важливі стандарти/ініціативи та стимулює зміни в поведінці організації – все це в дусі кращого обслуговування клієнтів.

Послуги таємних покупців потрібні людям з дуже різними ролями в компанії:

- власники бізнесу та топ-менеджмент. Для засновників компанії та представників топ-менеджменту особливо важливо бачити достовірну

інформацію про обслуговування компанії в цілому. Вони не мають можливості особисто приїжджати в кожну торгову точку/відділення, щоб особисто переконатися, що все працює саме так, як має бути;

- керівники відділу контролю якості/стандартів/внутрішнього аудиту. Керівники підрозділів, які відповідають за контроль якості, також рано чи пізно стикаються із необхідністю вибудовувати систему роботи з таємними покупцями. Таке завдання часто пов'язане з тим, що залучення підрядників для внутрішнього аудиту досить затратне, а аудити від керівників та менеджерів середньої ланки не завжди можуть бути об'єктивними та достовірними;

- фахівці з управління магазинами/супроводу роздрібної торгівлі. У великих торгових мережах створюються спеціальні відділи, які здійснюють супровід роботи роздрібних магазинів. Іноді в таких відділах вже збудовано систему проведення перевірок, найчастіше з використанням паперових носіїв. Робота таємних покупців може допомогти спеціалістам відділів управління магазинами скоротити грошові та тимчасові витрати на проведення аудитів, а також прискорити процес виявлення порушень;

- директори франчайзингових мереж. Перед директорами франчайзингових мереж стоїть завдання перевірки партнерів щодо дотримання стандартів і принципів роботи головної компанії. Таємні покупці можуть зробити це об'єктивно і з мінімальними витратами;

- виробники своєї продукції. Виробники можуть реалізовувати власні товари через партнерів: торговельні мережі, маркетплейси та ін. Щоб перевірити дотримання умов співробітництва (ціни, викладення товару, обслуговування тощо), можна проводити перевірки за допомогою таємних покупців [6].

Перевірки таємними покупцями можуть бути націлені на оцінку різних аспектів бізнесу:

1. Дотримання стандартів обслуговування клієнтів. Це найчастіше завдання, яке потрібно вирішити фахівцю з контролю якості та стандартів. Для цього потрібні об'єктивні дані про обслуговування клієнтів по всій роздрібній

мережі, які надходять безперервно, обробляються, формуються до звітів та завдань для усунення порушень.

У цій частині перевірки таємні покупці можуть відповідати на питання, що стосуються презентації товару/послуги та взаємодії з операторами, касирами, менеджерами з продажу та іншими працівниками.

2. Дотримання дрес-коду співробітниками. У компаніях із роздрібного бізнесу, сфери HoReCa, сфери послуг часто регламентовані вимоги до зовнішнього вигляду працівників. Планові внутрішні перевірки не завжди можуть показати об'єктивну картину, а таємні покупці, які обізнані про еталонні стандарти, можуть достовірно показати, чи дотримуються співробітники норми дрес-коду.

Залежно від галузі у цій частині можуть бути важливими стан одягу, взуття, макіяжу, манікюру, зачісок, бейджів співробітників.

3. Стан прилеглої території, вхідної групи, торговельного приміщення тощо. Перевіряючий оцінює чистоту всіх приміщень: наявність та заповненість урн, наявність пилу, бруду або неприємних запахів на прилеглій території, у вхідній групі, у торговому залі, примірювальній, касовій зоні тощо.

4. Подання товарів та оформлення торгової зали. На цьому етапі можна оцінити світловий та температурний режими в торговому залі, музичне тло, доступність інформації про товари: цінників, бірок, інформаційних табличок тощо.

5. Обслуговування у касовій зоні. Важливо, щоб під час продажу продавці-касири також дотримувалися прописаного протоколу компанії: вітали, питали про наявність дисконтної картки, інформували про акційні пропозиції та умови повернення, надавали касовий чек тощо. Ці та багато інших моментів може охопити під час візиту таємний покупець [7].

Перевірки таємних покупців через паперові звіти, онлайн-опитування або таблиці, на відміну від сучасних онлайн-сервісів, не дозволяють ефективно працювати з результатами перевірок. Для проведення по-справжньому

результативних перевірок у бізнесі чудово підходить сервіс мобільного аудиту на основі електронних чек-листів.

1.2 Мобільний додаток для «Тайного покупця»

З мобільним додатком таємні покупці можуть здійснювати весь спектр робочих завдань за допомогою персональних мобільних пристроїв – планшетів та смартфонів Android та iOS. Мобільні рішення забезпечують таємним покупцям доступ в режимі реального часу з торгових точок до необхідних для роботи даних та інструментів. Це значно спрощує виконання завдань та забезпечує краще розуміння необхідних дій у точках продажу.

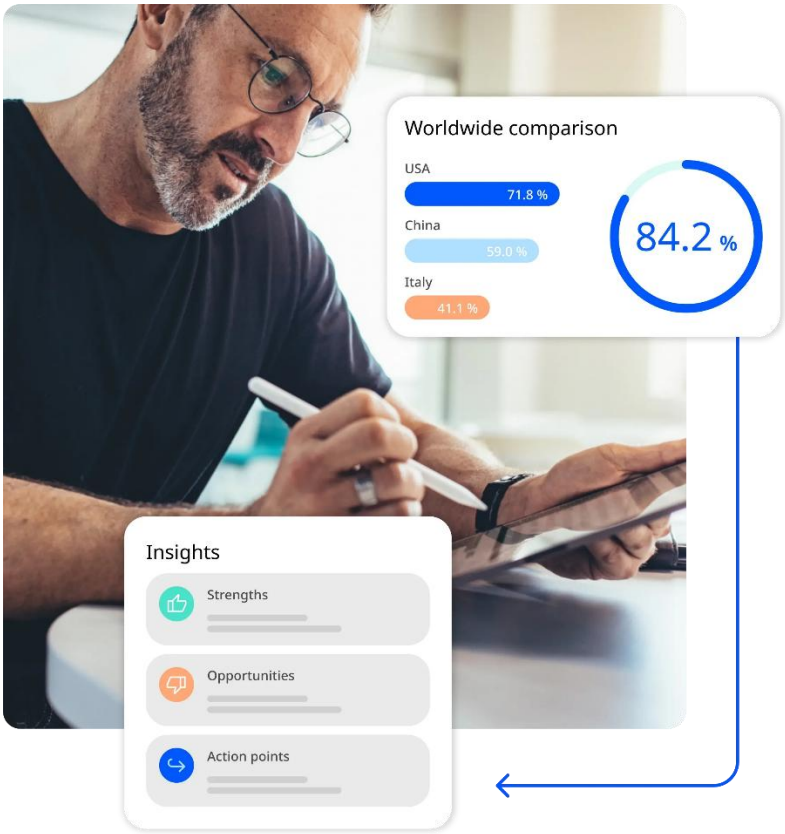

Рисунок 1.2 – Мобільний додаток для таємного покупця [8]

GPS технології забезпечують дистанційний контроль таємних покупців у режимі реального часу і дають можливість відстежувати переміщення таємних покупців, фіксувати час початку та закінчення робочого дня, час

проведений у точках продажу, витрачений на спілкування з персоналом та маршрут. Це дасть можливість бачите на карті всіх мобільних агентів будь-якої хвилини. Приємним бонусом може стати можливість оптимізації маршрутів таємних покупців, що скоротить пробіги автомобілів, зменшить витрати на паливо та заощадить час на переміщення від одного клієнта до іншого.

З мобільним додатком у таємних покупців буде більше часу для вирішення робочих питань. Більше того, електронний табель обліку робочого часу та переробок тепер ведеться автоматично. Зарплату можна розрахувати дуже швидко та дуже точно. Таким чином, мобільні технології дозволяють вирішити питання здійснення контролю за робочим часом таємних покупців.

Cloud-технології дозволяють значно знизити витрати на обладнання та посилити зв'язок офісу та польових команд. Усі документи, що заповнюються таємними покупцями, автоматично потрапляють у базу даних і вимагають часу на обробку з боку офісних працівників. Таємні покупці, виконуючи свою роботу за допомогою смартфонів та планшетів, заповнюють різні форми та звіти. Вся інформація безперервним потоком надходить в офіс для контролю та аналізу та доступна в електронному вигляді з будь-якого комп'ютера з будь-якої точки світу в офісній частині програми [9].

Подібна програма для таємних покупців забезпечить максимізацію продуктивності візитів польових співробітників за допомогою забезпечення доступу до корпоративних даних та можливості виконання робочих завдань за допомогою персональних мобільних пристроїв.

З подібною програмою для таємних покупців буде можливість керувати торговими агентами онлайн, так як дасть можливість не тільки бачити не тільки де знаходяться Ваші мобільні співробітники, але і що вже зроблено і що ще потрібно зробити. Будь-якої хвилини можна вносити зміни до їх роботи, додавати або коригувати завдання. Таким чином, Це дасть можливість легко контролювати діяльність таємних покупців і як результат отримати повну прозорість роботи співробітників та впевненість у якості та достовірності

отриманих даних.

Для створення подібного додатку треба:

1) визначити цілі і вимоги. Визначити, які завдання має вирішувати програма: збір оцінок, фото/відео фіксація, заповнення звітів, а також які компанії чи сфери послуг ви хочете охопити.

2) Проаналізувати, який функціонал необхідний для «таємного покупця» (наприклад, формування маршрутів, введення даних щодо кожного етапу покупки, можливість залишати коментарі та фото).

3) Продумати, як буде керуватися завданнями та звітністю для адміністратора програми.

4) Розробити дизайн та інтерфейс. Створити інтуїтивно зрозумілий і простий використання інтерфейс (UI/UX), щоб «таємному покупцю» було легко виконувати завдання.

5) Розробити дизайн, який буде відповідати вашим брендовим цілям та цілям користувачів.

6) Розробити функціонал додатку.

Для «таємного покупця»:

- можливість швидко та зручно заповнювати чек-листи, оціночні форми;
- додавання фото/відео матеріалів до звітів для підтвердження;
- формування та відображення маршрутів та завдань у додатку;
- можливість обміну повідомленнями з адміністратором.

Для адміністратора:

- управління користувачами: створення та управління обліковими записами «таємних покупців»;
- управління завданнями: створення, призначення та відстеження виконання завдань;
- звітність: перегляд та аналіз даних зі звітів, формування аналітичних звітів.

7) Тестування та запуск. Проведіть тестування програми на різних пристроях та операційних системах, щоб переконатися у його працездатності.

8) Зібрати відгуки користувачів та внести необхідні редагування.

1.3 Ананаліз подібних проектів

Perfect Store - інструмент збору польових даних, контролю викладки в магазинах та оцінки якості роботи польового персоналу. Скріншоти мобільного додатка представлені на рис. 1.3. Дані про додаток подано в таблиці 1.1.

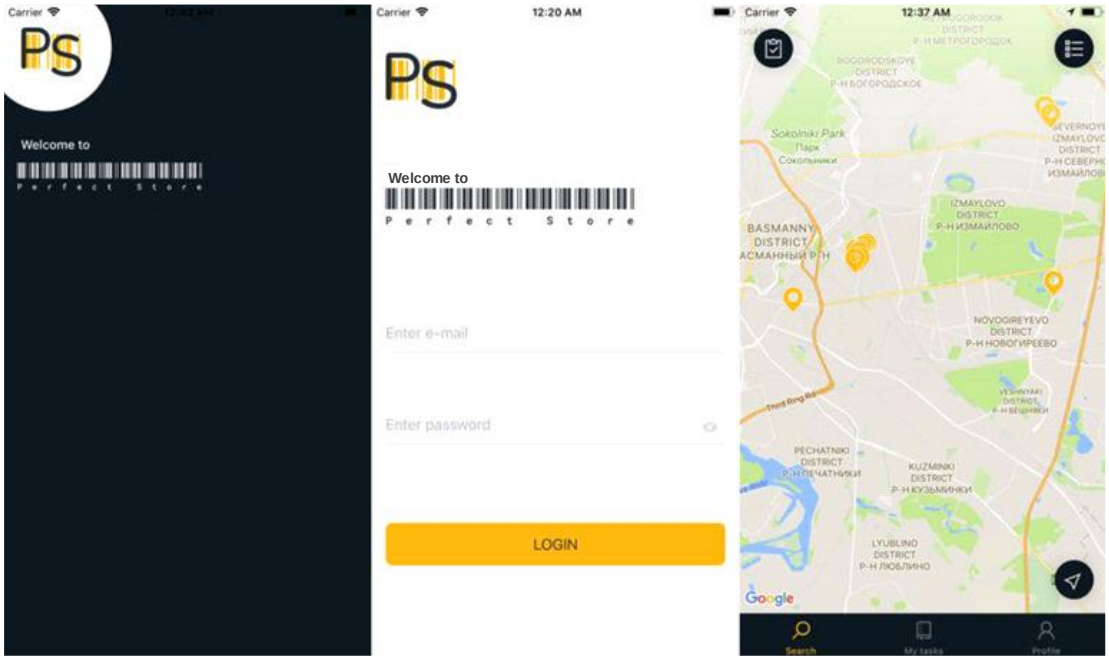

Рисунок 1.3 - Скріншоти мобільного додатка Perfect Store [10]

Таблиця 1.1 - Perfect Store

Рейтинг у Google Play	3/5 (на 19.04.2026)
Завдання на карті	Так
Розділ фотозвітів	Так
Сканування штрих-коду	Так
Розділ звіту за цінами	Нет
Розділ повідомлень	Нет

Bino CX – мобільний додаток для виконання завдань таємного покупця. Скріншоти мобільного додатка представлені на рисунку 1.4. Дані про додаток подано в таблиці 1.2.

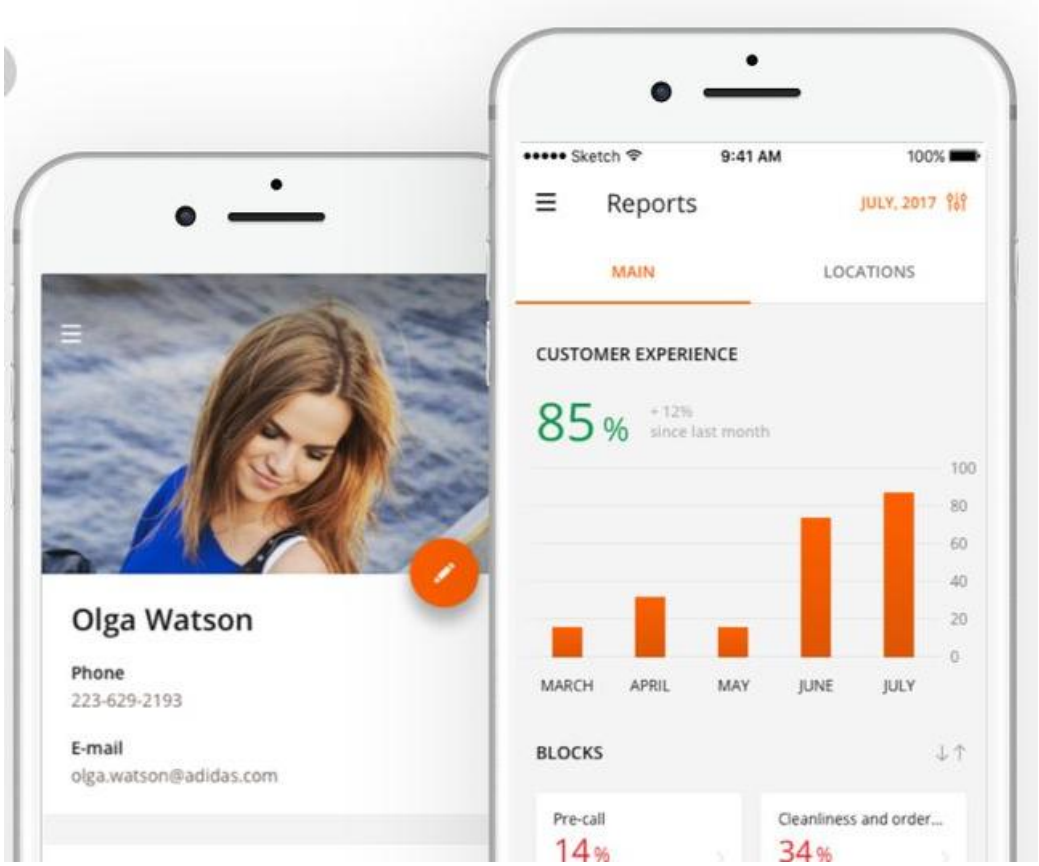

Рисунок 1.4 - Скріншоти мобільного додатка Bino CX [11]

Таблиця 1.2 - Bino

Рейтинг у AppStore	4.1/5 (на 19.04.2026)
Рейтинг у Google Play	3/5 (на 19.04.2026)
Завдання на карті	Так
Розділ фотозвітів	Так
Сканування штрих-коду	Нет
Розділ повідомлень	Нет
Розділ планограм	Нет
Розділ звіту за цінами	Нет

MD Audit - додаток для аудиту магазинів таємним покупцем, помічник аудитора торгових точок. Скріншоти мобільного додатка представлені рисунку 1.5. Дані про додатку представлені у таблиці 1.3.

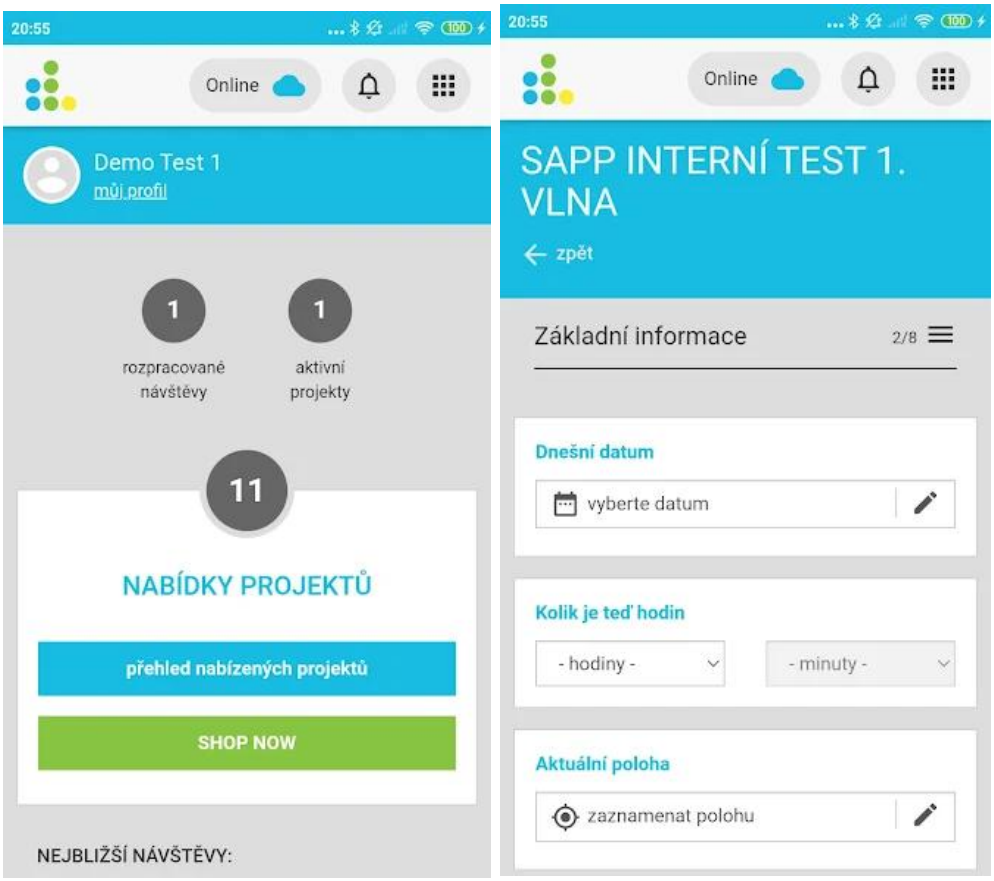

Рисунок 1.5 - Скріншоти мобільного додатка MD Audit [12]

Таблиця 1.3 - MD Audit

Рейтинг у AppStore	4.8/5 (на 20.04.2026)
Рейтинг у Google Play	3,3/5 (на 20.04.2026)
Завдання на карті	Нет
Розділ фотозвітів	Так
Сканування штрих-коду	Нет
Розділ повідомлень	Нет
Розділ планограм	Нет
Розділ звіту за цінами	Нет

Мінусом всіх розглянутих подібних проектів є вузька функціональність. Усі вони покривають потребу у відслідковуванні та редагуванні завдань, а також потребу у виконанні фотозвітів, але при цьому у всіх розглянутих проектах відсутні розділ планограм, звітів за встановленими цінами та внутрішніх повідомлень.

Висновки за розділом

У першому розділі було проаналізовано предметну область застосування методу «таємний покупець» та обґрунтовано доцільність використання мобільних технологій для автоматизації цього процесу. Розглянуто існуючі підходи до організації перевірок і визначено їхні обмеження, зокрема низьку ефективність паперових та частково цифрових рішень. Проведений аналіз аналогічних мобільних додатків показав, що більшість із них мають обмежений функціонал і не забезпечують повного покриття бізнес-потреб. Це дозволило сформувати вимоги до розроблюваної системи та визначити напрям її вдосконалення.

РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз вимог та опис функціонального призначення системи

2.1.1 Функціональні вимоги

У ході проектування мобільного додатка та аналізу технічного завдання замовника було визначено такі функціональні можливості, що надаються гостеві та користувачу.

1. Мобільний додаток повинен надавати відвідувачам можливість авторизуватися.
2. Мобільний додаток повинен надавати користувачеві можливість переглядати список актуальних завдань.
3. Мобільний додаток повинен надавати користувачеві можливість переглядати завдання на карті.
4. Мобільний додаток повинен надавати користувачеві можливість фільтрувати завдання.
5. Мобільний додаток повинен надавати користувачеві можливість завантажувати на сервер дані виконаного завдання.
6. Мобільний додаток повинен надавати користувачеві можливість виконання фотозвітів.
7. Мобільний додаток повинен надавати користувачеві можливість виконання звітів за встановленими цінами на товари та послуги, що надаються.
8. Мобільний додаток повинен надавати користувачеві можливість зчитування штрих-коду товару для швидкого пошуку та додавання.
9. Мобільний додаток повинен надавати користувачеві можливість перегляду списку внутрішніх повідомлень.
10. Мобільний додаток повинен надавати користувачеві можливість звернення на технічну підтримку.
11. Мобільний додаток повинен надавати користувачеві можливість

перегляду планограм.

12. Мобільний додаток повинен надавати користувачеві можливість встановлення фотографії профілю.

13. Мобільний додаток повинен надавати користувачеві можливість виходу з облікового запису [13].

2.1.2 Нефункціональні вимоги

Було визначено нефункціональні вимоги до програми, що розробляється.

1. Мобільний додаток має бути написаний мовою Swift [14].

2. Мобільний додаток має функціонувати на пристрої iPhone п'ятого поколінь (iPhone 4S) і вище.

3. Мобільний додаток має функціонувати на ОС iOS версії не нижче 9.0.

4. Мобільний додаток повинен бути доступним лише в портретній орієнтації.

5. Мобільний додаток повинен повністю відповідати вимогам та дизайну, затвердженим у ТЗ.

2.2 Варіанти використання системи

Для проектування мобільного додатка була використана мова графічного опису для об'єктного моделювання UML [15]. Було побудовано модель взаємодії зовнішнього актора з мобільним додатком у вигляді варіантів використання.

У ході проектування було виділено два актори:

Гість - людина, якій доступна можливість входу в профіль за допомогою смс-авторизації.

Користувач – авторизований гість, якому доступна можливість використання всіх функцій мобільного додатка.

Діаграма варіантів використання мобільного додатка передставлено на рис. 2.1.

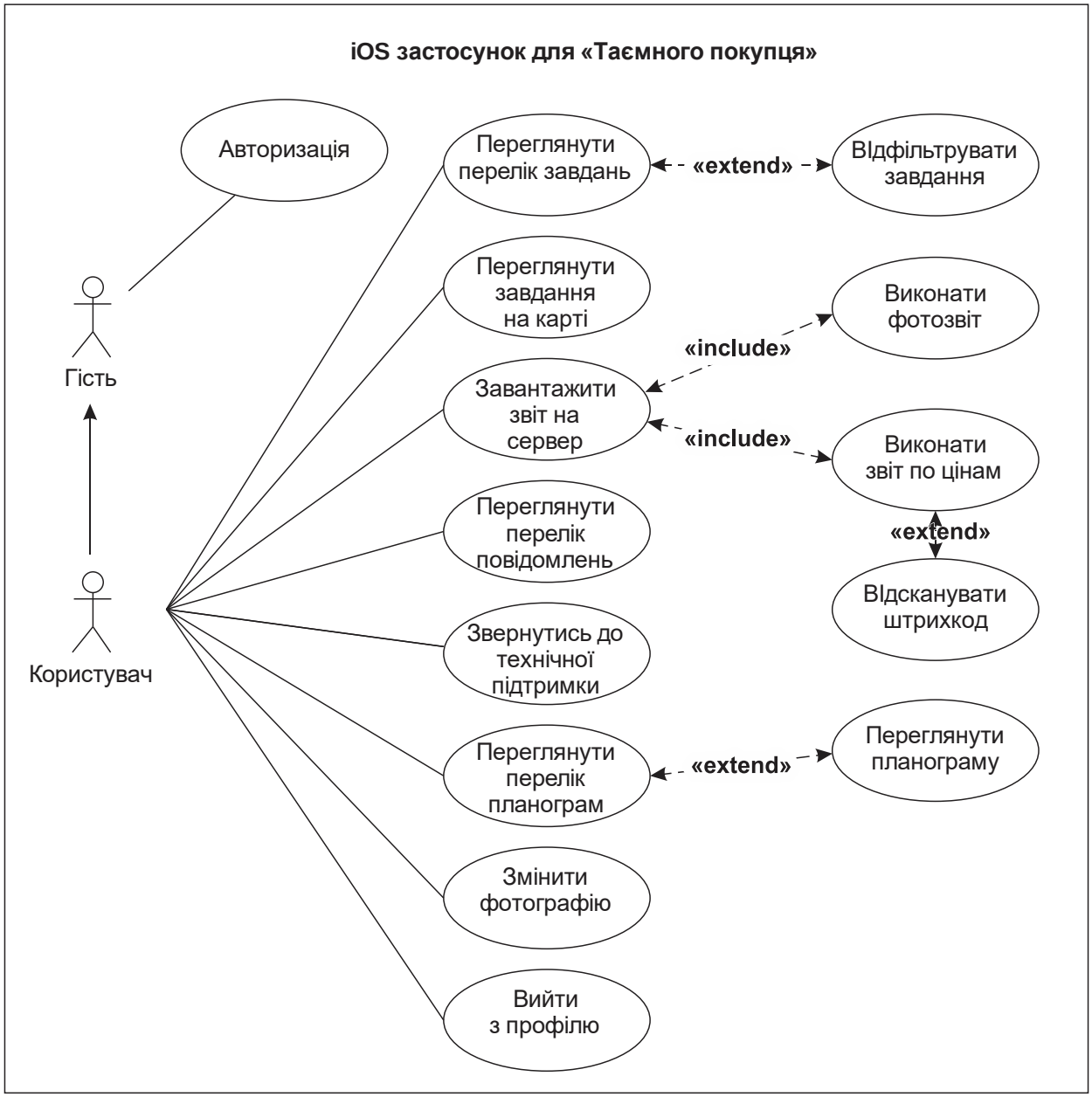

Рисунок 2.1 - Діаграма варіантів використання

Користувач може реалізувати такі варіанти використання мобільного додатка для «таємного покупця».

1. Поглянути завдання на карті – переглянути всі актуальні завдання на карті.

2. Поглянути список завдань – переглянути всі актуальні завдання у вигляді списку.

3. Відфільтрувати завдання – фільтрація списку завдань за назвою та статусом (завершена чи незавершена).

4. Завантажити звіт на сервер – надсилання на сервер даних виконаного завдання, що включає фотозвіт та звіт за цінами.

5. Виконати фотозвіт – зробити фотографії за вказаною тематикою та завантажити їх на сервер.

6. Виконати звіт за цінами – зафіксувати встановлені ціни на товари та послуги у підконтрольній точці.

7. Протсканувати штрих-код – рахувати штрих-код товару або послуги для швидкого додавання до звіту.

8. Пробратитися у технічну підтримку – заповнити форму зворотний зв'язок із зазначенням відповідної інформації про проблему.

9. Поглянути список повідомлень – перегляд даних, записаних на пристрій у момент отримання повідомлень.

10. Поглянути список планограм – перегляд усіх доступних користувачеві планограм.

11. Поглянути планограму – перегляд схеми викладання різних товарів на стелажах та вітринах.

12. Ізмінити фотографію – завантажити нову фотографію профілю на сервер.

13. Вийти з профілю – повернутися на екран авторизації зі зниженням прав користувача до прав гостя.

2.3 Компоненти системи

На рис. 2.2 представлена діаграма компонентів мобільного додатка на платформі iOS [16] для «таємного покупця». У процесі проектування програми використовувався архітектурний патерн MVC. Система складається з

наступних компонентів: модель, уявлення, контролер. Основною перевагою даного патерну є слабка зв'язаність між компонентами.

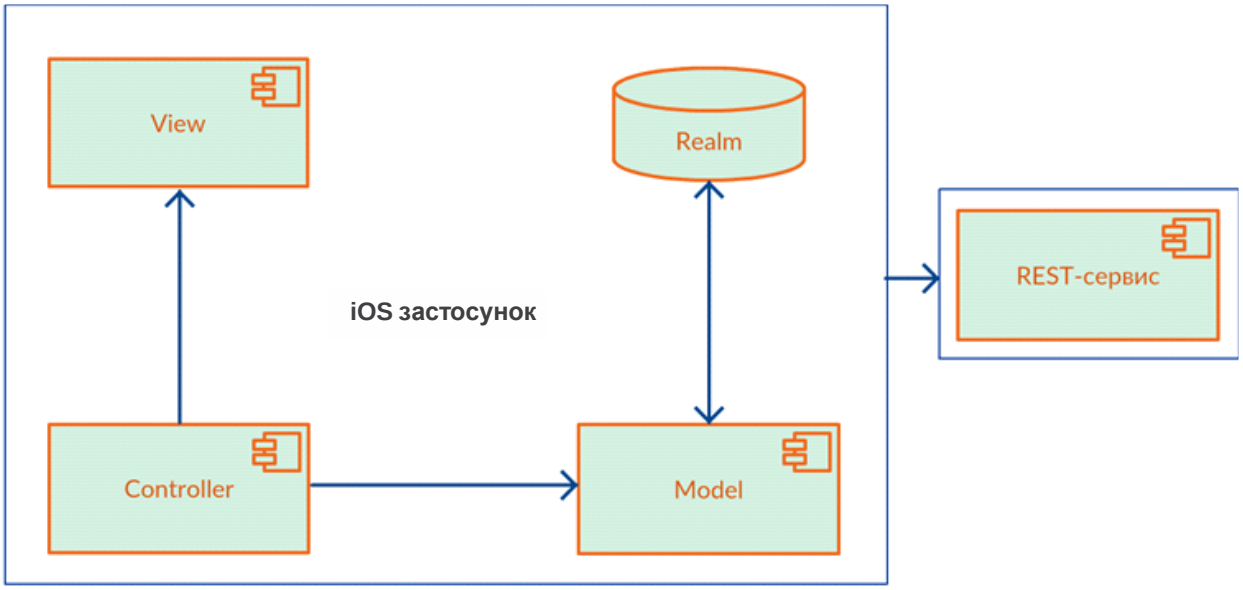

Рисунок 2.2 - Діаграма компонентів системи

Model – компонент програми, що відповідає за формування структури даних усередині програми та за способи маніпуляції над цими даними. Записує дані у внутрішню базу даних (Realm), де вони зберігаються та оновлюються протягом усього життя програми.

View – Подання мобільного додатка. Те, що бачить та з чим взаємодіє користувач: кнопки, поля введення, текст, зображення тощо.

Controller – обробник дій користувача, здійснює взаємодію моделі програми та її подання. Конфігурує подання, використовуючи дані з моделі. Передає дані, введені користувачем в модель, а також обробляє їх для відображення в поданні.

Основні контролери мобільного додатка, є.

1. LoginVC – контролер, який відповідає за процес авторизації користувача.

2. TasksVC – контролер, який відображає завдання користувача у вигляді списку.

3. MapVC – контролер, який відображає завдання користувача на карті.
4. SupportVC – контролер звернення до технічної підтримки.
5. SettingsVC – контролер налаштувань користувача.
6. NotificationsVC – контролер, що відображає список внутрішніх повідомлень.
7. PlanogramVC – контролер, який відповідає за відображення планограми.
8. PhotoReportVC – контролер, що відображає вибраний фотозвіт.
9. PriceReportVC – контролер, що відображає звіт щодо встановлення цінам.

REST-сервіс – зовнішня база даних мобільного додатку для «таємного покупця», взаємодія з якою здійснюється за допомогою REST-запитів [17].

Realm – внутрішня база даних мобільного додатка.

2.4 Діаграми послідовності

На рис. 2.3 представлена діаграма послідовності, на якій відображено процес авторизації користувача в системі для «таємного покупця», що включає такі кроки:

- користувач вводить телефонний номер у відповідне поле;
- додаток надсилає запит на сервер для надсилання СМС на введений номер телефону;
- сервер перевірять телефонний номер на наявність у базі даних користувачів;
- якщо телефонний номер є у базі даних користувачів, то сервер надсилає запит на СМС-сервіс для надсилання коду авторизації на вказаний номер;
- СМС-Сервіс відправляє користувачеві код авторизації на введений номер;
- користувач вводить отриманий у СМС код авторизації;
- програма запитує у сервера перевірку коду авторизації;

- сервер запитує у СМС-сервісу перевірку існування пари із введеного номера телефону та введеного коду авторизації;

- СМС-сервіс повідомляє серверу про існування перевіреної пари;

- сервер повідомляє додатку про існування пари та токен авторизації при існуванні пари;

- якщо пара, що перевіряється, існує, то додаток повідомляє користувачеві про успішну авторизацію і записує токен авторизації у внутрішню базу даних;

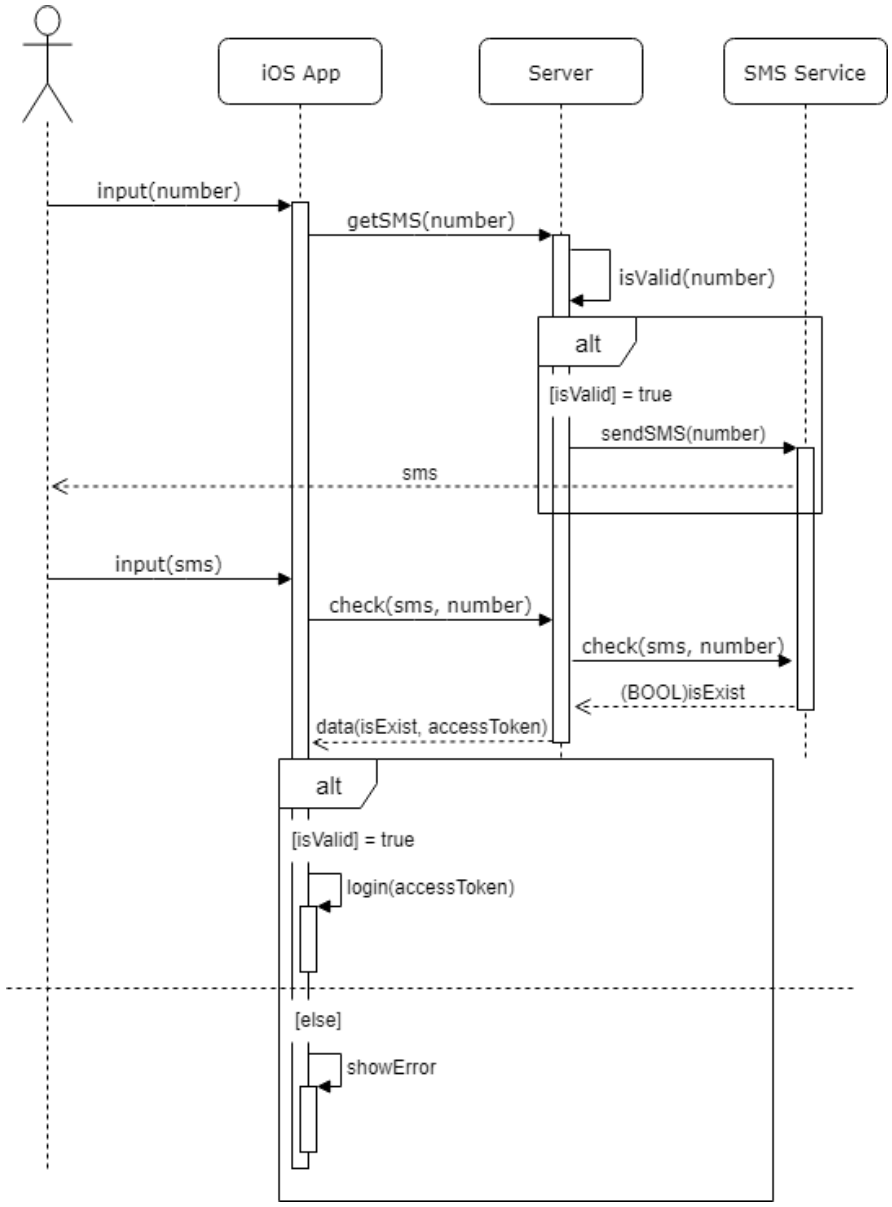

Рисунок 2.3 - Процес авторизації користувача у системі

- якщо пари, що перевіряється, не існує, то додаток повідомляє користувачеві про помилку при авторизації.

На рис. 2.4 представлена діаграма послідовності.

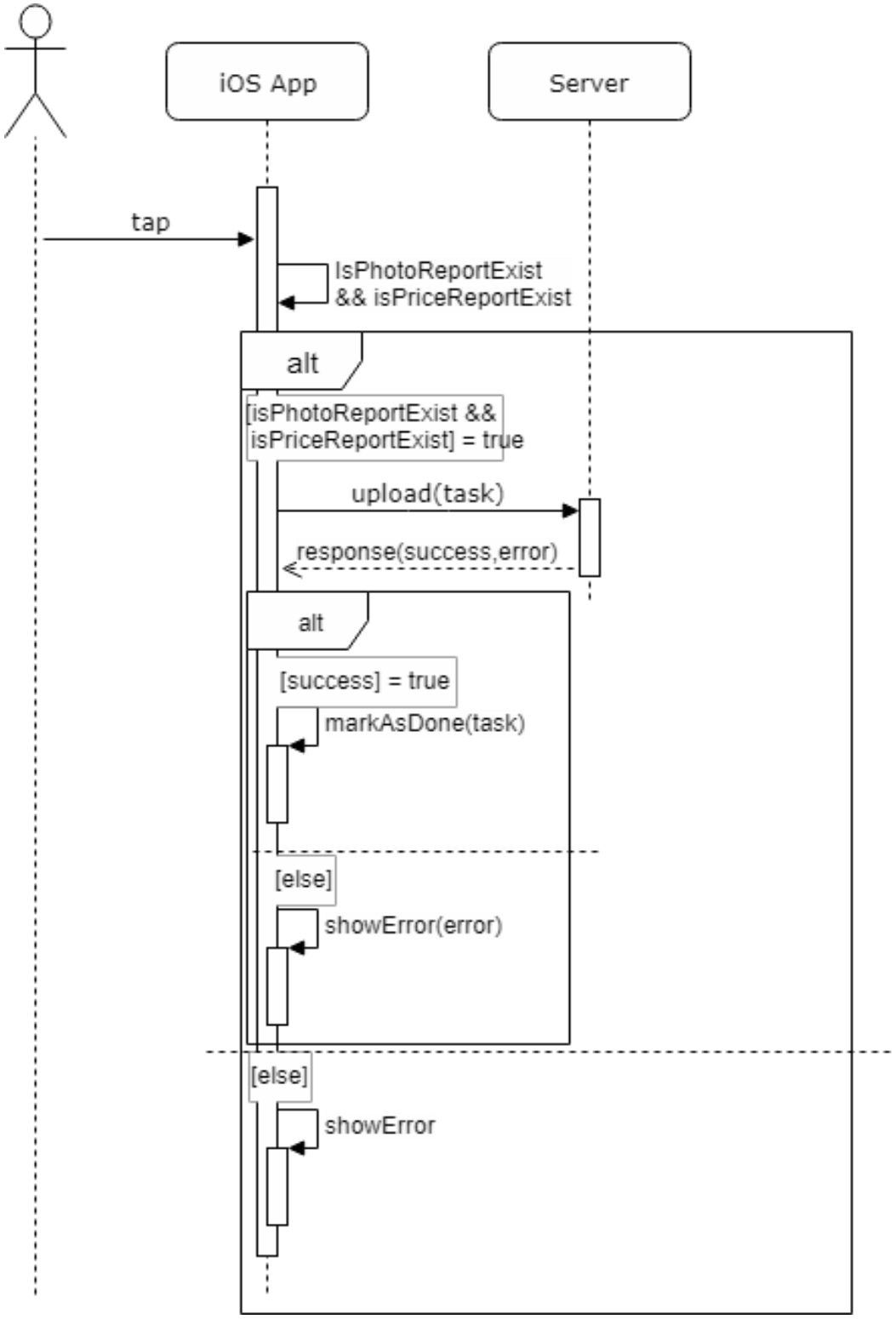

Рисунок 2.4 - Процес надсилання даних завдання на сервер

На діаграмі відображено процес завантаження звіту на сервер, що включає такі кроки:

1) користувач натискає кнопку завантаження завдання на сервер;

2) додаток перевіряє, чи заповнено фотозвіт та звіт за цінами для обраного завдання;

3) якщо фотозвіт або звіт за цінами не заповнено, додаток повідомляє користувача про помилку;

4) якщо і фотозвіт, і звіт про ціни були заповнені, то додаток завантажує на сервер дані заповненого завдання;

5) після завершення завантаження завдання сервер надсилає додатку відповідь про статус завантаження завдання;

6) якщо статус завантаження невдалий, програма повідомляє користувача про помилку, яка також знаходиться у відповіді з сервера;

7) якщо статус завантаження вдалий, програма позначає завдання як завершене.

2.5 Ананаліз існуючих рішень для реалізації додатку

На сьогодні існує кілька рішень для створення мобільних додатків.

Нативна розробка – розробка, за якої програмісти використовують інтегроване середовище розробки, що надається компанією-розробником платформи.

Кросплатформова розробка – розробка, при якій програмісти використовують спеціальні комплекси програмних засобів, завдяки яким можна створювати продукти відразу для кількох операційних систем.

До інтегрованих середовищ розробки для платформи iOS можна віднести IDE Xcode [18], розроблену компанією Apple, яка включає документацію та весь необхідний функціонал для розробки.

iOS SDK [19] – комплекс засобів розробки додатків під операційну систему iOS.

Interface Builder [20] – додаток для створення графічних інтерфейсів та створення зв'язків між подіями інтерфейсу та кодом програми.

iPhone Simulator – програма для симуляції запуску мобільних програм на реальному пристрої.

Природно з основних складових iOS SDK є фреймворк Cocoa Touch [21]. Цей фреймворк заснований на шаблоні проектування Model – View – Controller і включає наступні технології: CoreAnimation, мультизадачність та розпізнавання жестів. До складу Cocoa Touch входить безліч ключових бібліотек, які використовуються для створення мобільних додатків на платформі iOS, а саме:

1) UIKit Framework [22] – бібліотека, що надає класи, орієнтовані на графічні можливості користувальницького інтерфейсу;

2) Foundation Kit Framework [23] – бібліотека, що надає базові класи, що реалізують: рядки, колекції (масиви, набори, словари) та лічильники, файловий менеджер, архіватор об'єктів, обробник URL, повідомлення, багатопоточність, таймери, обробка винятків;

3) MapKit Framework – бібліотека, що здійснює взаємодію з картами та навігаційними можливостями iOS-пристроїв;

4) GameKit Framework – бібліотека для взаємодії із сервісом Game Center;

5) iAd Framework – бібліотека для реалізації сервісів контекстної реклами iAd усередині додатків.

Серед рішень кросплатформерної розробки на сьогоднішній день одним з найпопулярніших розробників як для розробки під iOS, так і під Android є фреймворк Xamarin [24].

Переваги Xamarin:

1) вможливість створювати програми відразу на кілька операційних систем;

2) для створення додатків необхідно знати лише C#.

Мінуси Xamarin:

1) швидкість роботи компілятора поступається швидкості компілятора в Xcode;

2) у безкоштовній версії обмежена розмірність проекту, при тому що інтегроване середовище розробки Xcode є безкоштовним;

3) рекі оновлення, в той час як Xcode оновлюється в момент виходу нової версії iOS. Немає можливості випускати швидкі оновлення з оптимізацією під нові версії ОС;

4) через кросплатформерну розробку ускладнюється процес верстки інтерфейсів, так як при збільшенні числа підтримуваних пристроїв збільшується кількість можливих розмірів екрану.

Для реалізації бази даних усередині мобільного додатка існує три найбільш популярні рішення.

Core Data [25] – нативна бібліотека від Apple для роботи з внутрішньою базою даних.

Realm [26] – мобільна база даних для iOS з підтримкою кроссплатформерності.

Рішення, засновані на зберігання даних у SQLite [27].

На рис. 2.5 представлений графік порівняння рішень для реалізації внутрішньої бази даних мобільного додатка на основі кількості опрацьованих запитів на секунду.

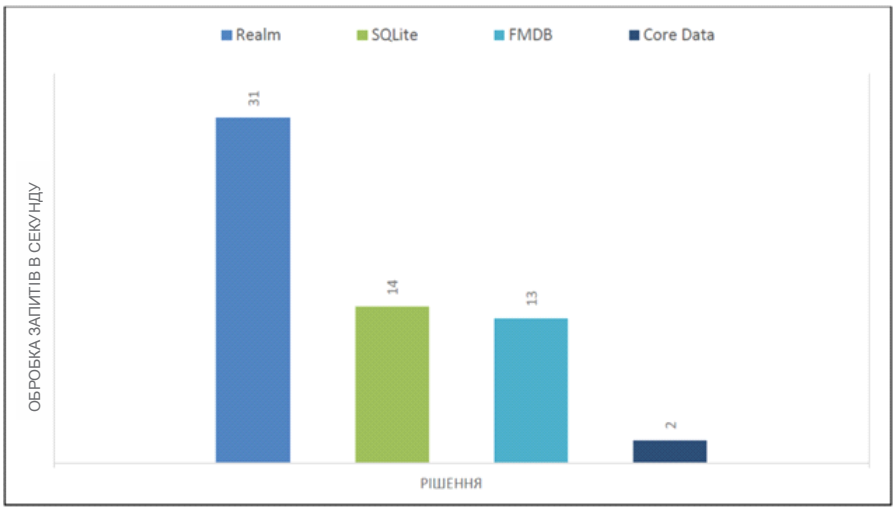

Рисунок 2.5 - Графік порівняння кількості оброблюваних запитів за секунду [28]

З представлених рішень для внутрішньої бази даних був обраний фреймворк Realm, так як у порівнянні з іншими рішеннями він швидше, простіше в установці та роботі і добре масштабуємо.

Висновки до другого розділу

У другому розділі виконано проектування мобільного додатка та серверної частини системи. Визначено функціональні та нефункціональні вимоги, що формалізують поведінку системи та обмеження її реалізації. Побудовано UML-моделі, зокрема діаграми варіантів використання, компонентів і послідовностей, що дозволило описати логіку взаємодії користувачів із системою. Обґрунтовано вибір архітектурного підходу MVC та технологічного стеку, включаючи використання REST-сервісів і бази даних Realm. У результаті сформовано цілісну архітектуру, придатну до подальшої програмної реалізації.

РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Моделювання взаємодії програмних компонентів

У мобільному додатку, що розробляється, було реалізовано 12 моделей, для кожної з них був створений клас, успадкований від базового класу RealmSwiftObject. Моделі містять інформацію про користувача, завдання, повідомлення, планограми, фотозвіти та звіти за встановленими цінами.

Схема реалізованих моделей у розробленому мобільному додатку представлена на рис. 3.1. Детальний опис усіх атрибутів, сутностей та зв'язків представлений у додатку А.

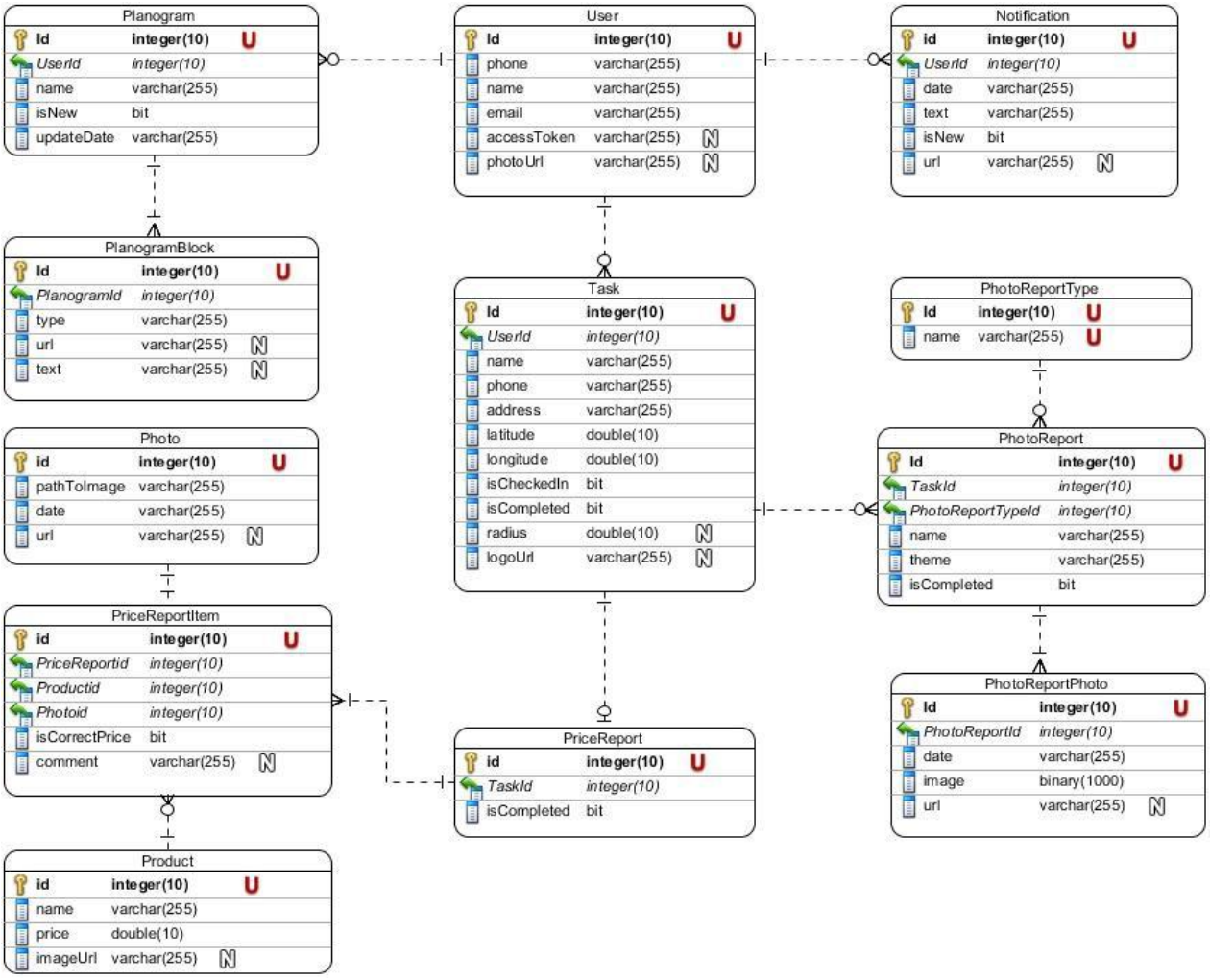

Рисунок 3.1 - Схема бази даних застосунку

На на риа. 3.2 представлений лістинг класу, що реалізує модель продукту. Product успадковується від базового класу Realm – Object. У моделі *Product* зберігаються атрибути id, name, imageUrl, price.

За допомогою методу primaryKey() задається первинний ключ. Також у лістингу представлений допоміжний ініціалізатор, який приймає на вхід словник dictionary, усередині якого зберігаються атрибути продукту, отримані з сервера.

```
class Product: Object {

    @objc dynamic var id = 0

    @objc dynamic var name: String?

    @objc dynamic var image: String?

    @objc dynamic var price: Double = 0.0

    @objc dynamic var volume: String?

    override static func primaryKey() -> String? {
        return "id"

    }

    convenience init(dictionary: [String: Any?]) {
        self.init()

        name = dictionary["name"] as? String

        if let idValue = dictionary["id"] as? Int {
            id = idValue

        }

        if let priceValue = dictionary["price"] as? Double {
            price = priceValue

        }

        if let urlString = dictionary["image_url"] as? String {
            if urlString.range(of: "https") == nil {

                image = API.photosURL + urlString

            } else {

                image = urlString

            }

        }

    }


}
```

Рисунок 3.2 - Лістинг класу, що реалізує модель продукту

На рис. 3.3 представлений лістинг видалення старих фотографій із внутрішньої бази даних Realm. Метод clearOldPhotoCache() – це метод класу, що видаляє сутність моделі Photo, з моменту створення яких минула доба. За допомогою об'єкта Realm() здійснюється вибірка фотографій відфільтрованих за предикатом, потім відбувається видалення об'єктів, що потрапили у вибірку. Метод повертає кортеж (tuple), що з двох параметрів: логічного параметра success і опціональної помилки error.

```
   class func clearOldPhotoCache() -> (success: Bool, error: NSError?)
      { do
            {
   NSDate)
let realm = try Realm()
let dayAgo = Date(timeInterval: -twentyFourHours, since: Date()) let
predicate = NSPredicate(format: "date < %@", dayAgo as let photos =
realm.objects(Photo.self).filter(predicate)
           try realm.write {
               for photo in photos {
                   if CameraManager.sharedInstance.deleteFileWithRelative-
   Path(photo.pathToImage!).success {
                       realm.delete(photo)
                   }
               }
           }
       } catch {
           return (false, Errors.databaseDeleteError)
       }
       return (true, nil)
   }
```

Рисунок 3.3 - Лістинг видалення старих фотографій із внутрішньої бази даних

3.2 Реалізація контролерів

У контролерах міститься логіка реакції мобільного додатка на дії користувача: обробка даних, що наводяться, натискань і різних жестів.

Також контролер формує подання, використовуючи дані, отримані з моделі, і записує в модель дані, введені користувачем у поданні. Таким чином, контролер пов'язує модель та подання.

Контролер PhotoReportVC відповідає за екран відображення фотозвіту у вигляді списку фотографій з інформацією про розмір файлів. Для конфігурації таблиці контролер PhotoReportVC обробляє делегати [19] UITableViewDelegate та UITableViewDataSource.

На рис. 3.4 представлений лістинг обробки делегатів UITableViewDelegate і UITableViewDataSource.

```
    func tableView(_ tableView: UITableView, numberOfRowsInSection section:
    Int) -> Int {

        return photosArray.count

    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: Index-
    Path) -> UITableViewCell {

        let cell = tableView.dequeueReusableCell(withIdentifier:
    PhotoReportPhotosListCell.reuseIdentifier, for: indexPath)

        if let photoCell = cell as? PhotoReportPhotosListCell {
            photoCell.configure(withPhotoReportPhoto: photosArray[index-

    Path.row], swipeCallback: { _ -> Bool in

                guard let idx = tableView.indexPath(for: photoCell) else {
    return false }

                let photoToRemove = self.photosArray.remove(at: idx.row)
                PhotoReportManager.remove(photos: [photoToRemove])

                tableView.beginUpdates()
                tableView.deleteRows(at: [idx], with:
                .automatic) tableView.endUpdates()

                if self.photosArray.isEmpty {
                    self.sendReportButton.isHidden =
                    true

                }

                return true

            })

        }

        return cell

}
```

Рисунок 3.4 - Лістинг обробки делегатів UITableViewDelegate та UITableViewDataSource

Також контролер PhotoReportVC містить логіку обробки видалення

фотографії з фотозвіту шляхом перетягування комірки таблиць у бік. Осередок таблиці PhotoReportPhotosListCell обробляє момент, в який користувач закінчує її перетягувати убік, після чого контролер обробляє подію swipeCallback та видаляє фотографію.

Контролер ScannerVC відповідає за екран сканування штрих-кодів товарів та послуг. Для розпізнавання штрих-коду контролер обробляє делегат AVMetadataMachineReadableCodeObject, листинг обробки цього делегує представлений рис. 3.5.

```
func metadataOutput(_ output: AVCaptureMetadataOutput, didOutput
metadataObjects: [AVMetadataObject], від connection: AVCaptureConnection) {

    if metadataObject.type == .ean13, let code = metadataOb-
ject.stringValue {

        self.captureSession?.stopRunning()
        TaskManager.sharedInstance.searchProduct(byBarcode: code, comple-

tion: { (success, product, err) in

            if success {

                guard let vc = self.storyboard?.instantiateViewController
(withIdentifier: "productInfoVC") as? ProductInfoVC else {

                    return

                }

                let fetchBlock = TaskManager.getPriceReportItemWith(prod-
uct: product!, task: self.task)

                vc.interactor.priceReportItem = fetchBlock.item

                vc.interactor.product=product

                self.setText(withType: .info)

            } else {

                self.setText(withType: .noProductError)
                self.captureSession?.startRunning()

                if let err = err, err == Errors.sendRequestError {
                    self.setText(withType: .noInternetError) return

                }

                Dispatch.doAfter(seconds: 2, {
                    self.setText(withType: .info)

                })

            }

        })

    }
```

Рисунок 3.5 - . Лістинг обробки делегату AVMetadataMachineReadable CodeObject.

3.3 Реалізація інтерфейсів

Для реалізації інтерфейсів у розробленому мобільному додатку використовувався інструмент InterfaceBuilder, розроблений компанією Apple. Він надає безліч елементів для побудови різних інтерфейсів: кнопки, лейбли, текстові поля, контейнери, таблиці та колекції даних.

Природною з особливостей InterfaceBuilder є вбудований Auto Layout [2], який динамічно прораховує розміри та розташування всіх об'єктів, спираючись на обмеження (constraints), задані розробником, рис. 3.6

Рисунок 3.6 – Застосунок для реалізації інтерфейсів Interface Builder [29]

У процесі розробки мобільного додатка було створено 6 файлів з розширенням *.storyboard, які представляють графічний інтерфейс екранів програми та зв'язку між ними.

Також у процесі розробки мобільного додатка було створено 12 файлів з розширенням *.xib, які містять графічний інтерфейс окремих екранів або частин екрана програми. Кожен екран у файлі *.*storyboard*і кожен файл із розширенням *.xib пов'язаний з єдиним класом.

Файли *.xib пов'язані з класами компонента view, а екрани *.*storyboard*пов'язані з класами компонента controller, рис. 3.7.

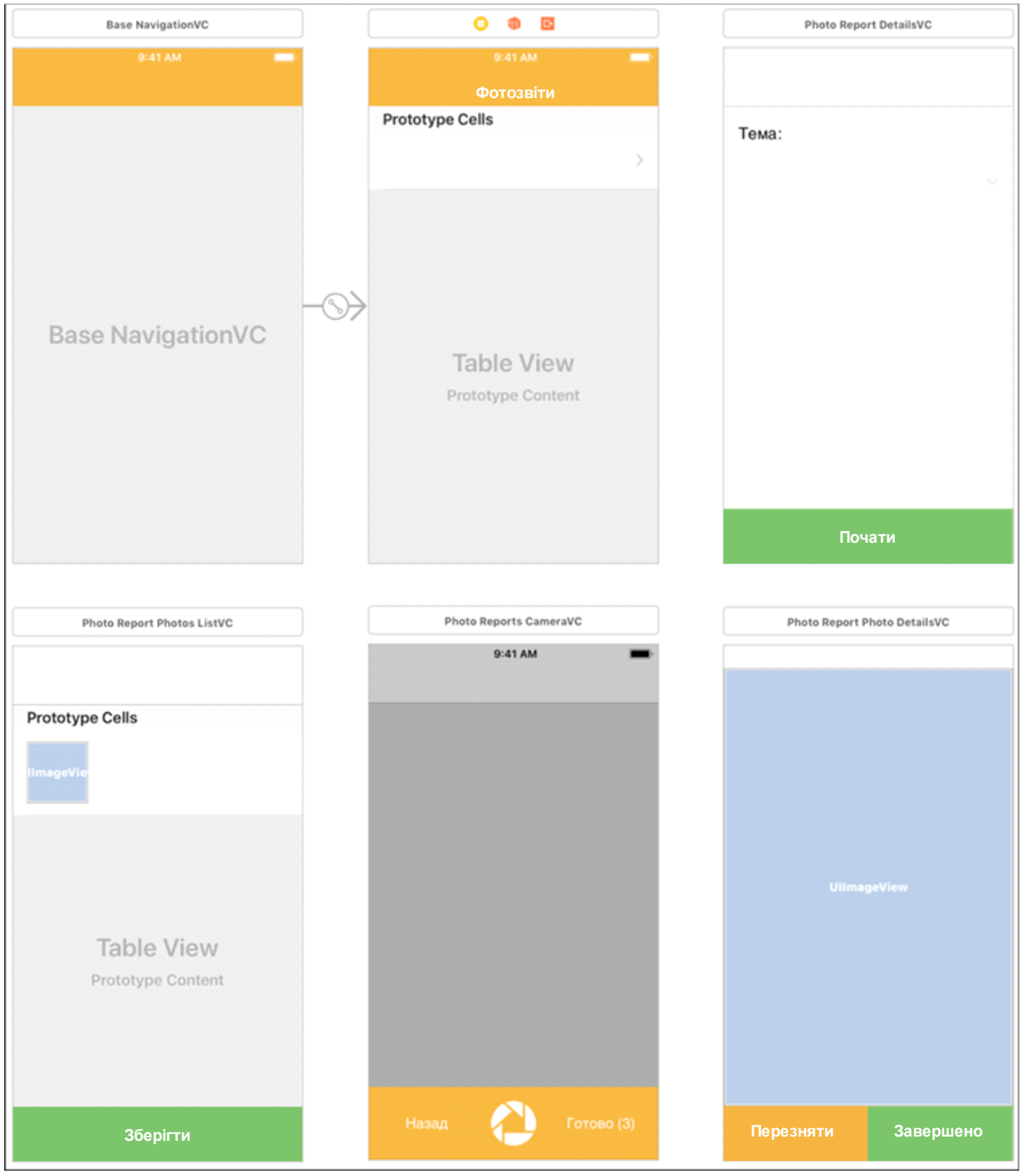

Рисунок 3.7 - Файл photoRepors.storyboard

3.4 Реалізація взаємодії з сервером

У розробленому додатку було реалізовано менеджер запитів

RequestManager, всередині якого інкапсульовано логіку взаємодії з REST-сервісом. RequestManager дозволяє здійснювати REST-запити до серверу та відповідає за розбір отриманих даних із формату JSON [30].

На рис. 3.8 представлений листинг логіки звернення до сервера за допомогою REST-запитів. За допомогою бібліотеки Alamofire відбувається створення та конфігурація запиту. Відповідь від сервера передається в completion блок методу send з логічним параметром success та результатом типу Any.

```
    class func send(path: String, methodType: HTTPMethod, params:
    Parameters?, completion: ((_ success: Bool, _ result: Any?) -> Void)?)
    {
        let fullString = API.serverURL + path
        if let url = URL(string: fullString) {
            Alamofire.request(url, method: methodType, parameters: params,
    encoding: methodType == .get ? URLEncoding.default : JSONEncoding.de-
    fault, headers: RequestManager.httpHeaders())
                .validate()
                .responseJSON { (dataResponse) in
                    guard let data = dataResponse.data else {
                        completion?(false, Errors.sendRequestError)
                        return
                    }
      do {
if let dictJson = try JSONSerialization.jsonObject(with: data, options: []) as?
[String: Any] {
                            let parsedResponse = RequestManager.parse(response:
                            dictJson)
                            parsedResponse.value)
                }
   completion?(parsedResponse.success,
                    } catch {
                        completion?(false, Errors.sendRequestError)
                    }
            }
        }
    }
```

Рисунок 3.8 - Лістинг логіки звернення до сервера за допомогою REST-запитів

На рис. 3.9 представлений листинг отримання списку завдань із сервера за допомогою GET-запиту.

```
func tasks(completion: ((_ tasks: [Task]?, _ error: NSError?) -> Void )?)
{
    RequestManager.send(path: API.Paths.tasks, methodType: .get, params:
nil) { (success, result) in
        if success {
            var tasksArray: [Task] = []
            if let taskDicts = result as? [[String: Any]] {
                for taskDict in taskDicts {
                    let task = Task(dictionary: taskDict)
                    tasksArray.append(task)
                }
            }
            Database.importObjects(tasksArray)completion?(tasksArray,
            nil)
        } else {
            completion?(nil, result as? NSError ??
Errors.unknownServerError)
        }
    }
```

Рисунок 3.9 - Лістинг отримання списку завдань

За допомогою методу send, описаного вище, відправляється GET-запит на сервер. При отриманні відповіді від сервера в completion блоці опрацьовується результат і відбуваються подальші маніпуляції над отриманими даними.

На рис. 3.10 представлений листинг надсилання даних виконаного звіту за допомогою запиту POST.

```
class func upload(task: Task, completion: ((_ success: Bool, _ error:
NSError? ) -> ())?) {
    RequestManager.send(path: API.Paths.uploadTask,
                        methodType: .post,
                       params: TaskManager.getReportParams(for: task))
    {(success, result) in
        completion?(success, success ? nil : result as? NSError ??
Errors.unknownServerError)
    }
```

Рисунок 3.10 - Лістинг надсилання даних звіту за встановленими цінами

У табл. 3.1 наведено короткий опис API серверної частини, що містить назву, тип, та опис REST-запитів.

Таблиця 3.1 - Короткий опис API серверної частини

Назва	Тип	Опис
checkIn	POST	Запит на авторизацію користувача
tasks	GET	Запит на отримання списку актуальних задач користувача
planograms	GET	Запит на отримання планограм користувателя
notifications	GET	Запит на отримання повідомлень користувача
priceReport	GET	Запит на отримання даних про звіт установленим цінам
photoReport	GET	Запит на отримання даних про фотозвіт
uploadTask	POST	Запит на завантаження даних про виконання звіту
productByBarcode	GET	Запит на отримання даних про товар або услузі за штрих-кодом
support	POST	Запит на надсилання форми зворотного зв'язку користувачем
updateAvatar	POST	Запит на оновлення фотографії профілю користувача
logout	POST	Запит на вихід із профілю

Для інтеграції сторонніх бібліотек використовується CocoaPods – засіб для керування залежностями бібліотек Cocoa. У проект інтегровані такі бібліотеки.

1. Realm – бібліотека для роботи із внутрішньою базою даних.
2. Alamofire – бібліотека для роботи з мережними http(s) запитами.
3. SDWebImage – бібліотека для асинхронного завантаження зображень із мережі та подальшого кешування.
4. GoogleMaps SDK – SDK Google для роботи з картами.
5. Crashlytics – бібліотека для збору збоїв та помилок мобільного додатка.

Для тестування розробленої мобільної програми було вибрано функціональне, інтеграційне та unit-тестування.

3.5 Функціональне тестування

Функціональне тестування - це тестування з метою перевірки програмного продукту на відповідність встановленим функціональним вимогам, тобто здатність програмного продукту в певних умовах вирішувати завдання, необхідні користувачам [31]. Функціональні вимоги визначають, що саме повинен виконувати програмний продукт і які завдання він має вирішувати. Набір тестів на функціональність представлений у табл. 3.2.

Таблиця 3.2 - Функціональні випробування

№	Назва	Кроки	Очікуваний результат	Пройден?
1	2	3	4	5
1	Авторизація в додатку	1. Запустити застосунок 2. Увести номер телефону 3. Ввести запропонований смс-код 4. Натиснути на кнопку входу	На екрані з'явиться карта з позначками актуальних завдань користувача.	Так
2	Перегляд зпису завдань	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку “Завдання”	На екрані з'явиться список задач користувача у вигляді таблиці	Так
3	Відфільтру-вати завдання	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку “Завдання” 3. Натиснути на кнопку "Фільтр" 4. Вибрати фільтр	На екрані з'являться тільки ті завдання, які підходять за параметром фільтра	Так
4	Завантажити звіт на сервер	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку надіслати звіт 3. Дочекатися завершення завантаження на сервер	Звіт позначиться як виконаний	Так

Продовження таблиці 3.2

1	2	3	4	5
5	Виконати фотозвіт	1. Вибрати актуальний звіт Натиснути на кнопку "Фотозвіти" 2. Вибрати тему фото- звіту 3. Натиснути кнопку “Почати” 4. Зробити необхідні фотографії 5. Натиснути на кнопку “Зберегти фотозвіт”	Фотозвіт позначиться як вдосконалий	Так
6	Виконати звіт щодо цін	1. Вибрати актуальний звіт 2. Натиснути кнопку “Звіт за цінами” 3. Заповнити всі картки товарів або послуг 4. Натиснути кнопку “Зберегти звіт”	Звіт за цінами позначиться як виконаний	Так
7	Відсканувати штрих-код	1. Перебуваючи у звіті з цінам натиснути на кнопку з зображенням штрих-коду 2. Навести на штрих- код товару чи послуги	На екрані з'явиться інформація про знайдений товар чи послуги	Так
8	Подивитись запис повідомлень	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку “Сповіщення”	Відкриється екран зі списком повідомлень користувача	Так
9	Подивитись завдання на карті	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку "Карта"	На екрані з'явиться карта з позначками актуальних завдань користувача.	Так
10	Подивитись зписок плано-грам	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку "Планограми"	Відкриється екран зі списком планограм користувачеві	Так
11	Подивитись планограму	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку "Планограми" 3. Вибрати будь-яку планограму	Відкриється екран з вибраною планограмою	Так
12	Змінити фотографію	1. Натиснути на кнопку бокового меню 2. Натиснути кнопку “Налаштування” 3. Натиснути на кнопку “Змінити фотографію” 4. Прикріпити фотографію 5. Натиснути кнопку “Завантажити”	Нове зображення завантажено на сервер і відобразиться в налаштуваннях	Так

Закінчення таблиці 3.2

1	2	3	4	5
13	Звернутися до технічної підтримки	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку "Допомога" 3. Обрати тему звернення 4. Заповнити обов'язкові поля 5. При необхідності прикріпити зображення 6. Натиснути на кнопку “Надіслати повідомлення”	Всі поля на екрані очистяться і з'явиться повідомлення про те, що звернення успішно надіслано	Так
14	Вийти з профіля	1. Натиснути на кнопку бокового меню 2. Натиснути на кнопку “Допомога” 3. Натиснути на кнопку "Вийти"	Відкриється екран авторизації	Так

3.5.1 Unit-тестування

Для автоматизації тестування та покриття частин розробленого мобільного додатка, відповідальних за взаємодію з сервером, розбір отриманих даних у моделі, та збереження моделей у внутрішню базу даних, були реалізовані Unit-тести з використанням інтегрованого в Xcode фреймворку XCTest [32].

Для тесту аналізу та запису даних у модель користувача було реалізовано наступний алгоритм:

1) створення тестових даних;

2) зопис об'єкта користувача з тестовими даними;

3) зопитування у внутрішню базу даних для отримання збереженого об'єкта;

4) перевірка даних на коректність.

На рис. 4.1 представлений листинг тесту аналізу та запису даних у модель користувача.

```
func testUserImport() {

    let userDict: [String: Any?] = ["id": 0,
                    "phone": "89221112233",
                    "name": "testUser",
                    "email": "test@mail.ru"]


    let realm=try! Realm()
    try! realm.write {
        realm.add(User(dictionary: userDict))
    }


    let users: [User] = realm.objects(User.self).filter("id == 0")


    XCTAssertTrue(users.first != nil)
    XCTAssertTrue(users.count == 1)
    XCTAssertTrue(users.first.name== "testUser")
    XCTAssertTrue(users.first.email== "test@mail.ru")
    XCTAssertTrue(users.first.phone == "89221112233")
```

Рисунок 4.1 - Лістинг тесту аналізу та запису даних у модель користувача

3.5.2 Інтеграційне тестування

Інтеграційне тестування – повна перевірка програмного продукту після його збирання з метою виявлення помилок, що виникають у процесі інтеграції програмних модулів або компонентів [33].

Для перевірки коректності відображення елементів інтерфейсу розроблений мобільний додаток був перевірений на розмірах екранів наступних iOS-пристроїв: iPhone 4S, iPhone 5, iPhone 13. В результаті перевірки коректності відображення елементів інтерфейсу помилок не було виявлено, всі екрани відповідні.

Крім того, розроблена мобільна програма була перевірена на пристроях з наступними версіями операційної системи: iOS 10, iOS 15, iOS 20. На всіх підтримуваних версіях операційної системи iOS розроблений мобільний додаток працює коректно і помилок не було виявлено.

Скріншоти основних екранів розробленого мобільного додатка представлені у додатку.

Висновок до третього розділу

У третьому розділі реалізовано програмну частину мобільного додатка відповідно до визначених вимог та спроєктованої архітектури. Розроблено моделі даних, контролери та інтерфейси користувача, забезпечено взаємодію з сервером через REST API. Інтегровано сторонні бібліотеки для роботи з мережею, базою даних і мультимедійними ресурсами.

Проведено тестування (функціональне, інтеграційне та unit-тестування), яке підтвердило коректність роботи системи та її відповідність поставленим вимогам. Отримані результати свідчать про працездатність і готовність програмного продукту до практичного використання.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ

4.1 Організаційно-правові основи забезпечення безпеки праці

Охорона праці в сучасних умовах є невід'ємною складовою організації трудової діяльності та одним із головних напрямків забезпечення безпеки працівників. Її основною метою є створення безпечних, комфортних і нешкідливих умов праці, які дозволяють зберегти життя, здоров'я та працездатність людини під час виконання професійних обов'язків. У сучасному суспільстві питання охорони праці мають особливе значення через активний розвиток інформаційних технологій, автоматизацію виробничих процесів та постійне зростання інтелектуального навантаження на працівників.

У сфері інформаційних технологій охорона праці відіграє важливу роль у забезпеченні безпечної роботи користувачів комп'ютерної техніки, програмного забезпечення та електронних засобів обробки інформації. Працівники ІТ-сфери значну частину робочого часу проводять за персональними комп'ютерами, що супроводжується тривалим статичним навантаженням, напруженням зору, психоемоційним перенапруженням та впливом інших шкідливих факторів виробничого середовища. Саме тому організація безпечного робочого місця та дотримання вимог охорони праці є необхідними умовами ефективної професійної діяльності.

Особливістю праці в ІТ-галузі є високий рівень інтелектуального навантаження та необхідність постійної концентрації уваги. Робота програмістів, розробників мобільних додатків, тестувальників та інших ІТ-спеціалістів пов'язана з аналізом великого обсягу інформації, виконанням складних логічних операцій та тривалою роботою за екраном монітора. За відсутності належної організації умов праці це може призводити до розвитку професійної втоми, зниження працездатності, погіршення стану здоров'я та виникнення професійних захворювань.

Важливе значення у сфері охорони праці мають організаційно-правові основи забезпечення безпеки працівників. В Україні питання охорони праці регулюються системою законодавчих та нормативно-правових актів, які визначають права та обов'язки працівників і роботодавців, а також встановлюють вимоги до створення безпечних умов праці.

Основним нормативно-правовим документом у даній сфері є Закон України «Про охорону праці», який визначає основні принципи державної політики у сфері безпеки праці, порядок організації охорони праці на підприємствах та вимоги щодо захисту працівників від небезпечних і шкідливих виробничих факторів [34]. Закон встановлює обов'язок роботодавця створювати належні умови праці, проводити інструктажі, навчання та забезпечувати працівників необхідними засобами безпеки.

Крім Закону України «Про охорону праці», важливу роль відіграє Кодекс законів про працю України, який регулює питання режиму праці та відпочинку, тривалості робочого часу, умов праці та соціального захисту працівників. Для працівників IT-сфери особливе значення мають державні санітарні норми та правила, що встановлюють вимоги до організації роботи з комп'ютерною технікою, рівня освітлення, параметрів мікроклімату та ергономіки робочого місця.

Важливим напрямком розвитку охорони праці є впровадження міжнародних стандартів у сфері безпеки праці. Використання сучасних підходів до управління професійними ризиками дозволяє своєчасно виявляти потенційні небезпеки та впроваджувати заходи щодо їх попередження.

4.2 Характеристика об'єкта та виявлення потенційних небезпек

Об'єктом проектування у розділі з охорони праці є автоматизоване робоче місце розробника мобільного додатку на платформі iOS. Дане робоче місце призначене для створення, тестування та супроводу програмного забезпечення для мобільних пристроїв компанії Apple. Основною функцією розробника є

написання програмного коду, аналіз та обробка інформації, налагодження роботи програмного продукту, тестування функціональних можливостей додатку та усунення помилок у процесі розробки.

Автоматизоване робоче місце розробника включає персональний комп'ютер або ноутбук, монітор, клавіатуру, комп'ютерну мишу, мережеве обладнання, засоби підключення до мережі Інтернет та спеціалізоване програмне забезпечення для розробки мобільних додатків. Також у процесі роботи можуть використовуватися мобільні пристрої для тестування програмного забезпечення.

Робота розробника мобільного додатку виконується переважно в офісному приміщенні або в умовах дистанційної роботи. Основною особливістю функціонування даного робочого місця є тривала робота за комп'ютером, значне інтелектуальне навантаження та необхідність постійної концентрації уваги. Під час виконання професійних обов'язків працівник тривалий час перебуває у сидячому положенні, працює з великим обсягом інформації та використовує електронне обладнання [35].

Умови праці розробника мобільного додатку характеризуються впливом ряду небезпечних та шкідливих виробничих факторів. До основних факторів належать напруження зору, статичне фізичне навантаження, нервово-емоційне перенапруження, вплив електромагнітного випромінювання, а також ризик ураження електричним струмом у разі несправності обладнання або порушення правил експлуатації електротехнічних засобів [36].

Під час роботи важливе значення мають параметри мікроклімату приміщення, рівень освітлення, ергономічність робочого місця та організація режиму праці й відпочинку. Недостатнє освітлення, тривала безперервна робота за монітором та неправильне розташування обладнання можуть негативно впливати на стан здоров'я працівника, викликати втому, зниження працездатності та розвиток професійних захворювань [36].

Для виявлення потенційних небезпек було проведено аналіз умов праці на автоматизованому робочому місці розробника мобільного додатку на платформі iOS. Результати аналізу наведені у таблиці 4.1.

Таблиця 4.1 – Результати аналізу умов праці розробника мобільного додатку [35, 36]

№	Потенційна небезпека	Джерело небезпеки	Можливі наслідки
1	Напруження зору	Тривала робота за монітором, неправильне освітлення	Погіршення зору, втома очей, головний біль
2	Статичне фізичне навантаження	Робота тривалий час у положенні сидячи	Захворювання опорно-рухового апарату
3	Нервово-емоційне навантаження	Висока концентрація уваги, великі обсяги інформації, дедлайни	Перевтома, стрес, зниження працездатності
4	Недостатнє освітлення робочого місця	Невідповідність нормам освітлення	Швидка втомлюваність, погіршення концентрації уваги
5	Електромагнітне випромінювання	Монітор, ноутбук, мережеве обладнання	Погіршення самопочуття, підвищена втома
6	Перегрів обладнання	Тривала робота комп’ютерної техніки	Збої в роботі обладнання, ризик короткого замикання
7	Монотонність праці	Виконання однотипних операцій	Психоемоційна втома, зниження уважності
8	Недостатня організація режиму праці та відпочинку	Тривала безперервна робота за ПК	Хронічна втома, зниження продуктивності праці

Для зниження рівня професійних ризиків необхідно забезпечити дотримання вимог охорони праці, правил електробезпеки та ергономічних норм організації робочого місця.

4.3 Дослідження ризику реалізації потенційних небезпек на об’єкті проектування та розробка заходів щодо їх попередження

Оцінювання ризиків є одним із найважливіших елементів системи управління охороною праці. Його основною метою є своєчасне виявлення потенційних небезпек, визначення рівня їх впливу на працівників та розробка заходів щодо попередження нещасних випадків і професійних захворювань. Проведення оцінки ризиків дозволяє не лише підвищити рівень безпеки праці, але й створити більш комфортні та ефективні умови трудової діяльності.

У сфері інформаційних технологій та під час роботи розробника мобільного додатку на платформі iOS питання оцінювання ризиків мають особливе значення, оскільки своєчасне виявлення небезпек дозволяє знизити ймовірність виникнення аварійних ситуацій та мінімізувати негативний вплив на здоров’я працівника.

Важливу роль у процесі управління охороною праці відіграє правильний вибір методу оцінювання ризиків. Використання відповідного методу дозволяє більш точно визначити причини виникнення небезпек, оцінити рівень ризику та встановити найбільш ефективні профілактичні заходи. Неправильно обраний метод оцінювання ризиків може призвести до неповного аналізу небезпечних факторів, що знижує ефективність системи управління охороною праці та підвищує ймовірність виникнення нещасних випадків.

Для аналізу причин виникнення небезпечних ситуацій на автоматизованому робочому місці розробника мобільного додатку доцільно використовувати метод «дерева відмов». Даний метод є одним із найбільш поширених способів аналізу ризиків та дозволяє детально дослідити причини виникнення небезпечної події шляхом побудови логічної схеми взаємозв’язків між окремими факторами [37].

Метод «дерева відмов» базується на послідовному визначенні причин, які можуть призвести до виникнення певної небезпечної ситуації. Аналіз

починається з визначення основної небезпечної події, після чого встановлюються всі можливі причини її виникнення. Для побудови логічних зв’язків між подіями використовуються операції «ТА» та «АБО», що дозволяє більш наочно відобразити взаємозалежність між різними факторами ризику.

Однією з головних переваг використання «дерева відмов» є можливість проведення детального аналізу першопричин небезпеки. Даний метод дозволяє визначити не лише безпосередні причини виникнення небезпечної ситуації, але й організаційні, технічні та людські фактори, які можуть впливати на рівень ризику. Це дає можливість більш ефективно планувати заходи щодо попередження небезпечних ситуацій та усунення причин їх виникнення [37].

Ще однією перевагою методу є наочність представлення результатів аналізу. Побудова логічної схеми дозволяє легко визначити найбільш критичні фактори ризику та встановити взаємозв’язок між окремими подіями. Це значно спрощує процес аналізу ризиків та підвищує ефективність прийняття рішень у сфері охорони праці.

Крім того, метод «дерева відмов» дозволяє оцінити вплив одночасної дії кількох небезпечних факторів, що є особливо важливим для автоматизованих робочих місць у сфері інформаційних технологій. Наприклад, виникнення небезпеки ураження електричним струмом може бути пов’язане одночасно з несправністю обладнання, пошкодженням ізоляції кабелів та порушенням правил експлуатації електротехнічних засобів [37].

Результати використання даного методу зображені на рис 4.1. Побудоване «дерево відмов» для небезпеки монотонності праці дозволяє визначити основні причини виникнення психоемоційної втоми та зниження працездатності працівника. Аналіз показує, що найбільший вплив мають тривале виконання однотипних операцій, відсутність достатніх перерв під час роботи та постійне навантаження на увагу під час розробки програмного забезпечення.

Монотонність праці є одним із поширених психофізіологічних факторів ризику для працівників сфери інформаційних технологій. Під час тривалого

виконання однотипних завдань у розробника мобільного додатку поступово знижується концентрація уваги, з'являється втома, погіршується швидкість реакції та підвищується ймовірність помилок у роботі. Саме тому важливим напрямком охорони праці є впровадження заходів, спрямованих на зменшення монотонності трудового процесу та підтримання стабільного психоемоційного стану працівника.

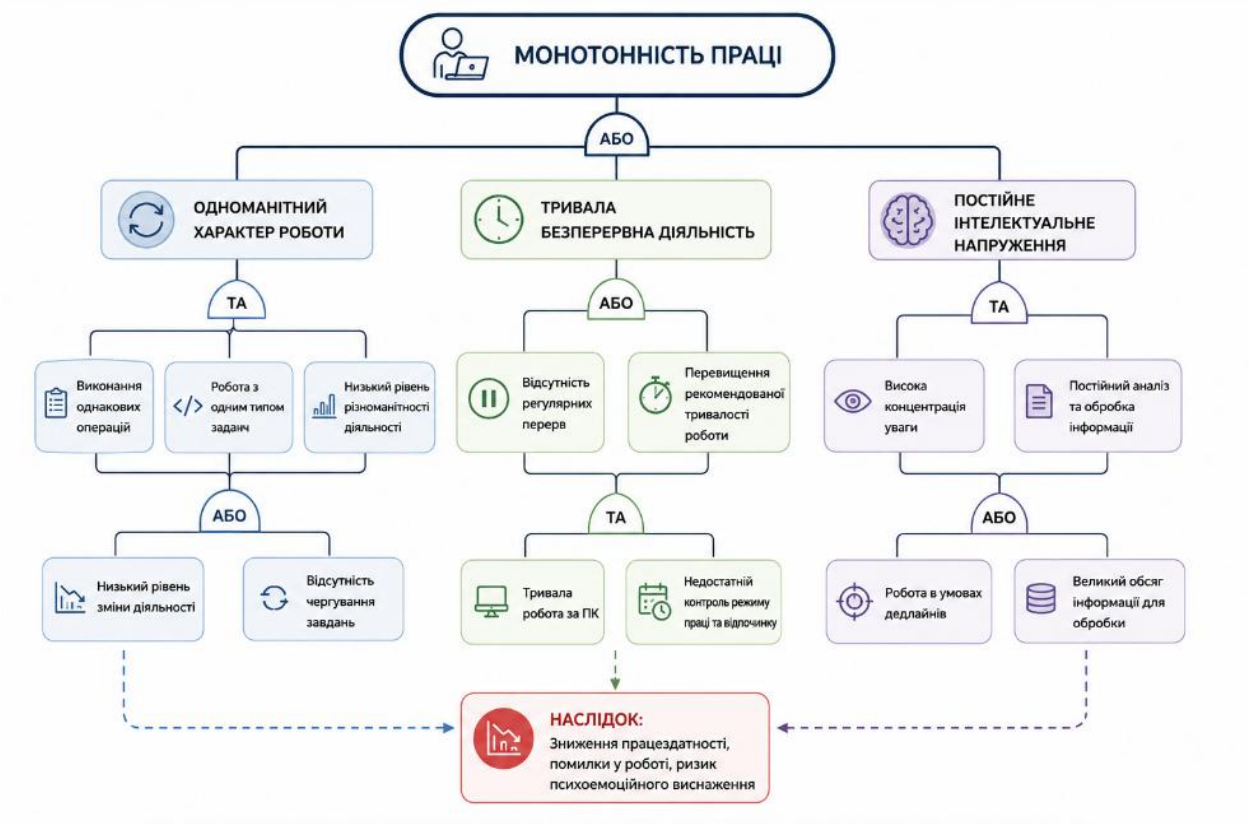

Рисунок 4.1 – Результати побудови дерева відмов для небезпеки, пов'язаної із монотонністю праці

Одним із найбільш ефективних способів боротьби з монотонністю праці є правильна організація режиму праці та відпочинку [38]. Під час роботи за комп'ютером рекомендується робити короткі перерви через кожні 45-60 хвилин роботи. Під час перерв працівнику доцільно змінювати вид діяльності, виконувати нескладні фізичні вправи або коротку розминку. Це дозволяє знизити психоемоційне навантаження та покращити загальний стан організму.

Для зменшення одноманітності роботи ефективним є чергування різних типів завдань [39]. Наприклад, процес програмування може поєднуватися з

тестуванням додатку, аналізом документації, перевіркою інтерфейсу або командною взаємодією. Зміна характеру діяльності дозволяє знизити рівень психологічної втоми та підтримувати увагу працівника протягом робочого дня.

Важливу роль відіграє створення комфортного та ергономічного робочого середовища. Наявність зручного робочого місця, достатнього освітлення, оптимального мікроклімату та низького рівня шуму позитивно впливає на працездатність працівника та знижує рівень перевтоми. Для працівників сфери IT також важливим є забезпечення психологічного комфорту та зменшення рівня стресових факторів.

Серед сучасних практик профілактики монотонності праці все більшого поширення набуває використання гнучкого режиму роботи [38, 39]. Можливість самостійно планувати робочий час, змінювати послідовність виконання завдань та працювати у комфортному темпі позитивно впливає на емоційний стан працівників і знижує рівень професійного вигорання.

Ефективним методом боротьби з монотонністю є також використання методики «Pomodoro». Даний підхід передбачає поділ робочого часу на короткі інтервали тривалістю приблизно 25 хвилин із подальшими короткими перервами. Такий режим дозволяє підтримувати високу концентрацію уваги та знижує психічне виснаження під час тривалої роботи з програмним кодом.

Для підтримання інтересу до роботи сучасні IT-компанії часто використовують практику командної взаємодії та колективного обговорення задач. Проведення робочих зустрічей, обговорення нових ідей та участь у спільному вирішенні проблем допомагають зменшити відчуття ізольованості та одноманітності праці [38, 39].

Також позитивний вплив має використання сучасних систем автоматизації та інструментів штучного інтелекту, які дозволяють зменшити кількість рутинних операцій під час розробки програмного забезпечення. Автоматизація повторюваних процесів дає можливість працівнику більше

уваги приділяти творчим та аналітичним задачам, що значно знижує рівень монотонності роботи.

Важливим профілактичним заходом є підтримання сприятливого психологічного клімату в колективі. Дружня атмосфера, підтримка з боку колег та керівництва, а також можливість професійного розвитку позитивно впливають на мотивацію працівника та знижують ризик психоемоційного виснаження [38, 39].

Висновки до четвертого розділу

Охорона праці є важливою складовою організації сучасної трудової діяльності, особливо у сфері інформаційних технологій, де працівники постійно взаємодіють із комп’ютерною технікою, програмним забезпеченням та електронними засобами обробки інформації. Забезпечення безпечних та комфортних умов праці спрямоване на збереження здоров’я працівників, підвищення рівня працездатності та попередження виникнення професійних захворювань і нещасних випадків.

У розділі з охорони праці було досліджено автоматизоване робоче місце розробника мобільного додатку на платформі iOS. Встановлено, що особливістю даного робочого місця є тривала робота за персональним комп’ютером, значне інтелектуальне навантаження, необхідність постійної концентрації уваги та виконання великої кількості аналітичних і програмних операцій. Такі умови праці можуть супроводжуватися впливом різних небезпечних та шкідливих факторів, серед яких статичне фізичне навантаження, напруження зору, нервово-емоційне перенапруження, монотонність праці та ризик ураження електричним струмом.

Особливу увагу в роботі було приділено дослідженню монотонності праці як одного з найбільш характерних психофізіологічних факторів для працівників ІТ-сфери. У результаті дослідження були запропоновані сучасні організаційні та ергономічні заходи, спрямовані на зниження рівня

монотонності праці та покращення умов роботи розробника мобільного додатку.

ЗАГАЛЬНІ ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було вирішено актуальне науково-практичне завдання, що полягає у розробці мобільного додатка для підтримки діяльності «таємного покупця» та підвищення ефективності контролю якості обслуговування у сфері роздрібної торгівлі та послуг. Проведений аналіз предметної області показав, що традиційні підходи до організації перевірок є недостатньо ефективними через фрагментарність збору даних, значні часові витрати на їх обробку та відсутність централізованих інструментів аналітики. Це обґрунтувало необхідність створення сучасного цифрового рішення на базі мобільних технологій.

У процесі дослідження було виконано системний аналіз існуючих програмних аналогів, що дозволило виявити їх ключові переваги та недоліки. Зокрема встановлено, що наявні рішення не забезпечують повного функціонального покриття задач користувачів, мають обмежені можливості звітності, комунікації та контролю виконання завдань. На основі цього було сформовано сукупність функціональних і нефункціональних вимог до розроблюваної системи, які стали основою для подальшого проєктування.

На етапі проєктування було розроблено архітектуру мобільного додатка та серверної частини із застосуванням сучасних підходів до побудови інформаційних систем. Використання архітектурного патерну MVC забезпечило розділення відповідальностей між компонентами системи, що підвищує її масштабованість та спрощує подальший супровід. Побудовані UML-діаграми дозволили формалізувати структуру системи та логіку взаємодії між її складовими, що є важливим для забезпечення узгодженості реалізації.

У ході реалізації програмного продукту було створено повнофункціональний мобільний додаток із використанням сучасних інструментів розробки та бібліотек. Реалізовано механізми авторизації користувачів, управління завданнями, формування фотозвітів і звітів за

цінами, обробки штрих-кодів, а також взаємодії з сервером через REST API. Використання локальної бази даних дозволило забезпечити ефективне збереження та обробку інформації на стороні клієнта, а інтеграція зовнішніх бібліотек підвищила продуктивність і надійність роботи системи.

Окрему увагу було приділено тестуванню програмного забезпечення. Проведено функціональне, інтеграційне та модульне тестування, результати якого підтвердили коректність реалізації всіх основних сценаріїв використання системи, стабільність її роботи на різних пристроях і відповідність визначеним вимогам. Це свідчить про готовність програмного продукту до практичного впровадження.

Таким чином, у роботі досягнуто поставлену мету — розроблено мобільний додаток для «таємного покупця», який забезпечує автоматизацію процесу збору та аналізу даних про якість обслуговування. Практичне значення отриманих результатів полягає у можливості використання розробленої системи підприємствами для підвищення рівня контролю, оптимізації бізнес-процесів та покращення клієнтського досвіду.

Перспективи подальшого розвитку полягають у розширенні функціональних можливостей системи, зокрема впровадженні аналітичних модулів, використанні технологій машинного навчання для обробки зібраних даних, а також масштабуванні рішення для підтримки більшої кількості користувачів і бізнес-сценаріїв.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скільки українців щодня користуються інтернетом: результати дослідження КМІС Електронний ресурс. URL: https://24tv.ua/tech/statistika-koristuvannya-internetom-zadovolenosti-tsifrovimi_n2737047 (дата звернення 22.05.2026 р.).

2. В Україні вирівнюється «цифровий розрив» між різними соціальними групами в користуванні інтернетом – дослідження Електронний ресурс. URL: https://detector.media/rinok/article/138653/2018-06-19-v-ukraini-vyrivnyuietsya-tsyfrovyy-rozryv-mizh-riznymy-sotsialnymy-grupamy-v-korystuvanni-internetom-doslidzhennya/ (дата звернення 22.05.2026 р.).

3. Таємний покупець Електронний ресурс. URL: https://uk.wikipedia.org/wiki/%D0%A2%D0%B0%D1%94%D0%BC%D0%BD%D0%B8%D0%B9_%D0%BF%D0%BE%D0%BA%D1%83%D0%BF%D0%B5%D1%86%D1%8C/ (дата звернення 22.05.2026 р.).

4. Таємний покупець – індикатор якості обслуговування Електронний ресурс. URL: https://mtp-global.com/uk/tajnyj-pokupatel/ (дата звернення 22.05.2026 р.).

5. Як організувати перевірку «Таємний покупець» у своєму магазині Електронний ресурс. URL: https://torgsoft.ua/articles/stati/tajemnyj-pokupets/ (дата звернення 25.05.2026 р.).

6. Бєлова К. О, Бредіхін В. М. Роль підготовки та очищення даних для підвищення якості роботи моделей інтелектуального аналізу / матеріали III (IX) Міжнародної науково-практичної конференції здобувачів вищої освіти і молодих учених «Інформаційні технології: теорія і практика», С. 432-434.

7. Робота таємним покупцем: суть професії, переваги, оплата Електронний ресурс. URL: https://grc.ua/blog/robota-tayemnym-pokupcem-sutj-profesiyi-perevagy-oplata (дата звернення 25.05.2026 р.).

8. 4Shoppers Таємний Покупець - App Store - Apple Електронний ресурс. URL: https://apps.apple.com/ru/app/4shoppers-

%D1%82%D0%B0%D1%94%D0%BC%D0%BD%D0%B8%D0%B9-%D0%BF%D0%BE%D0%BA%D1%83%D0%BF%D0%B5%D1%86%D1%8C/id927374681?l=uk (дата звернення 25.05.2026 р.).

9. Google запустив новий сервіс Task Mate для легкого заробітку Електронний ресурс. URL: https://pingvin.pro/gadgets/news-gadgets/google-zapustyv-novyj-servis-task-mate-dlya-legkogo-zarobitku.html (дата звернення 25.05.2026 р.).

10. Perfect Store - by TransMore Електронний ресурс. URL: https://play.google.com/store/apps/details?id=com.transmora.perfectstore&hl=uk (дата звернення 25.05.2026 р.).

11. Bino CX Електронний ресурс. URL: https://www.behance.net/gallery/54403901/BINO-CX (дата звернення 25.05.2026 р.).

12. MD Audit Електронний ресурс. URL: https://play.google.com/store/apps/details?id=ru.mdaudit.osa&hl=en_GB (дата звернення 25.05.2026 р.).

13. Бєлова К. Експертний застосунок оцінювання якості обслуговування в роздрібній торгівлі / Матеріали XI Всеукраїнської студентської науково-технічної конференції «Сталий розвиток міст» (83-ї студентської науково-технічної конференції ХНУМГ ім. О. М. Бекетова). С. 124-126.

14. Swift is the powerful, flexible, multiplatform programming language Електронний ресурс. URL: https://www.swift.org/ (дата звернення 29.05.2026 р.).

15. Посібник із побудови схем UML і моделювання баз даних Електронний ресурс. URL: https://www.microsoft.com/uk-ua/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling (дата звернення 29.05.2026 р.).

16. Платформа iOS що це таке Електронний ресурс. URL: https://nedo.biolio.cx.ua/articles/platforma-ios-shho-ce-take.html (дата звернення 29.05.2026 р.).

17. REST API: принципи роботи веб-сервісів Електронний ресурс. URL: https://wezom.com.ua/ua/blog/rest-api-printsipi-roboti-veb-servisiv (дата звернення 02.06.2026 р.).

18. IDE Xcode Електронний ресурс. URL: https://developer.apple.com/xcode/ (дата звернення 02.06.2026 р.).

19. iOS SDK Електронний ресурс. URL: https://uk.wikipedia.org/wiki/IOS_SDK (дата звернення 02.06.2026 р.).

20. Interface Builder Електронний ресурс. URL: https://developer.apple.com/forums/tags/interface-builder (дата звернення 02.06.2026 р.).

21. CocoaTouchFramework Електронний ресурс. URL: https://cocoapods.org/pods/CocoaTouchFramework (дата звернення 02.06.2026 р.).

22. UIKit Framework Електронний ресурс. URL: https://getuikit.com/ (дата звернення 02.06.2026 р.).

23. Foundation Kit Framework Електронний ресурс. URL: https://en.wikipedia.org/wiki/Foundation_Kit (дата звернення 02.06.2026 р.).

24. Build the future of apps with Xamarin Електронний ресурс. URL: https://xamarin.github.io/ (дата звернення 02.06.2026 р.).

25. Core Data Електронний ресурс. URL: https://developer.apple.com/documentation/coredata/ (дата звернення 02.06.2026 р.).

26. Май, Т. Н. Мобільний додаток для парсингу медійних даних на платформі iOS : дипломний проект бакалавра : 123 Комп’ютерна інженерія / Май Тієн Ноанг. – Київ, 2023. – 91 с.

27. SQLite Електронний ресурс. URL: https://sqlite.org/ (дата звернення 02.06.2026 р.).

28. SQlite - визначення, основні характеристики Електронний ресурс. URL: https://freehost.com.ua/ukr/faq/wiki/chto-takoe-sqlite/ (дата звернення 02.06.2026 р.).

29. A Bit About Interface Builder Електронний ресурс. URL: https://medium.com/swlh/a-bit-about-interface-builder-ceffaf484580 (дата звернення 08.06.2026 р.).

30. Що таке JSON. Усе про цей формат передачі даних в інтернеті Електронний ресурс. URL: https://apix-drive.com/ua/blog/useful/scho-take-json (дата звернення 08.06.2026 р.).

31. Види функціонального та нефункціонального тестування Електронний ресурс. URL: https://dan-it.com.ua/uk/blog/vidy-funkcionalnogo-i-nefunkcionalnogo-testirovanija/ (дата звернення 10.06.2026 р.).

32. XCTest Best Practices for iOS Testing Електронний ресурс. URL: https://maestro.dev/insights/xctest-best-practices-ios-testing (дата звернення 10.06.2026 р.).

33. Інтеграційне тестування Електронний ресурс. URL: https://qalight.ua/baza-znaniy/integratsijne-testuvannya/ (дата звернення 10.06.2026 р.).

34. Закон України «Про охорону праці» / Офіційний сайт Верховної Ради України. – Режим доступу: https://zakon.rada.gov.ua/go/2694-12 (дата звернення 10.06.2026 р.).

35. Wahlström J. (2005). Ergonomics, musculoskeletal disorders and computer work. Occupational medicine (Oxford, England). 55. 168-76. 10.1093/occmed/kqi083.

36. Maudgalya T., Wallace S., Daraiseh N., Salem O. (2006). Workplace stress factors and 'burnout' among information technology professionals: A systematic review. Theoretical Issues in Ergonomics Science. 7. 285-297. 10.1080/14639220500090638.

37. What is Fault Tree Analysis (FTA)? – Режим доступу: https://www.phmtechnology.com/functionality/what-is-a-fault-tree-analysis.html (дата звернення 12.06.2026 р.).

38. Office Ergonomics. – Режим доступу: https://www.ccohs.ca/oshanswers/ergonomics/office (дата звернення 12.06.2026 р.).

39. Psychosocial risks and mental health at work. – Режим доступу: https://osha.europa.eu/en/themes/psychosocial-risks-and-mental-health (дата звернення 12.06.2026 р.).