

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему «Мобільний додаток підтримки діяльності будівельної компанії»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»
освітньо-кваліфікаційного рівня «магістр»

Виконавець роботи: студент групи ІТ.м-12 Семеняко Сергій Олександрович

Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою _____

«___» грудня 2022 р.

Науковий керівник _____

(підпис)

к.т.н., доц. Нагорний В.В.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____

(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ІТ

_____ С. М. Ващенко
«__» _____ 2022 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Семеняко Сергій Олександрович
(прізвище, ім'я, по батькові)

- 1 Тема проекту** Мобільний додаток підтримки діяльності будівельної компанії
затверджена наказом по університету від «04» листопада 2022 р. №1013-VI
- 2 Термін здачі студентом закінченого проекту** «12» грудня 2022 р.
- 3 Вхідні дані до роботи** перелік вимог на розробку мобільного додатку підтримки діяльності будівельної компанії
- 4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** аналіз предметної області, постановка задачі та методи дослідження, моделювання та проектування, розробка мобільного додатку.
- 5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** актуальність, мета та задачі, об'єкт та предмет дослідження, аналіз існуючих аналогів, вимоги до мобільного додатку, контекстна діаграма в нотації IDEF0, декомпозиція першого рівня в нотації IDEF0, діаграма варіантів використання мобільного додатку, логічна модель, архітектура інформаційної системи, засоби реалізації, приклад роботи мобільного додатку, висновки.

6 Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускного проекту	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	29.10.22 – 05.11.22	
2	Огляд існуючих аналогів	06.11.22 – 08.11.22	
3	Постановка задачі щодо створення мобільного додатку	09.11.22 – 11.11.22	
4	Опис методів дослідження	12.11.22 – 14.11.22	
5	Вибір засобів реалізації	15.11.22 – 17.11.22	
6	Проектування процесу дослідження	18.11.22 – 20.11.22	
7	Моделювання варіантів використання	21.11.22 – 22.11.22	
8	Проектування бази даних	23.11.22 – 24.11.22	
9	Розробка необхідних модулів	25.11.22 – 30.11.22	
10	Підключення розроблених модулів	01.12.22 – 04.12.22	
11	Тестування мобільного додатку	05.12.22 – 09.12.22	
12	Реліз мобільного додатку	10.12.22 – 15.12.22	

Магістрант _____
(підпис)

Семеняко С.О.

Керівник роботи _____
(підпис)

к.т.н., доц. Нагорний В.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Мобільний додаток підтримки діяльності будівельної компанії».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 27 найменувань та 2 додатки. Загальний обсяг роботи – 64 сторінок, у тому числі 45 сторінок основного тексту, 3 сторінки списку використаних джерел та 4 сторінок додатків.

Кваліфікаційну роботу магістра присвячено розробці мобільного додатку підтримки діяльності будівельної компанії.

У роботі було проведено аналіз предметної області, проаналізована актуальність проблеми, оглянуто мобільні додатки аналоги, сформульовані функціональні вимоги, сформульовано мету та задачі, описано методи дослідження та обрано засоби реалізації.

У роботі виконано структурно функціональне моделювання, змодельовано варіанти використання, а також спроектовано базу даних.

Результатом проведеної роботи є реалізований мобільний додаток підтримки діяльності будівельної компанії.

Практичне значення роботи полягає у полегшенні завдання отримання інформації про проведену роботу на ділянці робітниками. Через мобільний додаток працівники завжди зможуть надсилати звіти про виконану роботу із прикріпленими фотографіями виконаної роботи.

Ключові слова: мобільний додаток, Android Studio, база даних, Java, робітник, звіт, ділянка, середовище розробки, мова програмування.

ЗМІСТ

РЕФЕРАТ	4
ВСТУП	6
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1. Огляд існуючих аналогів	7
ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	14
2.1. Мета та задачі	14
2.2. Обґрунтування вибору операційної системи	15
2.3 Огляд середовищ програмування, що використовуються для створення програм під ОС Android.....	17
2.3.1. Android Studio	17
2.3.2. Eclipse	18
2.3.2. IntelliJ IDEA	18
2.3.3. Висновок щодо вибору середовища розробки	19
2.4 Огляд мов програмування для ОС Android	19
2.4.1. JavaScript	20
2.4.2. C#	21
2.4.3. Java	22
2.4.4. Висновок щодо вибору мови програмування.....	22
МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ	23
3.1 Структурно функціональне моделювання.....	23
3.2 Моделювання варіантів використання.....	25
3.3 Проектування бази даних	26
РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ	30
4.1 Архітектура мобільного додатку	30
4.2 Програмна реалізація мобільного додатку	31
4.2 Використання мобільного додатку користувачем	33
ВИСНОВКИ	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	40
ДОДАТОК А	43
ДОДАТОК Б.	47

ВСТУП

У теперішній час компактні та прості у використанні пристрої грають у житті людини велику роль. Зростання їх популярності призвело до того, що сучасна людина практично не уявляє свого життя без їх використання.

Сучасні інформаційні технології впливають на багато галузей: економіки, промисловість, сільське господарство, торгівля та будівельну [1].

Починалося застосування ІТ у будівництві з вирішення нескладних розрахункових завдань. В даний же час – це найскладніші системи управління комплексними проектами: починаючи з проектування будівель, споруд, інженерних комунікацій та закінчуючи автоматизованими засобами контролю об'єктів державного нагляду.

На відміну від мобільного телефону, який завжди знаходиться поруч із власником, користувач не завжди має можливість бути на місці ремонтних робіт з тих чи інших причин. І навіть зробивши спроби дізнатися інформацію про те, як просуваються роботи на ділянці, зателефонувавши працівнику, не завжди можна точно зрозуміти по телефонній розмові кількість і якість виконаної роботи.

Полегшити завдання отримання інформації про проведену роботу якраз може мобільний пристрій, який завжди поруч, а точніше мобільний додаток на ньому. Через мобільний додаток працівники завжди зможуть надсилати звіти про виконану роботу із прикріпленими фотографіями виконаної роботи.

Таким чином, роботу, спрямовану на створення мобільного додатка, слід вважати актуальною.

Об'єктом дослідження являється підтримка діяльності будівельної компанії.

Предметом дослідження являється мобільний додаток підтримки діяльності будівельної компанії.

Мета дослідження полягає у розробці мобільного додатку підтримки діяльності будівельної компанії.

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Огляд існуючих аналогів

Управління будівельним об'єктом вимагає серйозної відповідальності, і для її підкріплення потрібне використання різних інструментів. Так, сучасний смартфон або планшет можуть бути досить цінним і практичним інструментом для різних завдань на будівельному об'єкті, особливо якщо використовувати багатофункціональні програми.

Нижче було розглянуто 3 мобільні додатки аналоги.

Timesheet – програма обліку робочих годин. Зручно і точно зафіксує, скільки було витрачено часу на завдання, обслуговування об'єкта, замовника. Можна формувати звіти та рахунки. Зручна навігація, детальний функціонал: автопідрахунок переробок, тариф робіт, фільтри за потрібними відрізками часу.

В додатку присутня навігація у календарі показана на рисунку 1.1

← Табель							
декабря 2019 (10:00h)							
Неделя	ВС	ПН	ВТ	СР	ЧТ	ПТ	СБ
49	1	2	3	4	5	6	7
50	8	9	10	11	12	13	14
51	15	16	17	+	19	20	21
10:00h	22	23	24	25	26	27	28
1	29	30	31	1	2	3	4
30:00h				7:30h	7:30h	7:30h	7:30h
4входы				10:00h	\$100,00		
9:00 AM - 1:00 PM							
Aadhk Timesheet App							
Design Work							
3:30h							
\$35,00							
2:00 PM - 5:00 PM							
Aadhk Timesheet App							
Development Work							
2:30h							
\$25,00							
6:00 PM - 9:00 PM							
Aadhk Timesheet App							
Personal							
3:00h							
\$30,00							
10:00 PM - 11:00 PM							
Aadhk Timesheet App							
1:00h							
\$10,00							

Рисунок 1.1 – Навігація у календарі

Monday – один з найвдаліших для підвищення ефективності в управлінні командою та проектів додаток. Планування та налаштування робочого процесу, наочна візуалізація стану роботи, відстеження завдань, спілкування з усією командою. Мінус – лише для управлінських завдань. Не підтримує завантаження таких матеріалів, як креслення.

В додатку є можливість управління командами та проектами (рис. 1.4)

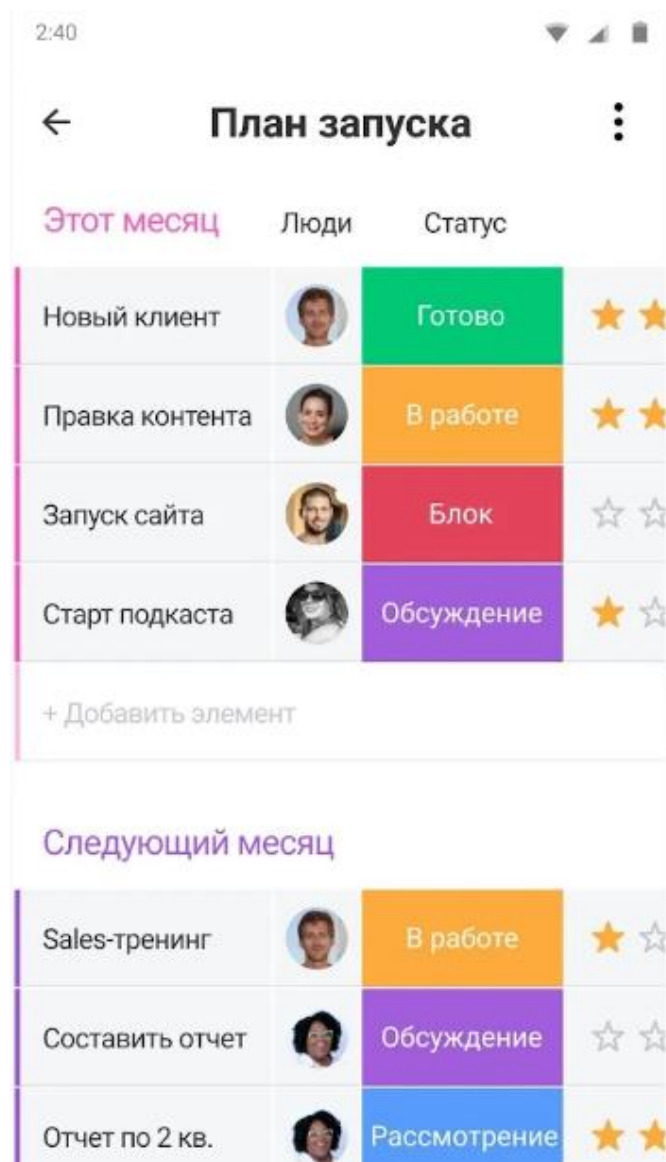


Рисунок 1.4 – Управління командами та проектами

В наявності є можливість планування, організації та відстеження роботи (рис. 1.5)

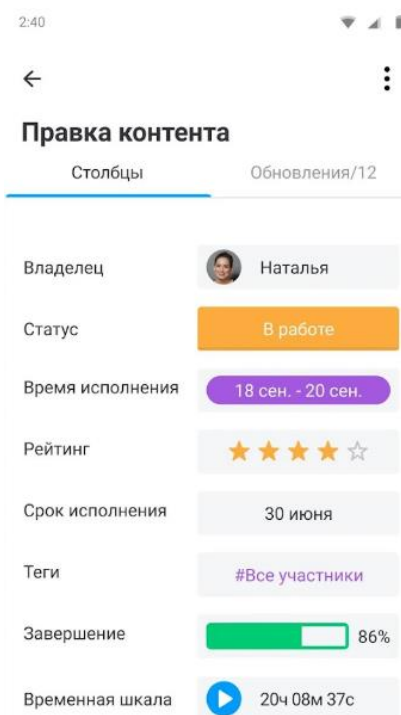


Рисунок 1.5 – Планування, організації та відстеження роботи

В додатку присутня можливість спілкування з командою (рис. 1.6)

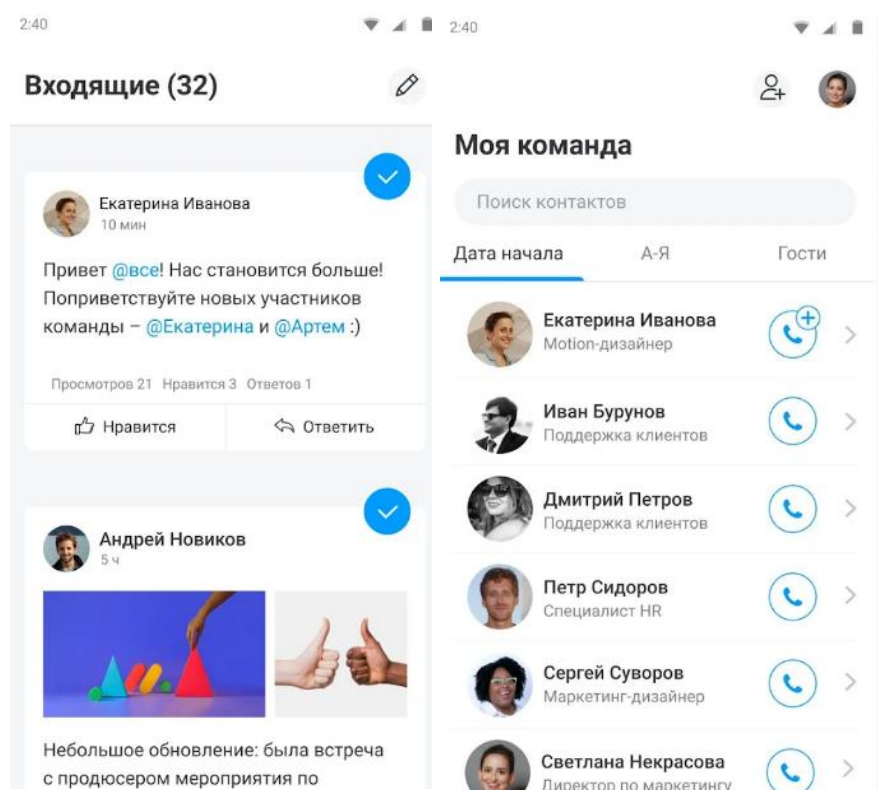


Рисунок 1.6– Спілкування з командою

Connecteam – програма допоможе об'єднати роботу команди віддалено за допомогою мобільного телефону. Спілкування в режимі реального часу, призначення та контроль завдань, термінів виконання, графік роботи, облік робочого часу, фідбек від співробітників. Цей додаток допоможе оптимізувати менеджмент та підвищувати продуктивність роботи цілої команди поза офісом. Гнучкі налаштування, доброзичливий інтерфейс.

Додаток дозволяє переглядати свій робочий графік і прогрес роботи (рис. 1.7)

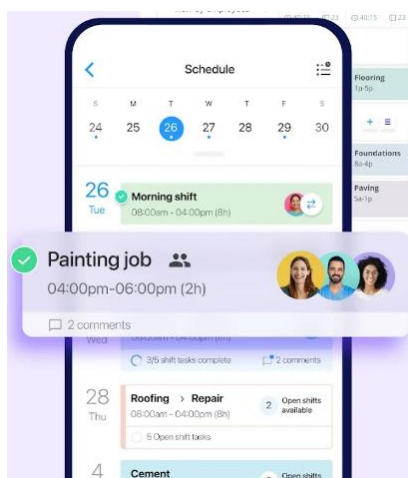


Рисунок 1.7 – Перегляд графіку і прогресу роботи

Є можливість створення звітів

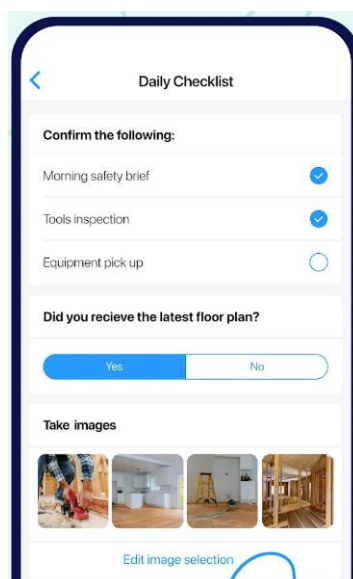


Рисунок 1.8 – Заповнення звітів

В додатку є можливість додавати людей до групового чату (рис. 1.9)

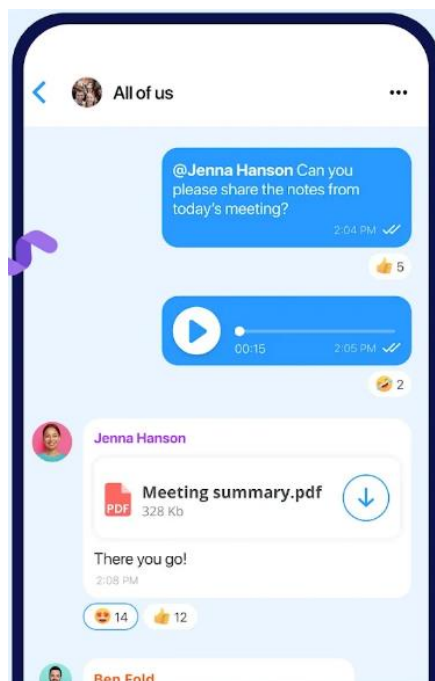


Рисунок 1.9 – Груповий чат

На основі переглянутих додатків аналогів Timesheet, Monday та Connecteam була створена таблиця порівняння додатків аналогів, для порівняння також в таблицю була додана інформація про розроблюваний додаток підтримки діяльності будівельної компанії.

Таблиця 1.1 – Таблиця порівняння додатків аналогів

Критерії порівняння	Timesheet	Monday	Connecteam	Розроблюваний додаток
Наявність групового чату	-	-	+	-
Створення звітів	звіт тільки про відпрацьований час	-	+	+
Відслідковувати прогрес роботи	-	+	+	+

Продовження таблиці 1.1 – Таблиця порівняння додатків аналогів

Критерії порівняння	Timesheet	Monday	Connecteam	Розроблюваний додаток
Управління командою	-	+	+	+
Можливість оплати витрат	-	-	-	+
Відслідковування витрат	+	-	-	+

Провівши порівняння та проаналізувавши три додатки Timesheet, Monday та Connecteam було вирішено створити власний мобільний додаток в якому будуть відсутні такі недоліки які були розглянуті в таблиці 1.1. Також було прийнято рішення про відсутність в розроблюваному додатку такої функції як груповий чат. На нинішній день вже існує дуже багато менеджерів, такі як WhatsApp, Viber, Telegram, Skype і так далі. Один з них точно встановлений у середньостатистичного користувача смартфона, і створення ще одного буде не дуже доречним, бо користувач може заплутатися і не згадати в якому саме менеджері він спілкувався з тією чи іншою особою.

ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1. Мета та задачі

Метою випускної кваліфікаційної роботи магістра є розробка мобільного додатку підтримки діяльності будівельної компанії.

Щоб досягти даної мети, треба виконати наступні задачі:

- проаналізувати актуальність проблеми;
- провести аналіз вже існуючих рішень на ринку;
- сформулювати функціональні вимоги;
- провести моделювання бізнес-процесів;
- реалізувати та протестувати додаток.

У додатку мають бути реалізовані такі функції:

- можливість дізнатися, чи ведуться роботи на ділянці;
- переглядати статтю витрат;
- перегляд звітів про виконану роботу;
- працівники могли б надсилати звіти з фото про виконану роботу протягом дня;
- можливість оплати послуг через додаток

При проектуванні системи необхідно врахувати такі вимоги:

1. Інтерфейс системи повинен бути максимально гнучким, щоб користувач не витрачав багато часу для освоєння сервісу.
2. Дизайн програми не повинен бути громіздким.
3. Вікно авторизації не повинно вимагати від користувача величезної кількості інформації, що вводиться
4. Для роботи програми потрібен доступ до інтернету, весь процес роботи з сервісом повинен здійснюватися в онлайн-режимі.

5. Додавання нових елементів не повинно порушувати архітектуру та цілісність програми.

2.2. Обґрунтування вибору операційної системи

При виборі операційної системи для мобільного пристрою розробник повинен керуватися такими факторами як:

- поширеність на світовому ринку;
- зручність розробки та поширеність на внутрішньому ринку (для збору відгуків та оцінки поведінки програми до виходу на світовий ринок).

Більшість мобільного ринку формують переважно дві платформи – Android та iOS. Дослідницька компанія Gartner надала новий звіт на лютий 2020 року, вказавши, що ринок мобільних пристроїв займають лише дві корпорації – Google, з платформою Android та Apple, з платформою iOS [2] (табл. 1.1).

Таблиця 2.1 – Частка поширення мобільних операційних систем, світові продажі на 2019-20 рік (В тис. од.)

ОС	продано в 2020 р., од.	частка ринку 2020 р., од.	продано у 2019 р., од.	частка ринку 2019 р., од.
Android	1,320,118.1	85.9	1,268,562.7	84.8
iOS	214,924.4	14.0	216,064.0	14.4
Інші	1,493.0	0.1	1,332.2	0.8
Усього	1,536,535.5	100.0	1,495,959.0	100.0

Отже, судячи з представлених даних у таблиці, можна зрозуміти, що велику частку світового ринку зайняла ОС Android з часткою 85.9%, на наступному місці розташувалася ОС iOS з часткою 14.0%. Частка, що залишилася, в 0.1% припала на гаджети під управлінням інших операційних систем.

Проаналізувавши представлені дані, можна зробити висновок про те, що найбільш оптимальним вибором операційної системи для мобільного додатка є ОС

Android. Цей вибір обґрунтований не лише часткою зайнятості ринку, а й повною відкритістю системи, що дозволяє розробляти програми без порушення будь-яких ліцензійних угод [3].

Виділяючи основні особливості системи, можна перерахувати позитивні властивості ОС Android:

1. Android – відкрита операційна система.
2. Простота налаштування системи для пересічного користувача.
3. Великий вибір безкоштовних програм.
4. Передбачено режим «Для Розробників», що дає можливість легко протестувати розроблену програму, не вимагаючи підпису та розблокування як на Windows Mobile та iOS.
5. Широкий вибір бібліотек які спрощують розробку додатків.

Одним з найбільш важливих і значущих критеріїв для розробника у виборі ОС є її широка поширеність на пристроях різних фірм, наприклад Samsung, Motorola, Xiaomi, Huawei і т.д [4]. Таким чином розробник може досягти ширшої поширеності своєї програми.

У той час як при розробці на iOS такої можливості не надається, і розробка може вестись виключно для мобільних пристроїв корпорації Apple [5].

У той же час, як і будь-який інший продукт, Android має як свої переваги, так і недоліки [6]. Спиратися виключно на переваги було б не вірно з точки зору як розробника, так і цільового користувача.

Тому, варто виділити ряд недоліків Android:

1. Повна відкритість системи може збентежити користувача через невміння налаштування системи під себе.
2. Високе споживання заряду пристрою, що вимагає від розробника додаткової оптимізації програми, що розробляється.
3. Високе споживання оперативної пам'яті пристрою, що з кожною новою версією ОС тільки зростає

Інтерфейс користувача забезпечує зв'язок між комп'ютером і користувачем, обмін діями та відповідями на них [7]. Юзабіліті – це зручність використання

продукту, з якою продукт може використовуватися певним користувачем для досягнення поставленої мети з належною ефективністю, продуктивністю, задоволеністю, а також зручне та просте використання ресурсів [8].

2.3 Огляд середовищ програмування, що використовуються для створення програм під ОС Android

Одним з найбільш важливих елементів у процесі розробки програми є вибір правильного середовища розробки, що залежить не тільки від цільової платформи, але й рівня підготовки самого розробника. Безперечно, розробник, що володіє широким спектром знань у галузі програмування, зможе розробити додаток навіть у блокноті, але на відміну від розробки додатків для персональних комп'ютерів, при розробці мобільних додатків слід дотримуватись ряду умов, що висуває сама платформа. Таких як наприклад маніфест програми, в якому визначаються всі залежності коду, що розробляється [9]. З метою прискорення робочого процесу, яким би розумним не був розробник, слід все ж таки використовувати середовища розробки, що автоматично створюють як файли маніфесту, так і сам архів виконуваних файлів для обраної платформи [10]. Як, наприклад, для ОС Android таким є файл з розширенням .apk.

Нижче розглянемо кілька видів середовищ розробки мобільних програм під операційну систему Android.

2.3.1. Android Studio

Android Studio – середовище розробки під Android, яке базується на середовищі розробки IntelliJ IDEA. Створено Google. Вона надає включені в себе інструменти для розробки та налагодження [11]. Варто сказати, що в цьому середовищі враховані такі можливості, як:

1. Підтримка розробки програм за допомогою Gradle.
2. Спеціально створений під Android рефакторинг.
3. Інструменти для пошуку проблем із продуктивністю, зовнішнім виглядом, сумісністю версій та інших.

4. Створення макетів програм, використовуючи майстер швидких шаблонів.

5. Візуальний редактор, щоб переглянути зміни в реальному часі, без необхідності встановлення програми на пристрій або запуску програми в емуляторі.

6. Вбудований ADV емулятор смартфонів різних версій та виробників.

2.3.2. Eclipse

Eclipse - це досить популярне середовище розробки з відкритим кодом. Завдяки такому підходу це середовище розробки можна легко налаштувати «під себе». Середовище розробки Eclipse підтримується командою Eclipse Foundation та поширюється повністю безкоштовно [12]. У її основі лежить колись відоме середовище розробки IBM VisualAge. Це середовище було комерційним продуктом, але компанія IBM передала його код спільноті, яка створила та розвиває Eclipse.

Eclipse має такі плюси як [13]:

- Гнучкість у налаштуванні;
- Кросплатформеність;
- Інтерфейс

Також Eclipse має такі мінуси як:

- Плагіни. Немає гарантії, що плагін, розроблений самостійно чи кимось іншим, може стабільно працювати в Eclipse і не викликати проблем
- Неінтуїтивність. Інтерфейс та наявність функціоналу у Eclipse мінімальні. Функціонала не вистачає для роботи над великими проектами, тому його потрібно розширювати за допомогою плагінів.

2.3.2. IntelliJ IDEA

IntelliJ IDEA – це IDE, інтегроване середовище розробки для Java, JavaScript, Python та інших мов програмування від компанії JetBrains [14]. Відрізняється великим набором інструментів для рефакторингу та оптимізації коду.

До переваг можна віднести:

- Функціональність
- Великий вибір інструментів для роботи з кодом
- Ергономічність

- Комфортність

До недоліків в свою чергу можна віднести:

- Вибагливість до системних ресурсів
- Урізана функціональність безкоштовної версії

2.3.3. Висновок щодо вибору середовища розробки

Звичайно, були розглянуті далеко не всі середовища розробки під ОС Android.

Стисло можна виділити зведену таблицю різних середовищ розробки [15]:

Таблиця 2.2 – Найбільш популярні IDE для розробки під ОС Android

Номер	IDE
1	Eclipse
2	Android Studio
3	Visual Studio
4	IntelliJ IDEA

Однак детально враховувалися найбільш популярні, легко налагоджуються та спеціалізовані під Android середовища. Наприклад, такі середовища як Eclipse та IntelliJ IDEA спрямовані більше на розробку звичайних додатків, через що не містять безліч додаткових модулів, необхідних для початку розробки мобільного додатка [16].

Зробивши аналіз найбільш популярних середовищ розробки додатків, вибір упав на середовище Android Studio. IDE Android Studio обрана як офіційно визнане середовище розробки компанією Google, а також через те, що в цьому середовищі міститься величезна кількість модулів, корисних при розробці мобільного додатка [17], такі як вбудований перегляд виду створюваних вікон, а також засіб емуляції мобільних пристроїв для налагодження програми.

2.4 Огляд мов програмування для ОС Android

На початку проектування мобільного додатка перед розробником постає нелегкий вибір використовуваної ним мови програмування [18]. Адже при нестачі

знань та досвіду програмування якоюсь мовою, розробник може сильно затягнути процес розробки, бо йому доведеться підтягувати свої знання до потрібного рівня.

Слід також зазначити, що мова програмування має вибиратися з мети написання програмного продукту. Так, наприклад, якщо метою є створення легкої програми, то не варто вибирати важку мову. Однак, якщо програма знаходиться на середньому рівні, або навіть важкому, то варто вибирати мову того ж рівня.

На сьогоднішній день створено безліч версій таких фреймворків, що спрощують процес розробки мобільних програм розробникам, що не знають мову Java [19]. Варто виділити групу мов програмування у зведену таблицю:

Таблиця 2.3 – Найбільш популярні мови програмування для розробки під ОС Android

Номер	IDE
1	Java
2	C#
3	Python
4	Kotlin
5	Lua
6	React Native
7	C/C++

Далі, хотілося б розглянути основні та найпопулярніші на сьогоднішній день види мов програмування, які використовуються розробниками у всьому світі.

2.4.1. JavaScript

JavaScript – мова програмування, створена фірмою Netscape Communication Corporation. Ця мова не має жодного відношення до мови Java. Початкова назва мови звучала як LiveScript, однак, через величезну популярність мови Java, виходячи з комерційних міркувань мова була перейменована на JavaScript [20]. Важлива особливість JavaScript – об'єктна орієнтованість. Розробнику, який використовує цю мову доступні численні об'єкти, такі як документи, гіперпосилання, форми, кадри і

так далі. Об'єкти характеризуються описовою інформацією (властивостями) та можливими діями (методами) [21].

Плюси використання JavaScript:

1. Кросплатформеність.
2. Безліч фреймворків.
3. Велика кількість документації.

Мінуси використання JavaScript:

1. Додаток може бути лише гібридним.
2. Незважаючи на кросплатформенність, у кожної платформи свої особливості, і написати єдиний для всіх платформ код не вдасться.

2.4.2. C#

C# – об'єктно-орієнтована мова програмування, розроблена компанією Microsoft. В основному використовується для створення програм під операційну систему Windows [22]. Для розробки мобільних додатків з використанням C# був створений такий фреймворк і платформа, як Xamarin, що є платформою, для створення кросплатформових додатків під Android, iOS і Windows 10.

Плюси використання C#:

1. Кросплатформеність.
2. Близькість до нативного додатку.
3. Повний пакет інструментів розробки.
4. Нативний UI, що дозволяє створювати інтерфейс програми.

Мінуси використання C#:

1. Досить низький відсоток розробників, які використовують цю мову для розробки мобільних додатків.
2. Не підходить для програм з високопродуктивною графікою.
3. Великий розмір програм.
4. Складність інтеграції сторонніх ресурсів

2.4.3. Java

На сьогоднішній день, ця мова є однією з найпопулярніших мов програмування в усьому світі. Вперше, мова стала доступною ще в 1996 році в компанії Sun Microsystems, що була поглинена компанією Oracle [23].

Ключовою особливістю даної мови є її транслювання у спеціальний байт-код, що не залежить від платформи. Потім цей байт-код виконується спеціальною Java машиною. Подібна архітектура є доказом кросплатформенності та апаратної переносимості програм, написаних на Java.

На даний момент Java є офіційною мовою розробки нативних додатків під операційну систему Android.

Плюси використання Java:

1. Офіційна мова розробки під ОС Android.
2. Повна та доступна документація.
3. Великий відсоток розробників, які використовують цю мову.
4. Простота інтеграції сторонніх модулів.

Мінуси використання Java:

1. Android версія Java відрізняється від звичайної версії мови, через що потрібно додаткове вивчення безлічі модулів.
2. Властиві Java проблеми із займанням великого обсягу оперативної пам'яті.

2.4.4. Висновок щодо вибору мови програмування

Зробивши аналіз найбільш популярних мов програмування, вибір упав на мову Java.

Для досягнення поставленої задачі та мети, цей вибір є найбільш оптимальним через те, що Java має велику кількість документації, і дає можливість тонкого налаштування програми розробникам.

МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

3.1 Структурно функціональне моделювання

Для структурно функціонального моделювання додатку, було створено контекстну діаграму в нотації IDEF0.

IDEF0 - нотація графічного моделювання, що використовується для створення функціональної моделі [24].

Контекстна діаграма для мобільного додатку підтримки діяльності будівельної компанії представлена на рисунку 3.1

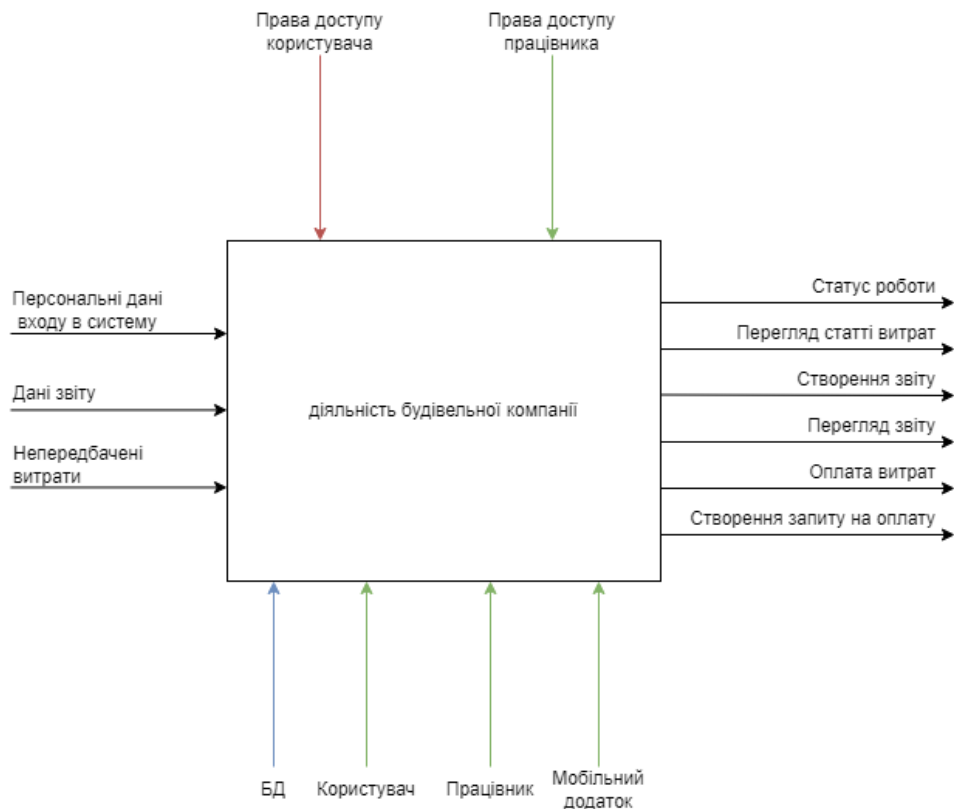


Рисунок 3.1 – Контекстна діаграма

У контекстній діаграмі був використаний такий перелік даних:

- На вході: Персональні дані входу в систему, дані звіту, непередбачені витрати;
- Механізмами виступають: БД, мобільний додаток, користувач, працівник;

- Управління: права доступу користувача/працівника;
- На виході: статус роботи, перегляд статті витрат, створення звіту, перегляд звіту, оплата витрат, створення запиту на оплату.

Наступним кроком для більш детального проектування додатку підтримки діяльності будівельної компанії була створена діаграма декомпозиції першого рівня [25]. Було виділено основних 5 під процесів:

- Авторизація;
- Робота на ділянці;
- Формування фінансових витрат
- Формування звітів

Діаграма декомпозиції представлена на рисунку 3.2.

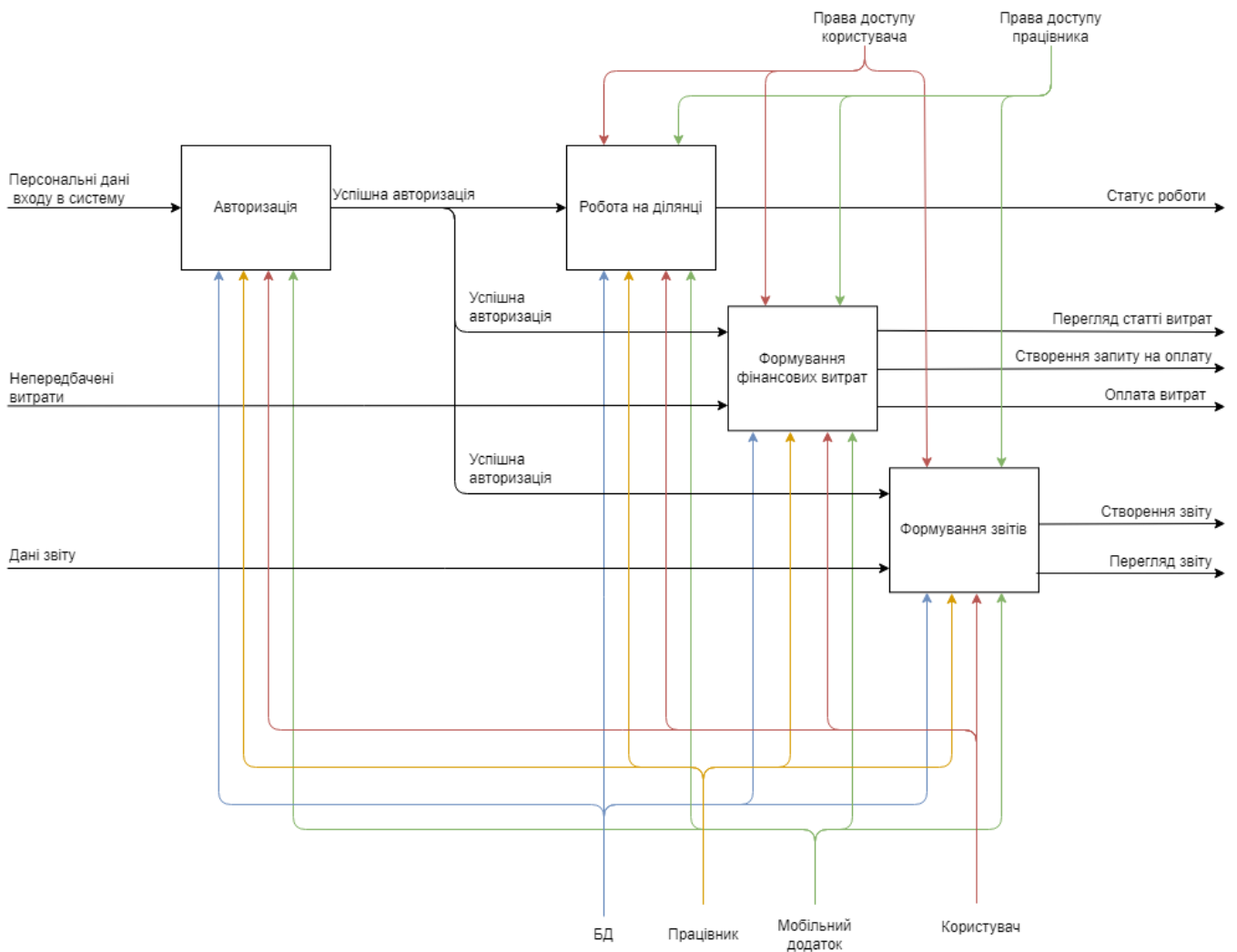


Рисунок 3.2 – Діаграма декомпозиції

3.2 Моделювання варіантів використання

Для того щоб представити функціональні вимоги до продукту, було прийнято рішення розробити діаграму варіантів використання(ВВ). На основі діаграми ВВ створюється прототип інтерфейсу [26].

Суть діаграми варіантів використання полягає в тому що система представляється у вигляді взаємодії акторів з системою.

Актори:

- Користувач – це користувач програми;
- Працівник;
- База даних – це сховище даних, яке буде зберігати інформацію;

Варіанти використання:

- Авторизація;
- Робота на ділянці;
- Перегляд статті витрат
- Створення запиту на оплату
- Оплата непередбачуваних витрат
- Створення звіту
- Перегляд звіту

Взаємодія акторів за варіантами використання:

- Користувач після авторизації може:
 - Переглядати чи ведуться роботи на ділянці;
 - Переглядати статтю витрат;
 - Оплачувати непередбачувані витрати;
 - Переглядати звіти працівників;
- Працівник після авторизації може:
 - Помічати чи ведуться роботи на ділянці;
 - Створювати звіт про виконану роботу;
 - Переглядати створені звіти;

- Створення запиту на оплату;
- Переглядати статті витрат;

Діаграма варіантів використання представлена на рисунку 3.3

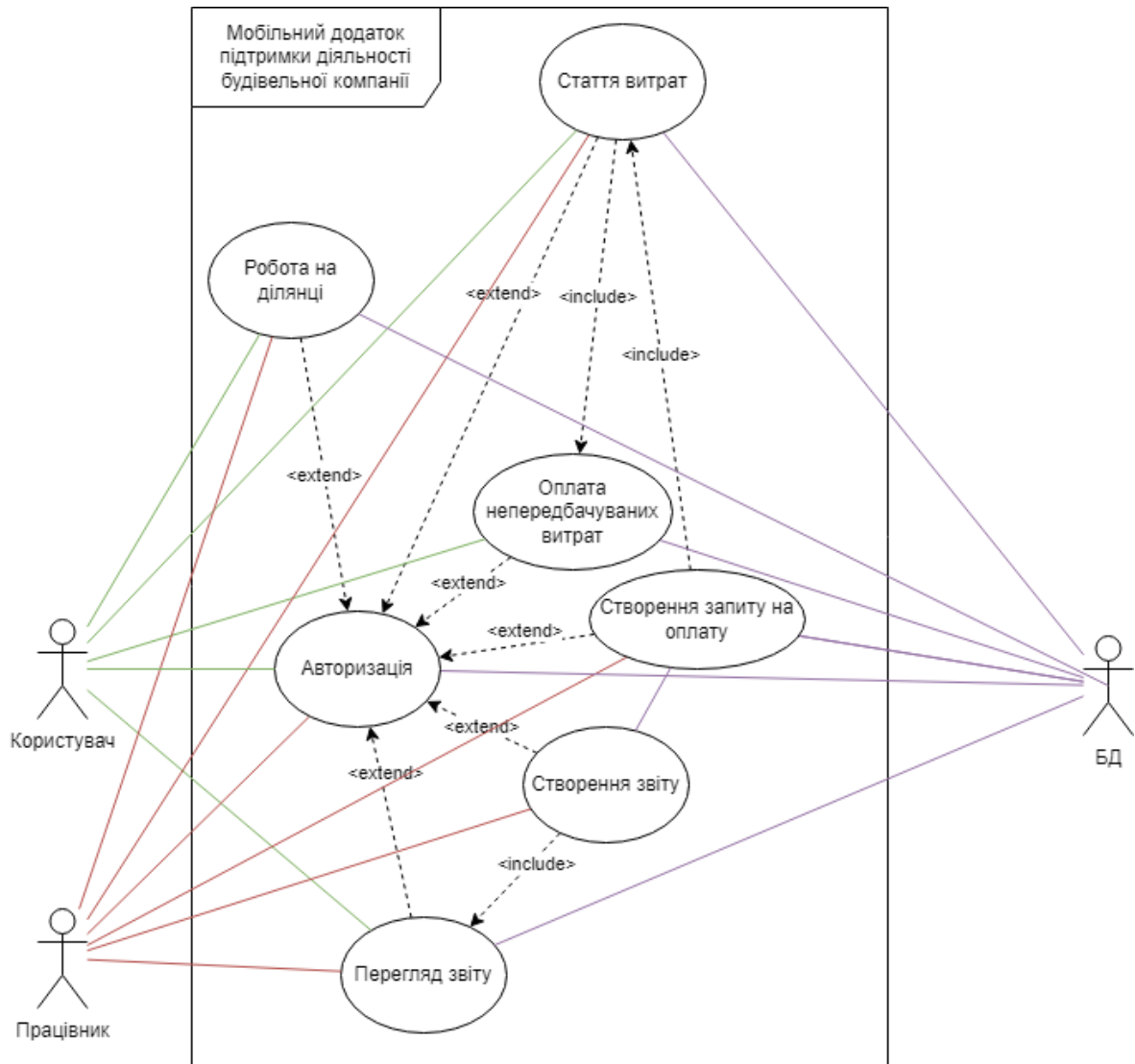


Рисунок 3.3 – Діаграма варіантів використання

3.3 Проектування бази даних

Щоб отримати графічне представлення бази даних мобільного додатку підтримки діяльності будівельної компанії була створена її логічна модель [27].

Логічну модель можна переглянути на рисунку 3.4

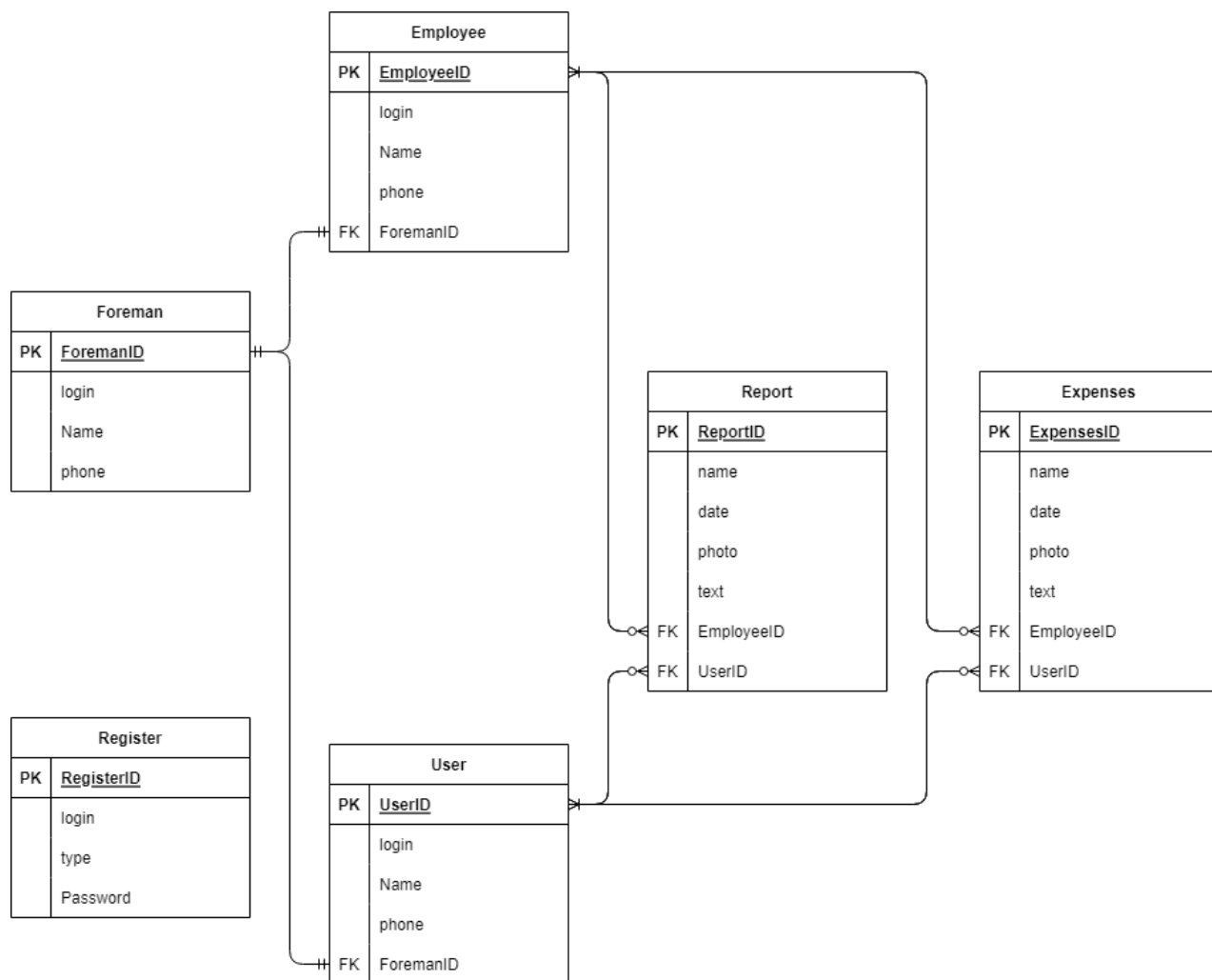


Рисунок 3.4 – Логічна модель

Опис сутностей:

User – містить особисті данні користувача

Опис атрибутів сутності User наведений в таблиці 3.1.

Таблиця 3.1 – Опис атрибутів сутності «User»

Атрибут	Опис
login	логін користувача
name	Ім'я користувача
phone	Контактний номер користувача
ForemanID	Унікальний номер виконробу

Foreman – містить основні данні виконробу

Опис атрибутів сутності Foreman наведений в таблиці 3.2.

Таблиця 3.2 – Опис атрибутів сутності «Foreman»

Атрибут	Опис
login	логін виконробу
name	ім'я виконробу
phone	контактний номер виконробу

Employee – містить основні данні працівника

Опис атрибутів сутності Employee наведений в таблиці 3.3.

Таблиця 3.3 – Опис атрибутів сутності «Employee»

Атрибут	Опис
login	логін працівника
name	ім'я працівника
phone	номер працівника
ForemanID	унікальний номер виконробу

Report – містить в собі данні звіту про виконану роботу

Опис атрибутів сутності Report наведений в таблиці 3.4.

Таблиця 3.4 – Опис атрибутів сутності «Report»

Атрибут	Опис
name	назва звіту
date	дата створення звіту
photo	фотографії виконаної роботи
text	опис фотографій і виконаної роботи

Продовження таблиці 3.4 – Опис атрибутів сутності «Report»

Атрибут	Опис
EmployeeID	унікальний номер працівника
UserID	унікальний номер користувача

Expenses - містить в собі данні звіту про затрачені кошти

Опис атрибутів сутності Expenses наведений в таблиці 3.5.

Таблиця 3.5 – Опис атрибутів сутності «Expenses»

Атрибут	Опис
name	назва звіту по витратам
date	дата створення звіту по витратам
photo	фото чеків
text	опис фотографій
EmployeeID	унікальний номер працівника
UserID	унікальний номер користувача

Register – містить логіни та паролі користувачів

Опис атрибутів сутності Register наведений в таблиці 3.6.

Таблиця 3.6 – Опис атрибутів сутності «Register»

Атрибут	Опис
login	логін користувача
type	тип юніту
password	пароль користувача

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

4.1 Архітектура мобільного додатку

Було створено модель архітектури мобільного додатку, на якій показані його компоненти та модулі. Мобільний додаток складається з серверної та клієнтської частини. До клієнтської частини відносяться такі модулі як інтерфейс мобільного додатку, модуль головного меню, модуль перегляду поточних робіт на ділянці, модуль перегляду або створення звітів, модуль перегляду або створення або оплати витрат, та модулі збору інформації. До серверної відносяться модулі обробки які отримали інформацію від модулів збору інформації з клієнтської частини завдяки проведеним маніпуляціям користувача.

Нижче можна переглянути архітектуру мобільного додатку на рисунку 4.1

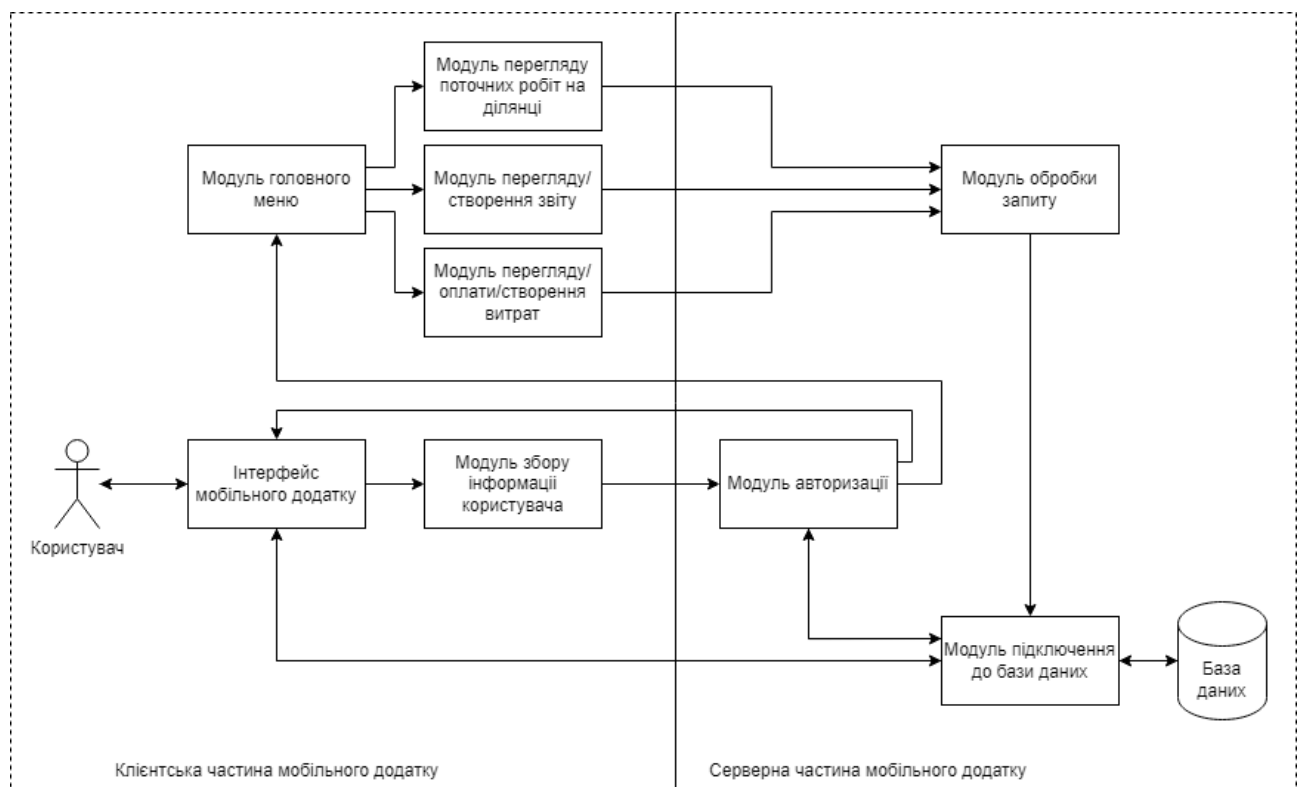


Рисунок 4.1 – Архітектура мобільного додатку

4.2 Програмна реалізація мобільного додатку

Програмна реалізація мобільного додатку для підтримки діяльності будівельної компанії відбувалася у середовищі Android Studio. В якості мови програмування була обрана Java. Структура мобільного додатку представлена на рисунку 4.2

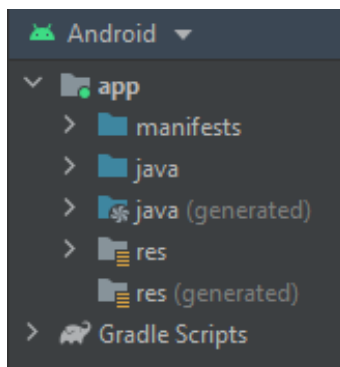


Рисунок 4.2 – Структура файлів мобільного додатку

Згідно показаної на рисунку 4.2 структури мобільного додатку, було зроблено опис основних папок (табл. 4.1)

Таблиця 4.1 – Опис структури мобільного додатку

№	Папка	Короткий опис вмісту папки
1	«manifest»	Містить в собі файл який дозволяє описати функціональні можливості і вимоги додатку до Android
2	«java»	Містить усі файли класів та активностей
3	«res»	Містить усі файли ресурсів та розмітки активностей

Опис розроблених сторінок мобільного додатку представлено в таблиці 4.2

Таблиця 4.2 – Опис розроблених сторінок мобільного додатку

№	Назва сторінки	Короткий опис сторінки
1	Сторінка авторизації	Дана сторінка містить: <ul style="list-style-type: none"> – поле для введення логіну; – поле для введення пароля; – кнопку увійти.

Продовження таблиці 4.2 – Опис розроблених сторінок мобільного додатку

№	Назва сторінки	Короткий опис сторінки
2	Головне меню	<p>Дана сторінка містить:</p> <ul style="list-style-type: none"> – посилання до сторінки поточних робіт на ділянці; – посилання до сторінки архіву статті витрат; – посилання до сторінки архіву звітів; – посилання до сторінки неоплачених витрат; – кнопку виходу з системи.
3	Сторінка поточних робіт на ділянці	<p>Дана сторінка містить:</p> <ul style="list-style-type: none"> – інформацію чи ведуться роботи на ділянці.
4	Сторінки архіву статті витрат	<p>Дана сторінка містить:</p> <ul style="list-style-type: none"> – інформацію про оплачені користувачем непередбачувані витрати;
5	Сторінки архіву звітів	<p>Дана сторінка містить:</p> <ul style="list-style-type: none"> – список створених робітниками звітів; – кнопку для працівників, для створення нового звіту;
6	Сторінка детального перегляду звіту	<p>Дана сторінка містить:</p> <ul style="list-style-type: none"> – поле з назвою звіту; – поле з коротким описом від працівників; – прикріплену робітником фотографію.
7	Сторінка створення звіту	<p>Дана сторінка містить:</p> <ul style="list-style-type: none"> – поле для створення назви звіту; – поле для створення опису теми звіту; – кнопку додання фотографії до звіту

Продовження таблиці 4.2 – Опис розроблених сторінок мобільного додатку

№	Назва сторінки	Короткий опис сторінки
8	Сторінка неоплачених витрат	Дана сторінка містить: <ul style="list-style-type: none"> – поле з назвою товару; – поле з ціною за товар; – кнопка для працівників, для створення нового запиту на оплату.
9	Сторінка створення запиту на оплату	Дана сторінка містить: <ul style="list-style-type: none"> – поле для створення назви товару; – поле для створення ціни за товар; – кнопка додання фотографії чеку товару.

4.2 Використання мобільного додатку користувачем

Після запуску мобільного додатку користувач потрапляє до вікна авторизації (рис 4.3).

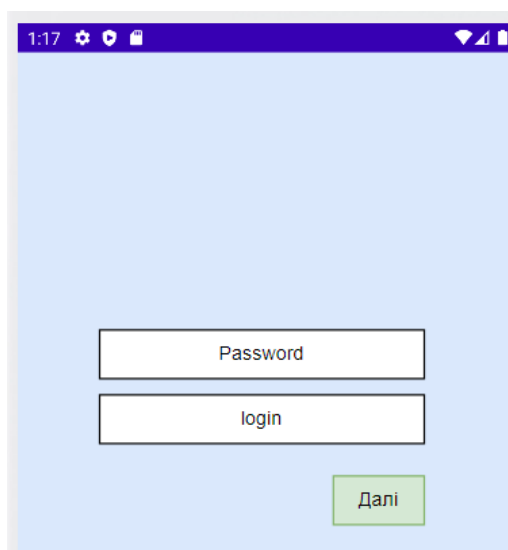


Рисунок 4.3 – Вікно авторизації у системі

Після успішної авторизації користувач потрапляє на головну сторінку додатку де в нього є можливість обрати одну з чотирьох функцій додатку: «роботи на ділянці», «стаття витрат», «звіти», «оплата» (рис 4.4).

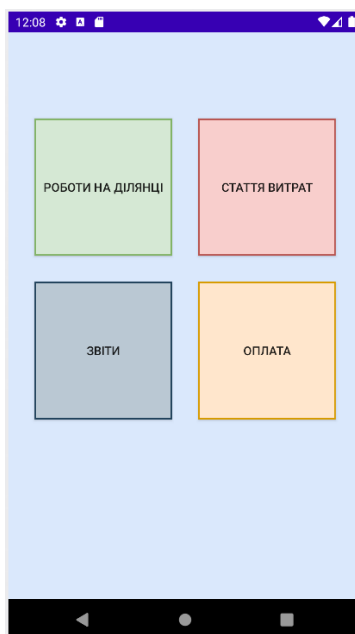


Рисунок 4.4 – Головна сторінка додатку

У вікні «роботи на ділянці» можна переглянути чи ведуться роботи на тій чи іншій ділянці (рис 4.5).

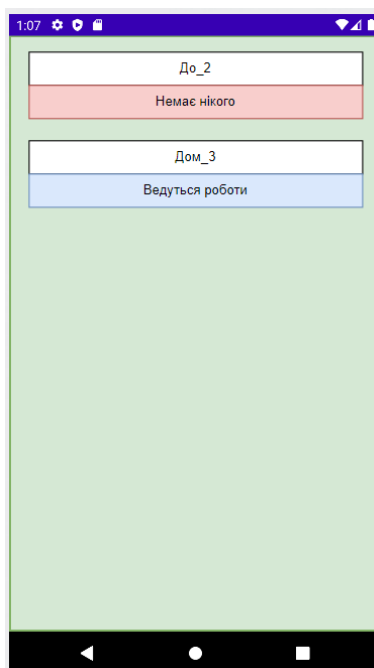


Рисунок 4.5 – Сторінка функції «робота на ділянці»

У вікні «стаття витрат» можна переглянути все оплачені користувачем товари (рис 4.6).

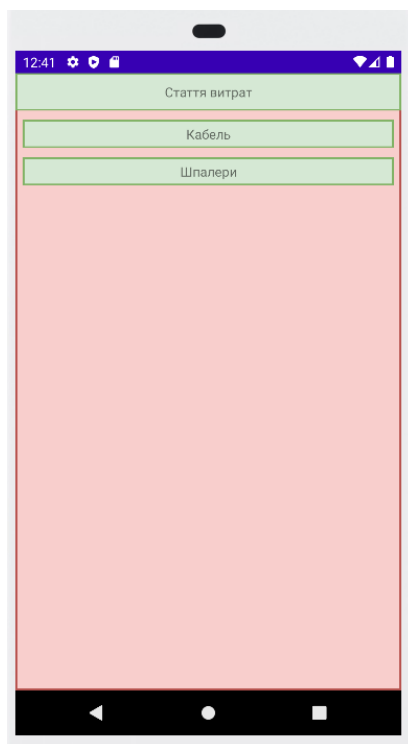


Рисунок 4.6 – Сторінка статті витрат

У вікні «звіти» можна переглянути створені робітниками звіти про виконану роботу на ділянці (рис 4.7).

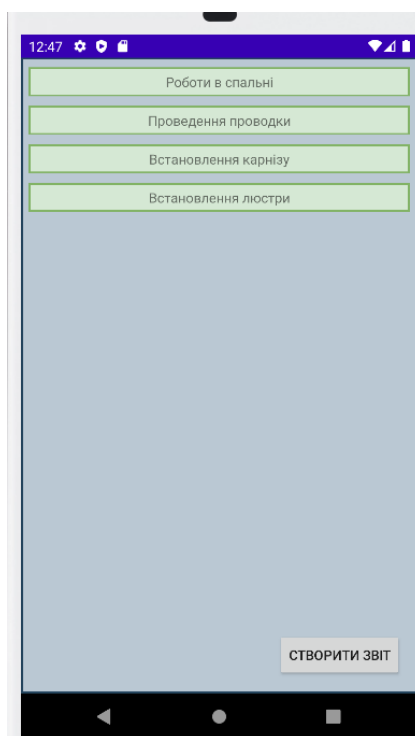


Рисунок 4.7 – Сторінка звітів

Також у робітника є можливість в вкладці «звіти» створити звіт про виконану роботу натиснувши на кнопку в правому нижньому кутку екрану (рис 4.8).

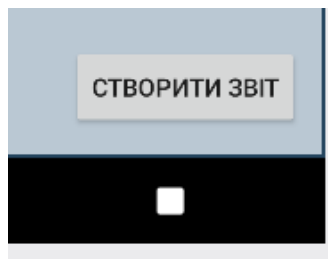


Рисунок 4.8 – Кнопка створення звіту

У вікні зі створенням звіту, можна написати назву самого звіту, написати коментар про виконану роботу та прикріпити фотографію до звіту (рис 4.9).



Рисунок 4.9 – Сторінка створення звіту

Є можливість переглядати розширений звіт робітників (рис 4.10)

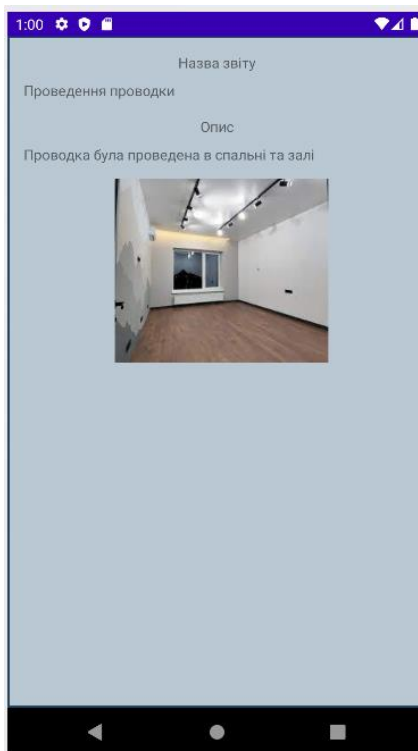


Рисунок 4.10 – Розширений звіт

У вікні «оплата» користувач може оплатити куплений робітниками товар, який їм не вистачало для продовження робіт (рис 4.11).

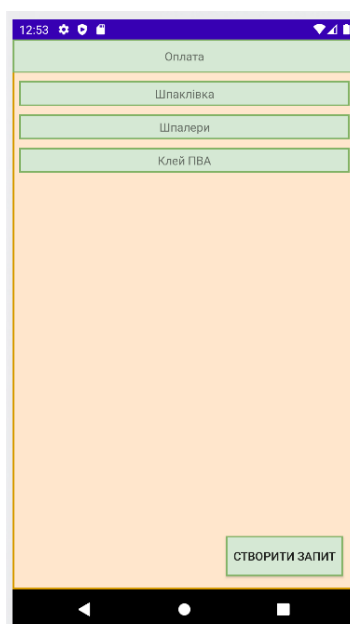


Рисунок 4.11 – Сторінка функції «оплата»

У вікні зі створенням запиту на оплату товару, можна написати назву самого товару, написати його ціну та прикріпити фотографію чеку цього товару (рис 4.12).



Рисунок 4.12 – Сторінка створення запиту на оплату купленого товару

З сторінки функції оплата є можливість переглядати розширену версію товарів які були куплені робітниками для ремонту (рис 4.13)

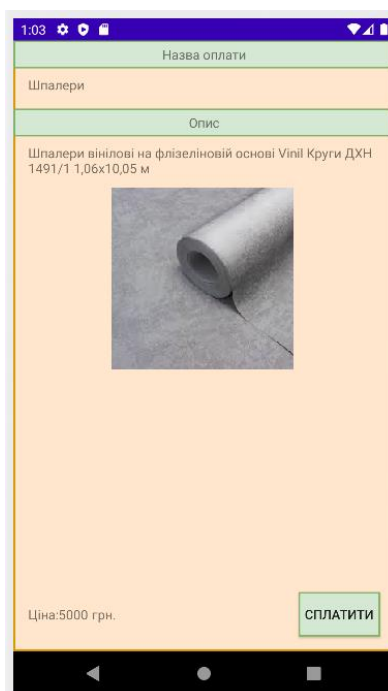


Рисунок 4.13 – Розширена версія товарів

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи магістра була створена WBS-структура для того щоб зрозуміти яких конкретних результатів треба досягти, щоб досягти цілі проекту. Була створена OBS-структура, розроблена діаграма Ганта та розрахована матриця ризиків проекту.

Був проведений аналіз предметної області, оглянуті існуючі мобільні додатки аналоги.

На основі цих даних були сформовані мета та задачі проекту.

Крім цього, проведено аналіз найбільш популярних операційних систем, мов програмування та середовищ розробки.

В якості операційного середовища для розробки мобільного додатка, був зроблений вибір на операційну систему Android, зважаючи на її найбільшу популярність і доступність. Вибраною мовою програмування стала офіційна мова програмування під операційну систему Android – Java. А середовищем розробки обрано Android Studio.

Також, були поставлені вимоги до програми, що розробляється, такі як вимоги до інтерфейсу користувача і до виконуваних функцій.

У ході роботи було зроблено структурно функціональне моделювання зі створення контекстної діаграми в нотації IDEF0. Також була створена модель варіантів використання, і створена модель бази даних.

Результатом проведеної роботи є реалізований додаток підтримки діяльності будівельної компанії. Розроблений додаток облегшить користувачам задачу по отриманню інформації про поточні ремонтні роботи, без потреби безпосередньо знаходитися на ділянці де ведуться роботи. Також облегшить оплату непередбачених витрат на будівництво. Додаток дозволить переглядати в звіті фото та коментарі робітників про виконану роботу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Галузі промисловості [Електронний ресурс] – режим доступу: <https://ips.ligazakon.net/document/F980188?an=36>
2. Харченко В. П., Знаковська Є. А., Бородин В. А. Операційні системи та системи програмування навч. посіб. Київ: НАУ 2012 – 365 с.
3. Android чи iOS [Електронний ресурс] – режим доступу: <https://gre4ka.info/zhyttia/70300-smartfon-z-android-chy-ios-i-akyi-krashche-obraty>
4. Зайцев В. Г. Комп'ютерні системи реального часу навч. посіб. Київ: КПІ, 2019 – 162 с.
5. Жураковський Б.Ю. Технології інтернету речей навч. посіб. Київ: КПІ, 2021 – 271 с.
6. Переваги та недоліки Android [Електронний ресурс] – режим доступу: <https://rivnepost.rv.ua/news/perevahi-ta-nedoliki-suchasnikh-smartfoniv-na-bazi-android>
7. Проектування інтерфейсу користувача [Електронний ресурс] – режим доступу: https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/282171/index.html
8. Сердюк С.М. Людино-машинна взаємодія конспект лекцій. Запоріжжя: ЗНТУ, 2014. – 58 с.
9. Васюта В.В. Технології розробки програмного забезпечення навч. посіб. Полтава: ПолтНТУ, 2017. – 218 с.
10. Збірка Java програм [Електронний ресурс] – режим доступу: <https://jak.koshachek.com/articles/zbirka-i-vikonannja-java-program.html>
11. Android Studio: переваги та особливості [Електронний ресурс] – режим доступу: <https://qagroup.com.ua/publications/android-studio-perevagy-ta-osoblyvosti/>
12. Початок роботи з платформою Eclipse [Електронний ресурс] – режим доступу: <https://jak.koshachek.com/articles/pochatok-roboti-z-platformuju-eclipse.html>

13. Eclipse переваги та недоліки [Електронний ресурс] – режим доступу: <https://senior.ua/articles/ide-eclipse-za--proti-vd-provdnih-programstv>
14. Про JetBrains [Електронний ресурс] – режим доступу: <https://itc.ua/ua/novini/jetbrains-rozrobnik-movi-programuvannya-kotlin-prizupinyaye-prodazhiv-v-rosiyi-ta-bilorusi/>
15. Мова програмування для створення Андроїд – додатку [Електронний ресурс] – режим доступу: <https://blogchain.com.ua/iak-vibrati-movy-programuvannia-dlia-stvorennia-android-dodatki/>
16. Шевчук І.Б. Об'єктно-орієнтоване програмування конспект лекцій. Львів: ЛНУ, 2020. – 32 с.
17. Табунщик Г.В., Каплієнко Т.І., Петрова О.А. Проектування та моделювання програмного забезпечення сучасних інформаційних систем навч. посіб. Запоріжжя: Дике Поле, 2016 – 250 с.
18. Недашківський О.М.. Планування та проектування інформаційних систем. – Київ, 2014. – 215 с.
19. Найкращі мови програмування для розробки програм для Android [Електронний ресурс] – режим доступу: <https://uaspectr.com/2022/01/20/najkrashhi-movy-programuvannya-dlya-rozrobky-android/>
21. Про JavaScript [Електронний ресурс] – режим доступу: <https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti>
20. Про JavaScript [Електронний ресурс] – режим доступу: <https://dou.ua/forums/topic/35184/>
22. Настенко Д. В. Навчальний посібник С# навч. посіб. Київ; НТУУ 2016 – 76 с.
23. Гарнавський Ю.А. Java Програмування навч. посіб. Київ; КПІ 2021 – 95 с.
24. Іващук В.В. Автоматизація бізнес процесів навч.-метод. посіб. Київ; НУХТ 2017 – 165 с.
25. Табунщик Г. В. Проектування, моделювання та аналіз інформаційних систем: Навчальний посібник / Г.В. Табунщик, Р.К. Кудерметов, А. В. Притула. – Запоріжжя : ЗНТУ, 2011. – 292 с.

26. Діаграма варіантів використання [Електронний ресурс] – режим доступу:
http://iwanoff.inf.ua/oop_kn/LabTraining05.html

27. Гайна Г.А. Основи проектування баз даних навч. посіб. Київ: КНУБА, 2017
– 204 с.

ДОДАТОК А

ПЛАНУВАННЯ РОБІТ

S – конкретність – ціллю проекту є розробка мобільного додатку підтримки діяльності будівельної компанії.

M – вимірюваність – якість роботи проекту буде оцінена дипломною комісією.

A – досяжність – ціль є досяжною оскільки розробник має необхідні уміння для виконання проекту.

R – реалістичність – для досягнення цілей проекту наявні всі необхідні ресурси та знання.

T – обмеженість в часі – проект повинен бути виконаний до 12.12.2022.

Для організації робіт під час виконання проекту була створена WBS діаграма.

WBS діаграма представлена на рисунку А.1.

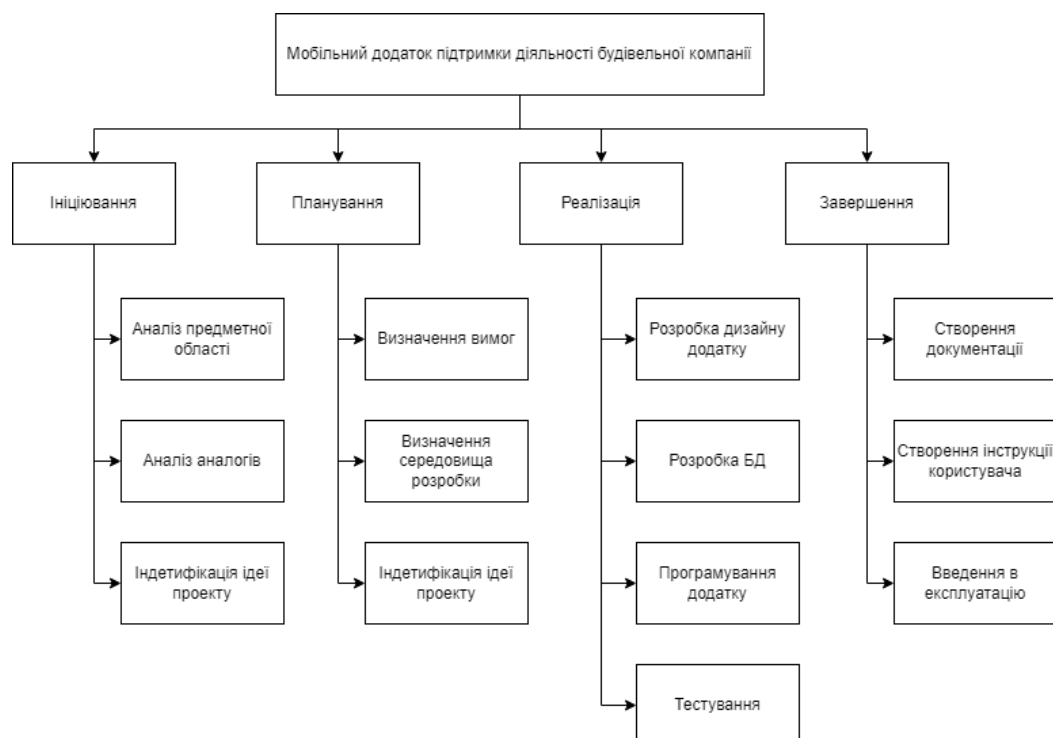


Рисунок А.1 – WBS діаграма проекту

На наступному етапі планування проекту була створена OBS Діаграма.

Учасниками розробки даного проекту є:

- Нагорний В.В. – куратор проекту;
- Семеняко С.О. – розробник.

OBS діаграма представлена на рисунку А.2.

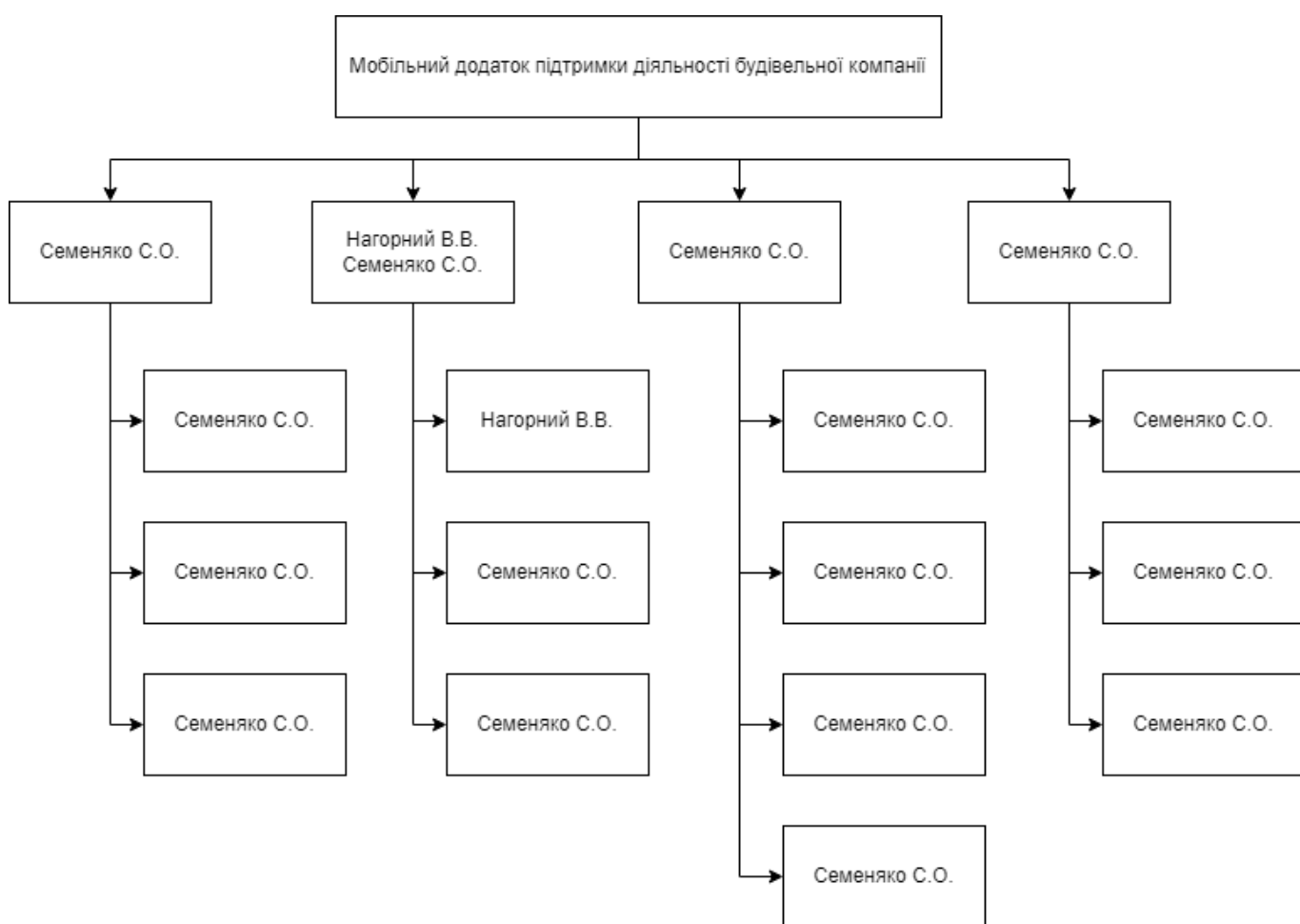


Рисунок А.2 – OBS-структура проекту

Побудова календарного графіку виконання ІТ – проекту

Робота над проектом велась починаючи з 03.10.2022 до 10.12.2022, таким чином робота йшла протягом 10 тижнів.

Календарний графік виконання проекту представлений на діаграмі Ганта, яка представлена на рисунку А.3.

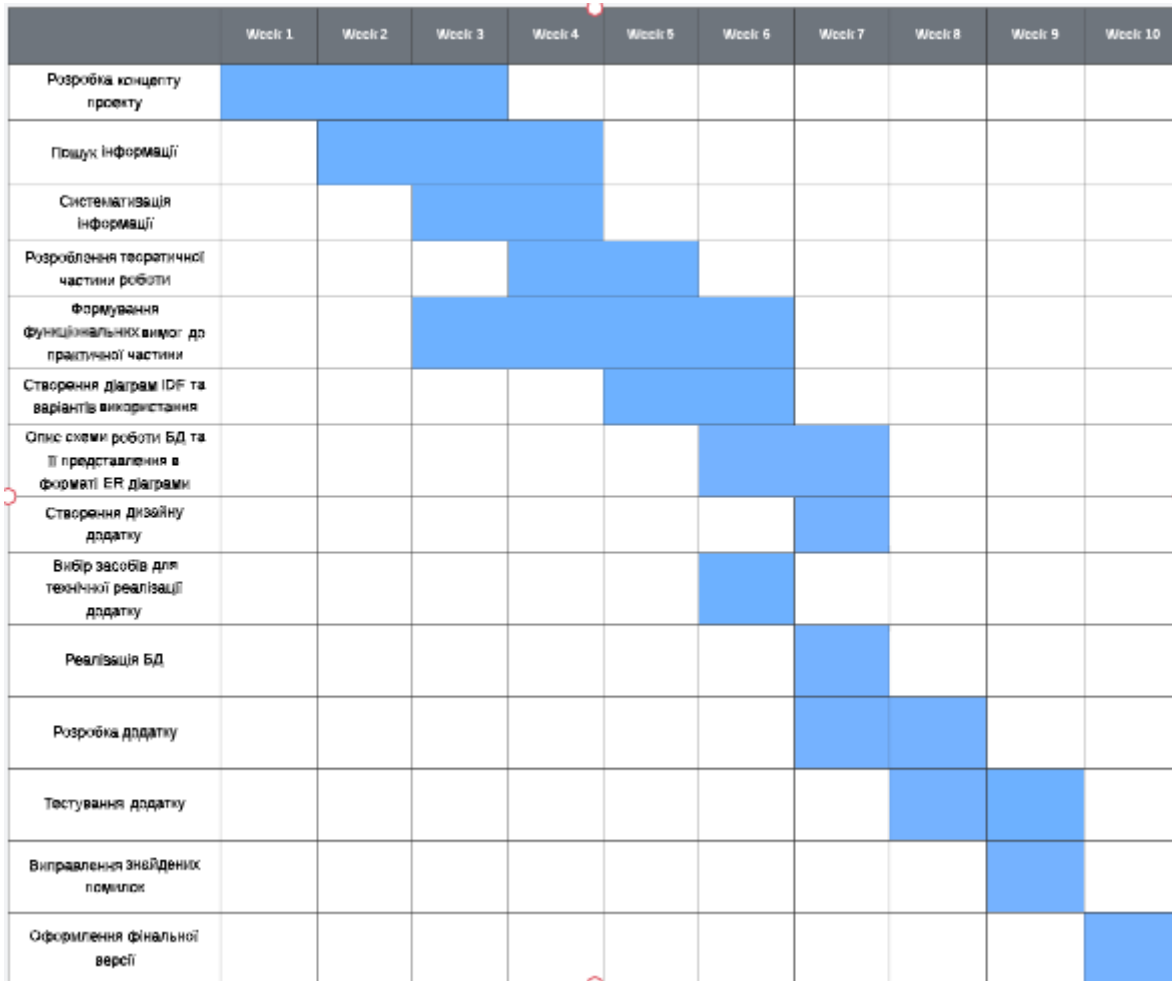


Рисунок А.3. – Календарний графік виконання проекту

Планування ризиків проекту

Ризики, які виникають в ході реалізації проекту представлені на таблиці А.4

Таблиця А.4. – Ризики проекту

Номер	Ризик	Вірогідність	Шкода
R1	Використання неактуальних даних	2	3
R2	Зміна функціональних вимог	2	2
R3	Інфраструктурні проблеми	4	4
R4	Втрата напрацювань через технічні неполадки	2	5

Продовження таблиці А.4 – Ризики проекту

R5	Оновлення бібліотек	3	1
R6	Затирання даних	1	3

Для кращої візуалізації, створимо матрицю, ця матриця дає певну оцінку кожному ризику, відповідно до відношення Вірогідність – Шкода. Матриця представлена на рисунку А.4

	Вірогідність	5	5	10	15	20	25
R3		4	4	8	12	16	20
R5		3	3	6	9	12	15
R1, R2, R4		2	2	4	6	8	10
R6		1	1	2	3	4	5
			1	2	3	4	5
		Шкода					
		R5	R2	R1, R6	R3	R4	

Рисунок А.4. - Матриця Вірогідність - Шкода

ДОДАТОК Б.**ЛІСТИНГ ПРОГРАМНОГО КОДУ**

cost_item_Activity.java

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import com.example.myapplication.adapter.MainAdapter_p;
import com.example.myapplication.db.MyDbManager;

public class cost_item_Activity extends AppCompatActivity {
    private MyDbManager myDbManager;
    private RecyclerView rcView;
    private MainAdapter_p mainAdapter_p;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cost_item);
        init();
    }

    private void init(){
        myDbManager = new MyDbManager(this);
        rcView = findViewById(R.id.rcView_p);
        mainAdapter_p = new MainAdapter_p(this);
    }
}
```

```

        rcView.setLayoutManager(new LinearLayoutManager(this));
        rcView.setAdapter(mainAdapter_p);
    }

    @Override
    protected void onResume() {
        super.onResume();
        myDbManager.openDb();
        mainAdapter_p.updateAdapter(myDbManager.getFromDb_p_r());
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        myDbManager.closeDb();
    }
}

```

Extended_payment.java

```

package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.example.myapplication.adapter.ListItem;
import com.example.myapplication.db.MyConstants;
import com.example.myapplication.db.MyDbManager;

public class extended_payment extends AppCompatActivity {

```



```
private TextView text_title, text_desc, text_price;
private ImageView imageView_ex;
private Button button;

private MyDbManager myDbManager;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_extended_payment);
    init();
    getMyIntents();
}

private void init(){
    myDbManager = new MyDbManager(this);
    text_title = findViewById(R.id.text_title);
    text_desc = findViewById(R.id.text_desc);
    text_price = findViewById(R.id.text_price);
    imageView_ex = findViewById(R.id.imageView_ex);
    button = findViewById(R.id.button);
}

private void getMyIntents(){
    Intent i = getIntent();
    if(i != null)
    {
        ListItem item = (ListItem)i.getSerializableExtra(MyConstants.LIST_ITEM_INTENT);
        text_title.setText(item.getTitle());
        text_desc.setText(item.getDesc());
        text_price.setText(item.getPrice());
        if("0".equals(item.getCheck_payment()))
            button.setVisibility(View.VISIBLE);
    }
}
```

```

        if(!item.getUri().equals("empty")){
            imView_ex.setVisibility(View.VISIBLE);
        }
    }
}

@Override
protected void onResume() {
    super.onResume();
    myDbManager.openDb();
}

public void onClick_pay(View view) {
    Intent i = getIntent();
    if(i != null) {
        ListItem item = (ListItem) i.getSerializableExtra(MyConstants.LIST_ITEM_INTENT);
        Log.d("Mylog","Title:" + item.getTitle() + " Price:" + item.getPrice() + "
Desc:" + item.getDesc() + " Uri:" + item.getUri() + " Check: 1" + " Id:" + item.getId());
        myDbManager.update(item.getTitle(), item.getPrice(), item.getDesc(),
item.getUri(),"1", item.getId());
        finish();
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    myDbManager.closeDb();
}
}

```

MainActivity.java

```

package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

```

```
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void onClick_works_on_the_site(View view) {
        Intent i = new Intent(MainActivity.this,works_on_the_site_Activity.class);
        startActivity(i);
    }

    public void onClick_cost_item(View view) {
        Intent i = new Intent(MainActivity.this,cost_item_Activity.class);
        startActivity(i);
    }

    public void onClick_reports(View view) {
        Intent i = new Intent(MainActivity.this,reports_Activity.class);
        startActivity(i);
    }

    public void onClick_payment(View view) {
        Intent i = new Intent(MainActivity.this,payment_Activity.class);
```

```
        startActivity(i);
    }
}

payment_Activity
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void onClick_works_on_the_site(View view) {
        Intent i = new Intent(MainActivity.this,works_on_the_site_Activity.class);
        startActivity(i);
    }

    public void onClick_cost_item(View view) {
        Intent i = new Intent(MainActivity.this,cost_item_Activity.class);
        startActivity(i);
    }
}
```

```
public void onClick_reports(View view) {
    Intent i = new Intent(MainActivity.this,reports_Activity.class);
    startActivity(i);
}
```

```
public void onClick_payment(View view) {
    Intent i = new Intent(MainActivity.this,payment_Activity.class);
    startActivity(i);
}
```

```
}
```

payment_create_Activity

```
package com.example.myapplication;
```

```
import androidx.annotation.Nullable;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.constraintlayout.widget.ConstraintLayout;
```

```
import android.content.Intent;
```

```
import android.net.Uri;
```

```
import android.os.Bundle;
```

```
import android.util.Log;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
import android.widget.ImageView;
```

```
import android.widget.Toast;
```

```
import com.example.myapplication.db.MyDbManager;
```

```
public class payment_create_Activity extends AppCompatActivity {
    private final int PICK_IMAGE_CODE = 123;
    private ImageView imageView;
    private ConstraintLayout photo_const;
    private Button onClick_add_button;
```

```

private MyDbManager myDbManager;
private EditText edTitle_p, edDesc_p, price;
private String tempUri = "empty";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_payment_create);
    init();
}

@Override
protected void onResume() {
    super.onResume();
    myDbManager.openDb();
}

//@SuppressLint("WrongConstant")
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == RESULT_OK && requestCode == PICK_IMAGE_CODE && data!=null){
        tempUri = data.getData().toString();
        Uri uri = Uri.parse(tempUri);
        //Uri uri =
Uri.parse("content://com.android.providers.downloads.documents/document/msf%3A34");
        Log.d("Mylog", "Uri:" + uri);
        Log.d("Mylog", "TempUri:" + tempUri);
        imageView.setImageURI(uri);
    }
}

private void init(){
    myDbManager = new MyDbManager(this);
}

```

```

edTitle_p = findViewById(R.id.edTitle_p);
edDesc_p = findViewById(R.id.edDesc_p);
price = findViewById(R.id.price);
photo_const = findViewById(R.id.photo_const_p);
onClick_add_button = findViewById(R.id.add_button_p);
imageView = findViewById(R.id.imageView_p);
}

```

```

public void onClick_add_button(View view) {
    photo_const.setVisibility(View.VISIBLE);
    onClick_add_button.setVisibility(View.INVISIBLE);
    Intent chooser = new Intent(Intent.ACTION_OPEN_DOCUMENT);
    chooser.setType("image/*");
    startActivityForResult(chooser, PICK_IMAGE_CODE);
}

```

```

public void onClickSave(View view) {
    String title_ = edTitle_p.getText().toString();
    String disc_ = edDesc_p.getText().toString();
    String price_ = price.getText().toString();

    if(title_.equals("") || disc_.equals("") || price_.equals("")){
        Toast.makeText(this, "Заголовок або ціна або опис порожній",
Toast.LENGTH_SHORT).show();
    }
    else {
        myDbManager.insertToDb(title_, price_, disc_, tempUri, "0");
        Toast.makeText(this, "Дані збережено", Toast.LENGTH_SHORT).show();
        finish();
    }
}
}

```

```

public void onClick_del_button(View view) {
    imageView.setImageResource(0);
    tempUri = "empty";
}

```

```
        photo_const.setVisibility(View.INVISIBLE);
        onClick_add_button.setVisibility(View.VISIBLE);
    }
```

```
@Override
protected void onDestroy() {
    super.onDestroy();
    myDbManager.closeDb();
}
```

```
}
```

report_create_Activity

```
package com.example.myapplication;
```

```
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;
```

```
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
```

```
import com.example.myapplication.db.MyDbManager;
```

```
public class report_create_Activity extends AppCompatActivity {
    private final int PICK_IMAGE_CODE = 123;
    private ImageView imageView;
    private ConstraintLayout photo_const;
```



```

private Button onClick_add_button;
private MyDbManager myDbManager;
private EditText edTitle, edDisc;
private String tempUri = "empty";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_report_create);
    init();
}

@Override
protected void onResume() {
    super.onResume();
    myDbManager.openDb();
}

//@SuppressWarnings("WrongConstant")
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == RESULT_OK && requestCode == PICK_IMAGE_CODE && data!=null){
        tempUri = data.getData().toString();
        Uri uri = Uri.parse(tempUri);
        //Uri uri =
Uri.parse("content://com.android.providers.downloads.documents/document/msf%3A34");
        Log.d("Mylog", "Uri:" + uri);
        Log.d("Mylog", "TempUri:" + tempUri);
        imageView.setImageURI(uri);
    }
}

private void init(){

```

```

myDbManager = new MyDbManager(this);
edTitle = findViewById(R.id.edTitle_p);
edDisc = findViewById(R.id.edDesc_p);
photo_const = findViewById(R.id.photo_const_p);
onClick_add_button = findViewById(R.id.add_button_p);
imageView = findViewById(R.id.imageView_p);
}

public void onClick_add_button(View view) {
    photo_const.setVisibility(View.VISIBLE);
    onClick_add_button.setVisibility(View.INVISIBLE);
    Intent chooser = new Intent(Intent.ACTION_OPEN_DOCUMENT);
    chooser.setType("image/*");
    startActivityForResult(chooser, PICK_IMAGE_CODE);
}

public void onClickSave(View view) {
    String title_ = edTitle.getText().toString();
    String disc_ = edDisc.getText().toString();
    if(title_.equals("") || disc_.equals("")){
        Toast.makeText(this, "Заголовок або Зміст порожній", Toast.LENGTH_SHORT).show();
    }
    else {
        myDbManager.insertToDb(title_, disc_, tempUri);
        Toast.makeText(this, "Дані збережено", Toast.LENGTH_SHORT).show();
        finish();
    }
}

public void onClick_del_button(View view) {
    imageView.setImageResource(0);
    tempUri = "empty";
    photo_const.setVisibility(View.INVISIBLE);
    onClick_add_button.setVisibility(View.VISIBLE);
}

```

```
@Override
protected void onDestroy() {
    super.onDestroy();
    myDbManager.closeDb();
}
}
```

reports_Activity

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import com.example.myapplication.adapter.MainAdapter;
import com.example.myapplication.db.MyDbManager;

public class reports_Activity extends AppCompatActivity {
    private MyDbManager myDbManager;
    private RecyclerView rcView;
    private MainAdapter mainAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_reports);
        init();
    }

    private void init(){
        myDbManager = new MyDbManager(this);
    }
}
```

```
rcView = findViewById(R.id.rcView_p);
mainAdapter = new MainAdapter(this);
rcView.setLayoutManager(new LinearLayoutManager(this));
rcView.setAdapter(mainAdapter); //Часть 2 16:07
}
```

```
@Override
```

```
protected void onResume() {
    super.onResume();
    myDbManager.openDb();
    mainAdapter.updateAdapter(myDbManager.getFromDb());
}
```

```
public void onClick_rep_create(View view) {
    Intent i = new Intent(reports_Activity.this,report_create_Activity.class);
    startActivity(i);
}
```

```
@Override
```

```
protected void onDestroy() {
    super.onDestroy();
    myDbManager.closeDb();
}
```

```
}
```

MainAdapter.java

```
package com.example.myapplication.adapter;
```

```
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
```

```
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.myapplication.R;
import com.example.myapplication.db.MyConstants;
import com.example.myapplication.extended_report;
import com.example.myapplication.report_create_Activity;

import java.util.ArrayList;
import java.util.List;

public class MainAdapter extends RecyclerView.Adapter<MainAdapter.MyViewHolder> {
    private Context context;
    private List<ListItem> mainArray;

    public MainAdapter(Context context) {
        this.context = context;
        mainArray = new ArrayList<>();
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.item_list_layout, parent,
false);
        return new MyViewHolder(view, context, mainArray);
    }

    @Override
    public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
        holder.setData(mainArray.get(position).getTitle());
    }

    @Override
```

```

public int getItemCount() {
    return mainArray.size();
}

```

```

static class MyViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {

```

```

    private TextView tvTitle;
    private Context context;
    private List<ListItem> mainArray;

```

```

    public MyViewHolder(@NonNull View itemView, Context context, List<ListItem>
mainArray) {

```

```

        super(itemView);
        this.context = context;
        this.mainArray = mainArray;
        tvTitle = itemView.findViewById(R.id.tvTitle);
        itemView.setOnClickListener(this);

```

```

    }

```

```

    public void setData(String title){
        tvTitle.setText(title);
    }

```

```

@Override

```

```

    public void onClick(View view) {
        Intent i = new Intent(context, extended_report.class);
        i.putExtra(MyConstants.LIST_ITEM_INTENT, mainArray.get(getAdapterPosition()));
        context.startActivity(i);
        //Log.d("Mylog", "Pressed:" + getAdapterPosition());
    }

```

```

}

```

```

public void updateAdapter(List<ListItem> newList){
    mainArray.clear();
    mainArray.addAll(newList);
    notifyDataSetChanged();
}

```

}

}