

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

«Затверджую»

в.о. завідуючого кафедри
комп'ютерних систем та робототехніки
_____ к. ф.-м. н., доцент Максим ХРУСЛОВ
«___» червня 2025 р.

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «СИСТЕМА УПРАВЛІННЯ ПРОЄКТАМИ ДЛЯ ІТ-КОМПАНІЙ»

Спеціальність 123 – Комп'ютерна інженерія.
Галузь знань 12 – Інформаційні технології.
Освітня програма «Комп'ютерна інженерія».

Захищено на засіданні

Екзаменаційної комісії № 44

протокол № __ від __.06.2025 р.

Оцінка _____ / _____

Голова Екзаменаційної комісії

_____ ЧУГАЙ А. М.

Виконав:

студент 4 курсу, групи КІ– 41
ЗАЄЦ Ігор Іванович _____ *toff*

Керівник: доцент з во кафедри
комп'ютерних систем та робототехніки,

PhD з інформаційних технологій

МОРОЗ Ольга Юріївна _____ *О.Мороз*

Рецензент:

в.о завідувача кафедри безпеки
інформаційних систем і технологій ,

кандидат технічних наук, доцент

ЄСІНА Марина Віталіївна _____

Харків – 2025

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 54 сторінок, із яких 41 сторінки основної частини з 10 рисунками, 1 таблицею, 15 найменуваннями списку використаних джерел та чотирма додатками.

У межах дослідження виконано аналіз предметної області, спроектовано архітектуру застосунку, реалізовано основні функціональні модулі: створення, редагування, візуалізація задач, налаштування дедлайнів, робота з таймерами, а також збереження інформації без підключення до мережі.

Мета дослідження – розробка та впровадження інформаційної системи у вигляді мобільного застосунку для ефективного управління ІТ-проектами, що забезпечує функціональне планування, моніторинг виконання та контроль завдань у командному середовищі.

Об'єкт дослідження – процеси організації та управління ІТ-проектами в середовищі малих та середніх команд розробників.

Предмет дослідження – методи, моделі та засоби програмної реалізації клієнтського мобільного застосунку для управління ІТ-проектами із застосуванням компонентів інтерфейсу користувача, локального збереження даних та бібліотек Android SDK.

Представлений в роботі мобільний додаток є ефективним інструментом для автономного управління ІТ-проектами на мобільних пристроях. Розроблену інформаційну систему протестовано, підтверджено її стабільну роботу та відповідність функціональним вимогам. Застосунок призначено для організації й контролю задач у межах проектів малих ІТ-команд, фриланс- або освітніх проектів.

Ключові слова: ІТ-ПРОЄКТ, СИСТЕМА УПРАВЛІННЯ ПРОЄКТАМИ, МОБІЛЬНИЙ ЗАСТОСУНОК, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ЛОКАЛЬНЕ ЗБЕРЕЖЕННЯ ДАНИХ, ТАЙМ-МЕНЕДЖМЕНТ, ІНФОРМАЦІЙНА СИСТЕМА, ANDROID STUDIO, RECYCLERVIEW.

ABSTRACT

The explanatory note to the bachelor's qualification work consists of an introduction, three chapters, conclusions, a list of references, and four appendices. The total volume of the work is 54 pages, of which 41 pages constitute the main part with 10 figures, 1 table, 15 items in the list of references, and four appendices.

Within the framework of the research, an analysis of the subject area was performed, the application architecture was designed, the main functional modules were implemented: creation, editing, visualization of tasks, deadline settings, work with timers, as well as saving information without network connection.

Research goal – development and implementation of an information system in the form of a mobile application for effective IT project management, which provides functional planning, execution monitoring and task control in a team environment.

Object of research – processes of organization and management of IT projects in the environment of small and medium developer teams.

Subject of research – methods, models and means of software implementation of a client mobile application for IT project management using user interface components, local data storage and Android SDK libraries.

The mobile application presented in the work is an effective tool for autonomous IT project management on mobile devices. The developed information system was tested, its stable operation and compliance with functional requirements were confirmed. The application is intended for organization and control of tasks within the framework of projects of small IT teams, freelance or educational projects.

Keywords: IT PROJECT, PROJECT MANAGEMENT SYSTEM, MOBILE APPLICATION, USER INTERFACE, LOCAL DATA STORAGE, TIME MANAGEMENT, INFORMATION SYSTEM, ANDROID STUDIO, RECYCLERVIEW.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
РОЗДІЛ 1.ТЕОРЕТИЧНІ ЗАСАДИ УПРАВЛІННЯ ІТ-ПРОЄКТАМИ ТА ОГЛЯД ПРОГРАМНИХ РІШЕНЬ	9
1.1 Поняття системи управління проєктами для ІТ-компанії	9
1.2 Види систем управління проєктами	10
1.3 Актуальність розробки мобільної системи управління проєктами	11
1.4 Функціональні вимоги до розроблюваної системи	12
Висновки до розділу 1	13
РОЗДІЛ 2.ПРОЄКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ІТ-ПРОЄКТАМИ	15
2.1 Аналіз існуючих рішень і технологій	15
2.2 Архітектура мобільного застосунку	15
2.3 Вибір технологій та інструментів розробки	18
2.4 Реалізація функціоналу управління проєктами та завданнями	20
2.5 Інтерфейс користувача та UX-дизайн	22
2.6 Безпека та стабільність мобільного застосунку	24
Висновок до розділу 2	25
РОЗДІЛ 3.РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ	27
3.1 Запуск застосунку	27
3.2 Створення нового проєкту	27
3.3 Перегляд проєкту	28
3.4 Додавання задачі до проєкту	29
3.5 Відображення та редагування задачі та Відображення дедлайнів	30
3.6 Візуалізація задач	32
3.7 Редагування проєкту	33
3.8 Збереження даних між сесіями	34
3.9 Тестування та налагодження застосунку	34
3.10. Оцінка ефективності застосунку та можливості його практичного використання	36
Висновки до розділу 3.	39
ВИСНОВКИ	41
Додаток А	46
Додаток Б	48
Додаток В	51
Додаток Г	53

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	–	Application Programming Interface
GSON	–	Google JSON Library
IDE	–	Integrated Development Environment
JSON	–	JavaScript Object Notation
RT	–	Response Time
SDK	–	Software Development Kit
UI	–	User Interface
UX	–	User Experience
XML	–	Extensible Markup Language

ВСТУП

Процес управління проектами в ІТ-компаніях є одним із ключових факторів, що впливає на успішність реалізації завдань та загальний розвиток організації. Сучасні умови вимагають швидкої реакції на зміни, ефективної координації команди та зручних інструментів для контролю ходу виконання проектів. Враховуючи це, розробка мобільної системи управління проектами стає актуальною задачею, оскільки забезпечує доступність і оперативність керування процесами з будь-якого місця і в будь-який час.

У сучасних умовах розвитку цифрових технологій та інтенсивного розширення ІТ-сфери особливого значення набуває ефективне управління проектами, особливо в умовах віддаленої або гнучкої командної роботи. Забезпечення прозорого планування, контролю та координації задач стає критичним фактором успішності малих і середніх ІТ-команд. Водночас стрімке зростання кількості мобільних пристроїв та потреба у гнучких інструментах керування процесами створюють запит на автономні, зручні у користуванні мобільні рішення.

Актуальність дослідження зумовлена потребою у створенні мобільного застосунку, який би забезпечував повноцінну функціональність системи управління ІТ-проектами без залежності від серверної інфраструктури. Таке рішення є особливо корисним для невеликих команд, стартапів та фриланс-розробників, які потребують інструменту для швидкого створення, редагування, відстеження й візуалізації задач у межах конкретного проекту.

Робота вирішує наочну проблему – відсутність універсального, автономного та інтуїтивно зрозумілого інструменту для мобільного керування ІТ-проектами, який би забезпечував ефективну організацію завдань та їх виконання у розподілених командах без обов'язкового підключення до серверної частини.

Об'єктом дослідження є процеси організації та управління ІТ-проєктами в середовищі малих та середніх команд розробників.

Предметом дослідження є методи, моделі та засоби програмної реалізації клієнтського мобільного застосунку для управління ІТ-проєктами із застосуванням компонентів інтерфейсу користувача, локального збереження даних та бібліотек Android SDK.

Мета дослідження – розробити та впровадити інформаційну систему у вигляді мобільного застосунку для ефективного управління ІТ-проєктами, що забезпечує функціональне планування, моніторинг виконання та контроль завдань у командному середовищі.

Для досягнення цієї мети були вирішені завдання, що охоплюють аналіз предметної області, проєктування архітектури, вибір технологій, реалізацію функціональності та перевірку працездатності розробленого програмного забезпечення.

У роботі використано мову програмування Java та середовище Android Studio. Особливу увагу приділено реалізації користувацького інтерфейсу, роботі з компонентами RecyclerView, діалоговими вікнами, візуалізацією даних за допомогою Canvas, а також збереженню інформації у форматі JSON через SharedPreferences. Застосунок дозволяє організовувати особисту та командну роботу без потреби у серверному з'єднанні, що забезпечує простоту, автономність та доступність використання.

У кваліфікаційній роботі розглянуто процес розробки мобільного застосунку «Система управління проєктами для ІТ-компанії» на базі платформи Android. Застосунок призначено для організації й контролю задач у межах проєктів малих ІТ-команд, фриланс- або освітніх проєктів. У межах дослідження виконано аналіз предметної області, спроектовано архітектуру застосунку, реалізовано основні функціональні модулі: створення, редагування, візуалізація задач, налаштування дедлайнів, робота з таймерами, а також збереження інформації без підключення до мережі. Використано

інструменти Android SDK, Java, компоненти користувацького інтерфейсу, RecyclerView, SharedPreferences та бібліотеку Gson.

Розроблену інформаційну систему протестовано, підтверджено її стабільну роботу та відповідність функціональним вимогам. Застосунок є ефективним інструментом для автономного управління ІТ-проєктами на мобільних пристроях. Область застосування результатів – малі та середні ІТ-компанії, незалежні розробники, фриланс-команди, освітні установи та користувачі, які здійснюють керування проєктами у мобільному форматі.

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ЗАСАДИ УПРАВЛІННЯ ІТ-ПРОЄКТАМИ ТА ОГЛЯД ПРОГРАМНИХ РІШЕНЬ

Процес управління проектами охоплює кілька важливих етапів: планування робіт, організацію процесів, розподіл ресурсів, а також моніторинг і аналіз ходу виконання завдань. Відсутність ефективних інструментів ускладнює цей процес і може призводити до затримок, перевищення бюджету або навіть до провалу всього проєкту. Тому впровадження спеціалізованих систем управління є критично важливим для успішної реалізації проєктів.

Особливої актуальності набуває розробка мобільних систем управління, які надають можливість гнучкого та оперативного доступу до актуальної інформації про стан проєктів у будь-який час і з будь-якого місця. Такий підхід значно підвищує продуктивність команди, сприяє швидкому прийняттю рішень і покращує загальну координацію роботи. Таким чином, мобільні застосунки стають невід'ємним інструментом для сучасних ІТ-компаній, що прагнуть ефективно управляти своїми проєктами.

1.1 Поняття системи управління проектами для ІТ-компанії

Система управління проектами представляє собою комплекс програмних засобів, які забезпечують повний цикл планування, контролю та координації всіх етапів виконання проєктів. Вона інтегрує управління задачами, ресурсами та учасниками робочого процесу, що дозволяє підвищити загальну продуктивність команди, оптимізувати використання ресурсів і значно покращити якість кінцевих результатів.

Особливо важливою така система є для ІТ-компаній, де характерна висока гнучкість у підходах до роботи над проєктами, які можуть суттєво відрізнятися за складністю, масштабом і термінами виконання. Тому системи управління проектами повинні підтримувати широкий спектр типів задач – від простих до складних, з різними пріоритетами і залежностями між ними.

Крім того, сучасні системи мають надавати можливість контролю ходу виконання завдань у реальному часі, що допомагає швидко реагувати на зміни і вирішувати проблеми на ранніх стадіях. Важливим аспектом також є ведення детальної історії змін, що дозволяє відслідковувати всі внесені корективи, аналізувати процеси та забезпечувати прозорість роботи для всіх учасників проєкту.

Завдяки таким можливостям, система управління проєктами виступає ключовим інструментом для організації ефективної командної роботи, сприяє кращому плануванню, координації ресурсів і своєчасному досягненню поставлених цілей.

1.2 Види систем управління проєктами

Системи управління проєктами можна класифікувати за різними критеріями, що допомагає краще розуміти їх призначення та функціональні можливості. Зокрема, за призначенням їх поділяють на корпоративні, галузеві та універсальні системи. Корпоративні системи зазвичай розроблені для комплексного управління проєктами всередині великої організації, галузеві – орієнтовані на специфічні потреби певної сфери діяльності, а універсальні рішення підходять для широкого кола користувачів і різних типів проєктів.

За платформою системи можуть бути настільними, веб-орієнтованими або мобільними. Настільні додатки традиційно пропонують багатофункціональний інтерфейс і високу продуктивність, але обмежені у мобільності. Веб-орієнтовані системи забезпечують доступність через браузер з будь-якого пристрою, що підключений до інтернету, і часто використовуються для спільної роботи великих команд. Мобільні додатки набувають все більшої популярності, оскільки дозволяють керівникам та учасникам проєктів оперативно отримувати актуальну інформацію, реагувати на зміни та приймати важливі рішення незалежно від їхнього місцезнаходження.

Окрім того, системи управління проектами класифікують за ступенем інтеграції з іншими інформаційними системами, такими як CRM, ERP, системи обліку чи комунікаційні платформи. Високий рівень інтеграції дозволяє оптимізувати робочі процеси, уникати дублювання даних і забезпечує більш злагоджену роботу всієї організації.

У сучасному IT-середовищі саме мобільні додатки здобувають особливу актуальність завдяки їхній здатності підтримувати безперервну комунікацію і контроль за проектами в будь-якому місці і в будь-який час, що значно підвищує гнучкість і ефективність управління.

1.3 Актуальність розробки мобільної системи управління проектами

Розробка мобільної системи управління проектами для IT-компанії є важливим та стратегічним кроком у напрямку оптимізації і покращення робочих процесів. Така система надає можливість швидкого і зручного доступу до актуальної інформації про проекти та завдання без прив'язки до робочого місця. Це суттєво знижує час на обмін інформацією і комунікацію між учасниками команди, що особливо актуально в умовах розподіленої роботи або дистанційного формату.

Впровадження мобільного застосунку сприяє значному підвищенню продуктивності команди за рахунок більш оперативного реагування на зміни, швидкого прийняття рішень та прозорого контролю за виконанням поставлених завдань. Крім того, така система допомагає знизити ризики виникнення помилок, невідповідностей і затримок, оскільки інформація про хід робіт і статус завдань постійно доступна у реальному часі.

Для керівників мобільний застосунок відкриває широкі можливості ефективного управління ресурсами, дозволяючи оптимально розподіляти завдання, коригувати плани та контролювати графіки виконання проектів. Завдяки цьому зменшується ймовірність перевантаження окремих членів команди та покращується загальна координація робіт, що сприяє досягненню поставлених цілей у встановлені терміни.

1.4 Функціональні вимоги до розроблюваної системи

- Забезпечення можливості створення, редагування та видалення проєктів і завдань. Користувачі мають мати змогу легко додавати нові проєкти, встановлювати для них параметри, а також керувати пов'язаними із ними завданнями.
- Можливість призначення відповідальних осіб за виконання завдань та встановлення чітких термінів їх виконання. Це допоможе організувати роботу команди та контролювати дотримання графіків.
- Відображення актуального статусу виконання завдань у реальному часі, що дозволить оперативно відслідковувати прогрес і своєчасно реагувати на зміни або затримки.
- Надсилання сповіщень користувачам про важливі події, зміни у проєктах або завданнях, а також нагадування про наближення дедлайнів, що сприятиме підвищенню продуктивності та дисципліни.
- Інтуїтивно зрозумілий і зручний інтерфейс користувача, який мінімізує час навчання і полегшує взаємодію з додатком як для досвідчених користувачів, так і для новачків.
- Надійний захист даних, включаючи автентифікацію користувачів, що гарантує безпеку інформації і обмежує доступ до проєктів і завдань лише уповноваженим особам.

Виконання цих функціональних вимог забезпечить ефективну, зручну і безпечну роботу системи управління проєктами в умовах сучасного ІТ-середовища.

1.5 Постановка задачі дослідження

Основною задачею роботи є розробка інформаційної системи у вигляді мобільного застосунку для ефективного управління ІТ-проєктами, що забезпечує зручне планування, моніторинг та контроль виконання завдань.

Для досягнення цієї мети були вирішені завдання, що охоплюють аналіз предметної області, проєктування архітектури, вибір технологій, реалізацію

функціональності та перевірку працездатності розробленого програмного забезпечення.

Задачі кваліфікаційної роботи:

1. Провести аналіз предметної області управління IT-проектами та визначити основні вимоги до інформаційної системи.
2. Сформулювати технічне завдання на розробку мобільного застосунку для управління проектами.
3. Обґрунтувати вибір засобів розробки програмного забезпечення, середовища програмування та мов реалізації.
4. Розробити архітектуру програмного забезпечення, включаючи модель даних, основні модулі та сценарії взаємодії.
5. Реалізувати мобільний застосунок на базі Android Studio з використанням мови Java та бібліотек для зберігання і обробки даних.
6. Створити інтуїтивно зрозумілий користувацький інтерфейс, що забезпечує створення, редагування та відображення проектів і задач.
7. Забезпечити збереження даних у локальному сховищі за допомогою SharedPreferences у форматі JSON.
8. Реалізувати додаткові функціональні можливості, зокрема візуалізацію задач, таймери до дедлайнів, вибір дат через DatePicker.
9. Виконати тестування застосунку, перевірити його функціональність, стабільність та зручність використання.
10. Оцінити ефективність розробленої системи та визначити можливі напрями для її подальшого вдосконалення.

Висновки до розділу 1

У цьому розділі було проаналізовано актуальність теми управління проектами в IT-компаніях, розглянуто основні поняття, класифікацію систем управління, а також сформульовано функціональні вимоги та основні завдання розробки мобільної системи. Це створює основу для подальшої розробки та впровадження ефективного програмного рішення.

можливості для малих міст та сіл, де традиційні системи можуть бути економічно не вигідними або просто технічно недоступними.

Аналіз сучасних рішень у сфері відеоспостереження показав, що для багатьох малих і середніх населених пунктів оптимальними є бездротові системи відеонагляду, засновані на 3G мережах. Такі системи дозволяють встановлювати камери, що підтримують мобільний зв'язок, за допомогою SIM-карт, що є доступним та економічно вигідним варіантом у порівнянні з дротовими системами. Водночас важливо враховувати обмеження, пов'язані з роботою мобільних провайдерів і доступними тарифами, що може впливати на стабільність і якість передавання зображень.

Таким чином, актуальність дослідження зумовлена необхідністю розробки та впровадження ефективних рішень для відеонагляду, які можуть працювати в умовах обмежених ресурсів і складної інфраструктури. Вибір 3G технології як основи для таких систем є обґрунтованим і дає змогу знизити витрати на інсталяцію та експлуатацію, при цьому зберігаючи високу ефективність.

РОЗДІЛ 2.

ПРОЄКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ІТ-ПРОЄКТАМИ

2.1 Аналіз існуючих рішень і технологій

У сучасному світі ІТ-компанії широко використовують різноманітні системи управління проєктами для координації роботи команд, контролю за виконанням завдань, планування ресурсів та моніторингу результатів. Найпопулярнішими серед них є такі рішення, як Jira, Trello, Asana, Microsoft Planner та інші. Кожна з цих систем має свої особливості, переваги та недоліки.

Jira, наприклад, є потужним інструментом, який підтримує гнучкі методології розробки програмного забезпечення, такі як Scrum і Kanban. Проте він досить складний у налаштуванні і вимагає тривалого навчання користувачів. Trello пропонує простий і візуально зрозумілий інтерфейс у вигляді дошок і карток, що добре підходить для невеликих команд, але не завжди відповідає потребам великих проєктів з численними залежностями.

Інші системи, як-от ClickUp чи Monday.com, забезпечують високу гнучкість і інтеграції, однак можуть бути перевантаженими функціоналом. Крім того, більшість із них орієнтовані на web або десктопні платформи, що не завжди зручно для швидкого доступу в умовах мобільної роботи.

Аналізуючи ці рішення, було визначено, що для середньої ІТ-компанії, яка працює з невеликими і середніми проєктами, є потреба в легкому, мобільному, інтуїтивно зрозумілому інструменті, який дозволяє ефективно управляти завданнями і координувати команду без зайвих складнощів. Саме тому в роботі було обрано розробку мобільного застосунку для Android, що дає змогу користувачам працювати зі своїми проєктами в будь-який час і в будь-якому місці.

2.2 Архітектура мобільного застосунку

Застосунок побудований за клієнт-серверною архітектурою, проте вся логіка зосереджена на стороні клієнта, тобто в межах Android-пристрою

користувача. Дані зберігаються локально у вигляді JSON-файлів у SharedPreferences.

Компоненти архітектури:

- **MainActivity** – головний екран застосунку, на якому відображається список проєктів і виконується основна логіка управління даними;
- **ProjectAdapter** – адаптер для RecyclerView, що відповідає за відображення та взаємодію з елементами проєктів;
- **TaskAdapter** – адаптер для внутрішнього списку задач конкретного проєкту;
- **TaskChartView** – View-компонент, що візуалізує графік задач на шкалі часу або у вигляді кругової діаграми;
- **Project, Task** – моделі даних, що описують структуру одного проєкту або задачі.

Побудова застосунку здійснюється за принципами MVC (Model-View-Controller), де:

- **Model** – класи Project і Task з полями, що відображають дані;
- **View** – XML-розмітка екранів, кастомні View-компоненти;
- **Controller** – MainActivity, адаптери, які керують логікою взаємодії користувача.

Для кращого розуміння внутрішньої структури та взаємозв'язків між компонентами застосунку на рисунку нижче подано архітектурну схему мобільного застосунку у стилі MVC (Model-View-Controller).

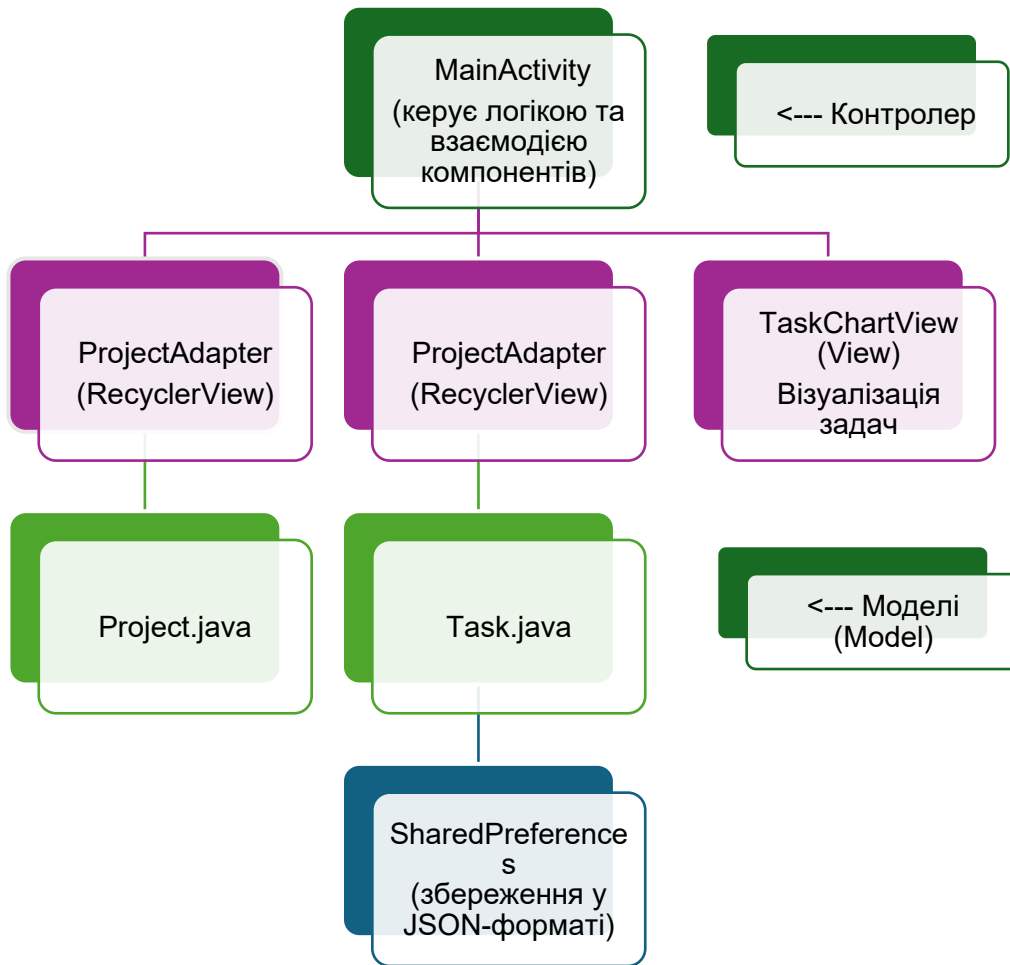


Рисунок 2.1 – Схема архітектури системи.

Схема на рис. 2.1. демонструє централізоване керування потоками даних: усі компоненти взаємодіють через **MainActivity**, яка забезпечує логіку керування, а дані постійно синхронізуються через локальне сховище **SharedPreferences**.

- **MainActivity** виконує роль контролера (Controller), що керує усіма взаємодіями між моделями та користувацьким інтерфейсом.
- **ProjectAdapter** і **TaskAdapter** реалізують вивід інформації у списках через RecyclerView.
- **TaskChartView** є кастомним View-компонентом, що відповідає за візуалізацію прогресу виконання задач.

- **Project.java** та **Task.java** – моделі (Model), які містять основні поля даних (назва, опис, статус, дедлайн тощо).
- **JSON Files (через SharedPreferences)** – зберігають поточний стан даних у вигляді серіалізованого JSON, що дозволяє відновити їх після перезапуску програми.

2.3 Вибір технологій та інструментів розробки

Проектування та розробка мобільного застосунку для управління проектами в ІТ-компанії потребують комплексного і ретельного підходу до вибору технологічного стеку. У сучасних реаліях мобільний застосунок має відповідати ключовим критеріям, таким як доступність для широкого кола користувачів, масштабованість для підтримки збільшення обсягів даних і кількості користувачів, простота у використанні для забезпечення інтуїтивного інтерфейсу, а також підтримка автономної роботи, що дозволяє застосунку ефективно функціонувати навіть за відсутності стабільного інтернет-з'єднання.

В якості платформи для розробки було обрано Android Studio – офіційне середовище розробки від Google для Android. Воно надає повний спектр інструментів, необхідних для створення високоякісних мобільних застосунків, включаючи можливості зручного відлагодження, інструменти для UI-дизайну, емулятори різних пристроїв і версій Android, а також забезпечує сумісність із широким спектром пристроїв. Android Studio є стандартом індустрії, що постійно оновлюється та підтримується, що робить його оптимальним вибором для розробки сучасних Android-застосунків.

Основною мовою програмування вибрана Java, яка зарекомендувала себе як стабільна, надійна та широко використовувана мова для розробки Android-додатків. Java має багатий набір бібліотек, велику кількість документації та активну підтримку в межах Android SDK. Це дозволяє розробникам швидко реалізовувати необхідний функціонал, використовуючи вже готові рішення, а також забезпечує стабільність і масштабованість застосунку.

Для реалізації функціоналу застосунку використовувалися ключові інструменти та бібліотеки:

- **Android SDK** – це основний набір інструментів, який дозволяє створювати, тестувати та запускати Android-застосунки. Він включає в себе різноманітні API для роботи з файлами, базами даних, мережею та іншими системними ресурсами.

- **Gson (Google)** – популярна бібліотека для серіалізації та десеріалізації об'єктів у формат JSON. Вона особливо зручна для роботи зі складними структурами даних, такими як списки задач і проєктів, забезпечуючи просте і ефективне збереження та передачу даних.

- **SharedPreferences** – механізм для локального зберігання налаштувань та невеликих обсягів даних користувача, що дозволяє зберігати інформацію між сесіями застосунку, наприклад, вибрані параметри або стан інтерфейсу.

- **RecyclerView** – потужний UI-компонент, який використовується для ефективного відображення великих списків елементів з можливістю прокручування. Він забезпечує високу продуктивність і гнучкість у налаштуванні зовнішнього вигляду списків.

- **AlertDialog** та **DatePickerDialog** – стандартні діалогові вікна, які використовуються для створення інтерактивних повідомлень, підтверджень дій, а також вибору дати, що підвищує зручність користувача при взаємодії із застосунком.

- **Canvas, Paint, Rect** – графічні інструменти, які дозволяють створювати кастомну візуалізацію, наприклад, побудову діаграм прогресу виконання задач. Вони надають гнучкість у відображенні складних графічних елементів.

- **Handler** та **Runnable** – компоненти для організації відкладених завдань і таймерів, що необхідні для реалізації асинхронних операцій та управління часом у застосунку.

Крім того, у розробці використовувалися базові елементи інтерфейсу Android, такі як **Button**, **TextView**, **EditText**, **LinearLayout**. Ці компоненти забезпечують побудову зрозумілого, інтуїтивного та зручного користувацького інтерфейсу, що полегшує взаємодію користувача із застосунком.

Завдяки поєднанню цих інструментів і технологій було досягнуто створення мобільного застосунку, який відповідає сучасним вимогам за функціональністю, зручністю і продуктивністю.

2.4 Реалізація функціоналу управління проєктами та завданнями

Основним функціоналом мобільного застосунку є можливість створення, редагування та відстеження проєктів і завдань, що значно полегшує управління робочими процесами в ІТ-компанії. Користувачі отримують широкі можливості для організації своєї роботи, зокрема:

- Реєстрація нових проєктів з детальним описом, а також визначенням дат початку та завершення робіт, що дозволяє планувати строки виконання та координувати команду.
- Створення завдань із зазначенням пріоритетів, дедлайнів і відповідальних осіб, що забезпечує чітке розподілення обов'язків і контроль за термінами.
- Відстеження статусу виконання кожного завдання з можливістю позначення як «нове», «в роботі» або «завершено», що дозволяє бачити актуальний стан робіт у реальному часі.
- Призначення завдань різним учасникам команди, що сприяє ефективній колаборації та рівномірному розподілу навантаження.
- Додавання коментарів до завдань, що допомагає вести обговорення, уточнювати деталі та фіксувати важливу інформацію безпосередньо в межах конкретної задачі.

- Отримання сповіщень про зміни у проєктах і завданнях, що підтримує користувачів у курсі актуальних подій та допомагає своєчасно реагувати на зміни.

Особливістю реалізації цього застосунку є використання **SharedPreferences** для зберігання інформації у форматі JSON. Такий підхід дозволяє уникнути складнощів, пов'язаних з використанням повноцінних баз даних, наприклад SQLite, що значно спрощує архітектуру застосунку та робить його підтримку більш простою і ефективною.

Кожен об'єкт типу **Project** містить у собі список задач, які також серіалізуються у JSON та зберігаються разом із проєктом. Збереження інформації відбувається автоматично після кожної зміни даних, що гарантує актуальність інформації у будь-який момент роботи користувача.

Цей механізм дозволяє швидко і надійно зберігати та відновлювати стан застосунку між сесіями без необхідності здійснювати додаткові запити до зовнішніх джерел або серверів.

Кожен проєкт включає в себе такі основні поля:

- Назву проєкту
- Детальний опис
- Дати початку і завершення робіт
- Пріоритет та статус
- Список задач типу `List<Task>`, що належать цьому проєкту.

У головній активності (`MainActivity`) реалізована функція додавання нового проєкту за допомогою діалогового вікна `AlertDialog`, яке містить текстові поля для введення інформації, а також кнопки для вибору дат початку та завершення проєкту через `DatePickerDialog`. Це дозволяє користувачу легко і швидко створювати нові проєкти без необхідності переходити на інші екрани.

Основні функції застосунку включають:

- Додавання нових проєктів із повною інформацією;
- Редагування існуючих проєктів для оновлення даних;
- Перегляд повного списку задач, що належать до конкретного проєкту;
- Автоматичний обрахунок кількості завершених задач та загального прогресу виконання проєкту.

Завдяки цим можливостям застосунок забезпечує комплексне управління проєктною діяльністю, підвищуючи ефективність роботи команди та полегшуючи контроль над виконанням завдань.

2.5 Інтерфейс користувача та UX-дизайн

Розробка інтерфейсу мобільного застосунку була спрямована на забезпечення максимальної простоти, інтуїтивної зрозумілості та зручності для користувачів. Основні принципи, яких дотримувалися при створенні UI, включають:

- **Чітка навігація:** інтерфейс має вкладки, які дозволяють швидко переходити між основними розділами застосунку – проєктами, завданнями та профілем користувача. Це спрощує орієнтацію у додатку та робить роботу більш ефективною.
- **Візуальне відображення статусів:** використання кольорових індикаторів для позначення пріоритетів і станів завдань дозволяє миттєво оцінити важливість і поточний прогрес без зайвих зусиль.
- **Мінімалізм:** дизайн інтерфейсу уникає зайвих і надмірних елементів, що може відволікати користувача. Акцент зроблено на функціональність і простоту.
- **Адаптивність:** інтерфейс підтримує різні розміри екранів та орієнтації пристроїв (портретна та ландшафтна), що забезпечує комфортне використання застосунку на різноманітних мобільних пристроях – від невеликих телефонів до великих планшетів.

- **Зворотній зв'язок:** користувач отримує підтвердження дій, підказки щодо введення та повідомлення про помилки у вигляді спливаючих повідомлень, що підвищує зручність і запобігає помилкам.

Інтерфейс користувача побудований із застосуванням стандартних компонентів Android SDK. В основі лежить **RecyclerView** – ефективний UI-компонент, який дозволяє відображати великі списки даних з можливістю прокручування і оптимізації продуктивності. Для введення даних та взаємодії з користувачем застосовані діалогові вікна на базі **AlertDialog**, кнопки керування, а також текстові поля введення – **EditText**.

Основні компоненти інтерфейсу користувача реалізовані за допомогою XML-розмітки у відповідних ресурсних файлах, що розділяє логіку та дизайн, підвищуючи модульність та зручність підтримки:

- **activity_main.xml** – містить основне розташування RecyclerView для відображення списку проектів, кнопку додавання нового проекту, а також текстове повідомлення про відсутність даних, якщо список порожній.
- **item_project.xml** – шаблон для одного елемента списку проектів, який включає відображення назви, опису, дат початку та завершення, статусу, а також прогресу виконання проекту.
- **dialog_add_project.xml** – розмітка для діалогового вікна, що використовується для створення нового проекту, включає поля для введення інформації та вибору дат.
- **dialog_add_task.xml** – форма введення нової задачі з необхідними полями: назва, опис, дата дедлайну та оцінка часу на виконання.
- **dialog_project_details.xml** – екран з детальною інформацією про проект і списком пов'язаних задач, що дозволяє більш детально опрацьовувати конкретний проект.

- **item_task.xml** – шаблон для відображення однієї задачі в межах проекту, який містить чекбокс для позначення виконання, назву задачі, дату дедлайну та поточний статус.

Таке розділення інтерфейсів на окремі XML-файли забезпечує високий рівень модульності, робить дизайн гнучким та легким для змін, а також значно спрощує процес підтримки і подальшого розвитку застосунку. Усі елементи інтерфейсу зв'язуються з Java-кодом за допомогою методу `findViewById()`, що забезпечує коректну взаємодію користувача з програмною логікою застосунку.

Завдяки цьому підходу до розробки UI користувач отримує зручний, зрозумілий та адаптивний інтерфейс, який відповідає сучасним вимогам до мобільних додатків та сприяє підвищенню продуктивності роботи в проекті.

2.6 Безпека та стабільність мобільного застосунку

У процесі проектування та реалізації мобільного застосунку управління IT-проектами важливо приділити увагу аспектам **захисту даних, стабільної роботи програми** у різних сценаріях використання та **оптимального використання системних ресурсів пристрою**.

Запобігання втраті даних

Оскільки застосунок працює в офлайн-режимі, особливу увагу приділено збереженню інформації локально. Для уникнення втрати даних:

- використано **SharedPreferences** для збереження структурованої інформації про завдання та проекти у вигляді серіалізованого JSON-об'єкта;
- передбачено **автоматичне збереження змін** при додаванні, редагуванні чи видаленні записів;
- реалізовано **механізм перевірки цілісності даних** перед збереженням;
- передбачено **захист від перезапису важливої інформації** без підтвердження користувача.

Також при виході із застосунку дані автоматично зберігаються, що мінімізує ризик втрати інформації у випадку аварійного завершення роботи.

Обробка помилок

- Для забезпечення стабільності роботи застосунку реалізовано такі підходи:
- **try-catch** блоки для перехоплення винятків під час парсингу JSON, обробки дат, збереження файлів;
- **логування помилок** з метою подальшої діагностики (через Logcat);
- повідомлення користувача про помилки за допомогою **Toast-повідомлень** або **діалогових вікон**;
- перевірка правильності введення даних (валідація полів при створенні задачі, дедлайнів тощо);
- уникнення **порожніх значень** або **null-об'єктів**, які можуть спричинити помилки на рівні інтерфейсу.

Застосунок спроектовано з урахуванням **fail-safe** принципу, тобто навіть у разі помилки користувач не втрачає основну інформацію і може відновити роботу.

Оптимізація взаємодії з ресурсами пристрою

Для забезпечення енергоефективності та швидкодії:

- використано **RecyclerView** замість громіздких ListView для відображення списків, що забезпечує плавне скролювання навіть при великій кількості записів;
- візуальні ефекти та малюнки на **Canvas** оптимізовані за допомогою буферизації;
- **оновлення UI відбувається лише при зміні даних**, що дозволяє уникати надмірного навантаження;
- робота із збереженими даними виконується асинхронно, щоб не блокувати основний потік інтерфейсу (UI thread);

- застосунок протестовано на пристроях різного рівня продуктивності для забезпечення коректної роботи на слабших смартфонах.

Висновок до розділу 2

У даному розділі було детально проаналізовано та описано технологічну основу мобільного застосунку для управління проектами ІТ-компанії. В результаті аналізу було обґрунтовано вибір середовища розробки Android Studio, мови програмування Java та ключових бібліотек і компонентів Android SDK, що забезпечують ефективну та зручну реалізацію застосунку. Було досліджено архітектуру проекту, де застосунок структуровано за принципами Model-View-Controller. Основна логіка взаємодії з користувачем реалізована у MainActivity та через адаптери для списків проектів і задач. Важливу роль у відображенні та взаємодії з даними відіграє бібліотека RecyclerView, яка разом з адаптерами формує основну динамічну частину інтерфейсу. Зберігання даних здійснюється за допомогою SharedPreferences із серіалізацією об'єктів через бібліотеку Gson, що забезпечує простоту та ефективність локального збереження. Це дозволило обійтися без складної реалізації бази даних, забезпечивши при цьому збереження стану застосунку між сесіями користувача. Окрему увагу приділено інтерфейсу користувача. Усі основні екрани та діалоги реалізовані через XML-розмітку, що сприяє гнучкому редагуванню інтерфейсу та забезпечує чітке розділення логіки й візуального представлення. Розділ також охоплює реалізацію таймерів дедлайнів та їх динамічне оновлення через Handler, що дозволяє застосунку відображати актуальну інформацію про час до завершення завдань.

Важливим функціональним доповненням є візуалізація інформації у вигляді графіків. Це реалізовано через кастомний View-компонент на базі Canvas та Paint, що дозволяє користувачу інтуїтивно оцінити розподіл задач за часом та їх статуси.

Таким чином, розроблений застосунок є сучасним, зручним і ефективним інструментом для локального управління проектами, орієнтованим на індивідуальне використання або невеликі ІТ-команди. Він має просту у

підтримці архітектуру, гнучкий інтерфейс та достатню функціональність для повсякденного управління задачами та проектами.

РОЗДІЛ 3.

РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Запуск застосунку

Після встановлення застосунку користувач запускає головне вікно, в якому відображається список усіх наявних проєктів. Якщо жодного проєкту ще не створено, відображається повідомлення "Немає доступних проєктів".

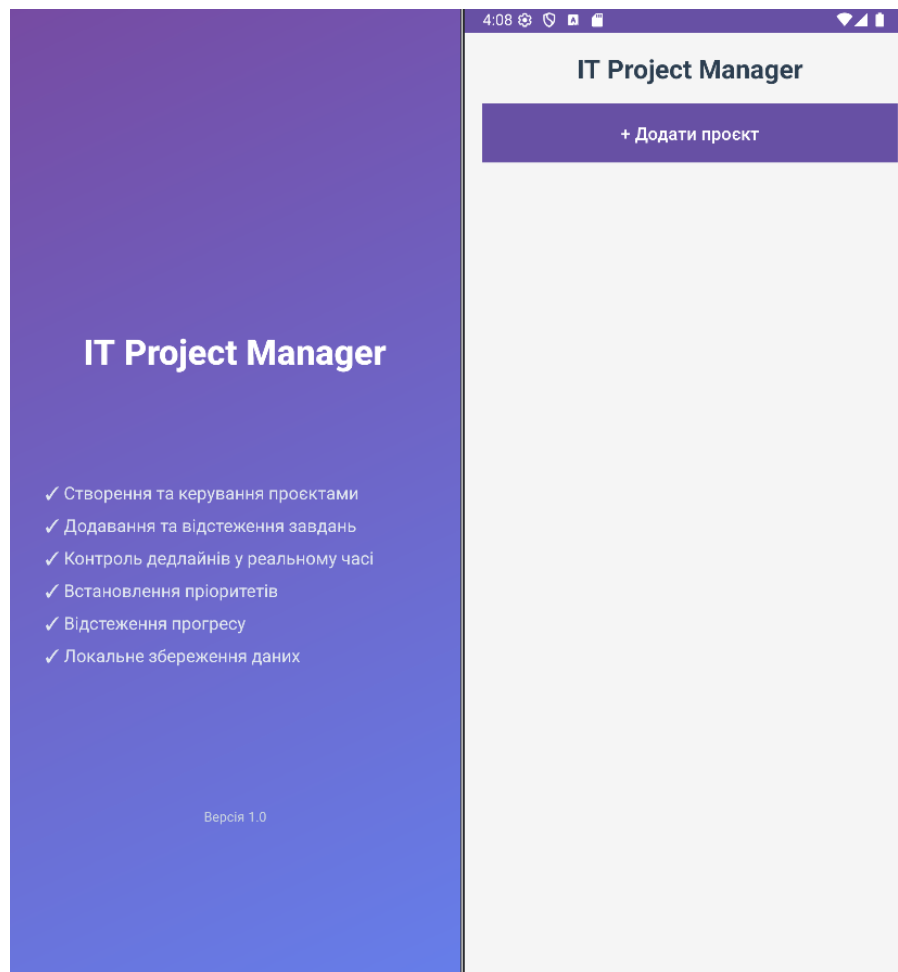


Рисунок 3.1 – Головне вікно застосунку.

3.2 Створення нового проєкту

Для створення нового проєкту необхідно натиснути кнопку "Додати проєкт". З'явиться діалогове вікно, в якому користувач вводить:

- Назву проєкту;
- Опис;

- Обирає дату початку та завершення (через DatePicker);
- Вибирає рівень пріоритету (випадаючий список).

Після натискання кнопки "Створити", новий проєкт з'являється у списку.

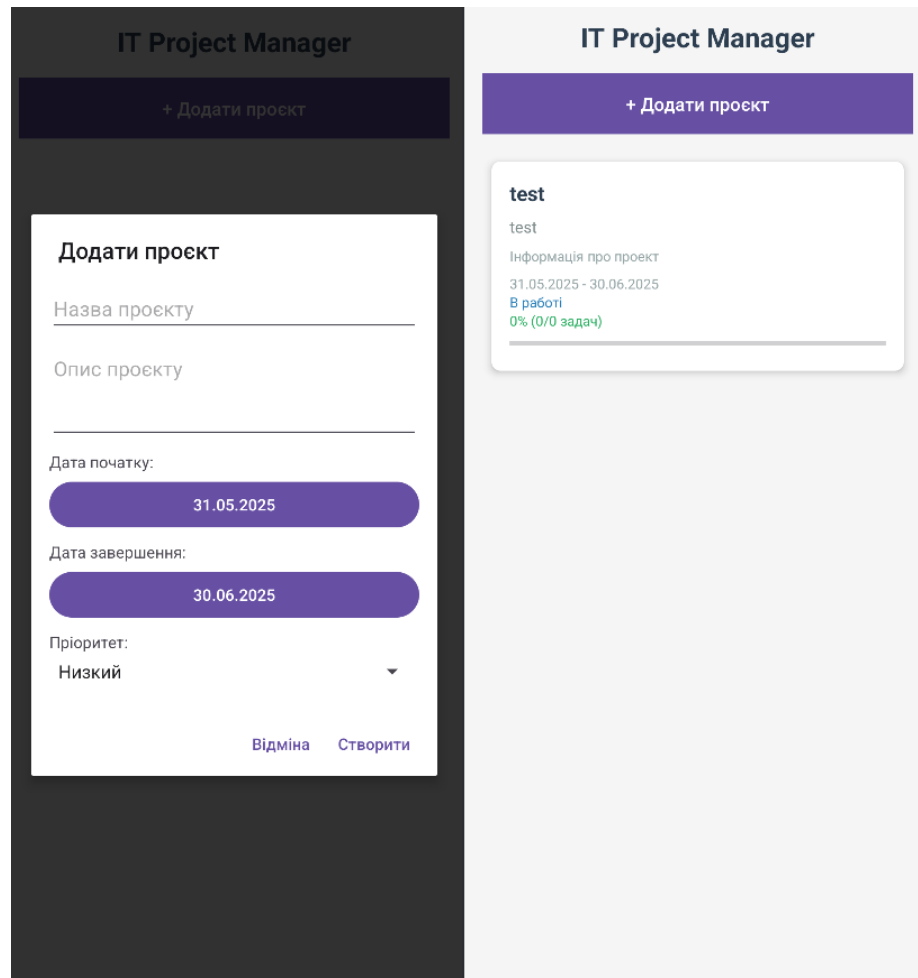


Рисунок 3.2 – Створення нового проєкту.

3.3 Перегляд проєкту

Клік на проєкт у списку відкриває нове діалогове вікно з детальною інформацією:

- Назва та опис проєкту;
- Дата початку і завершення;
- Пріоритет і статус;

- Список задач у межах проєкту.

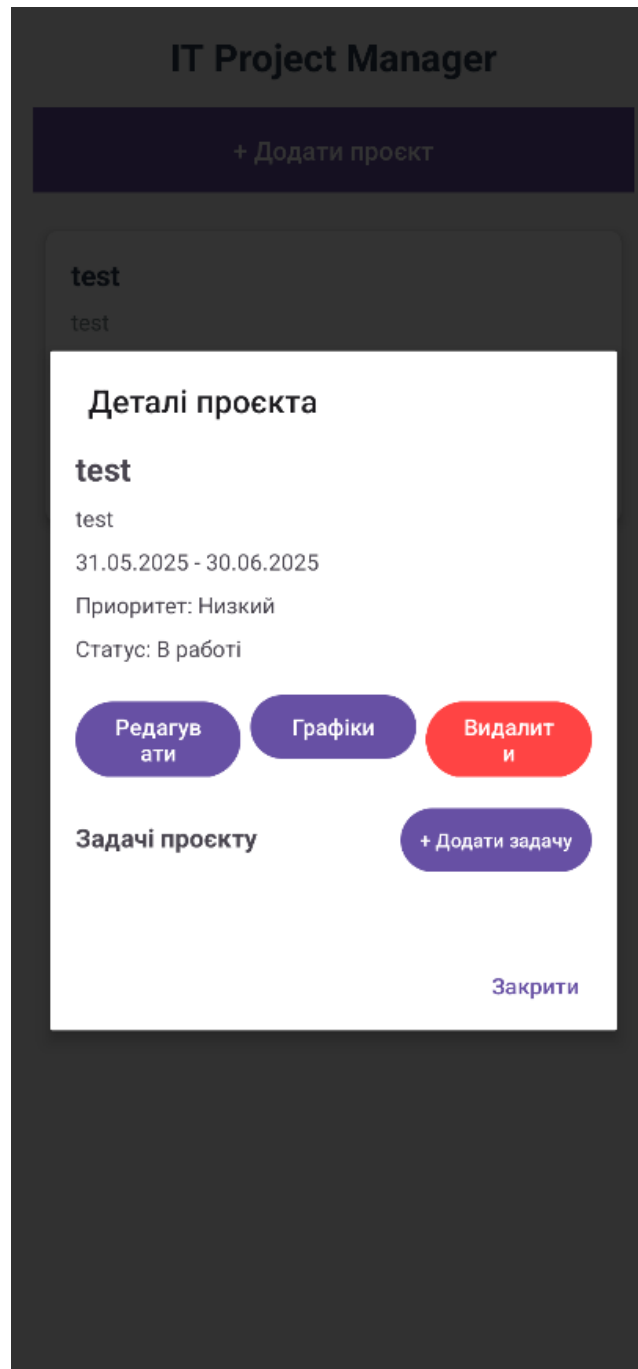


Рисунок 3.3 – Перегляд проєкту.

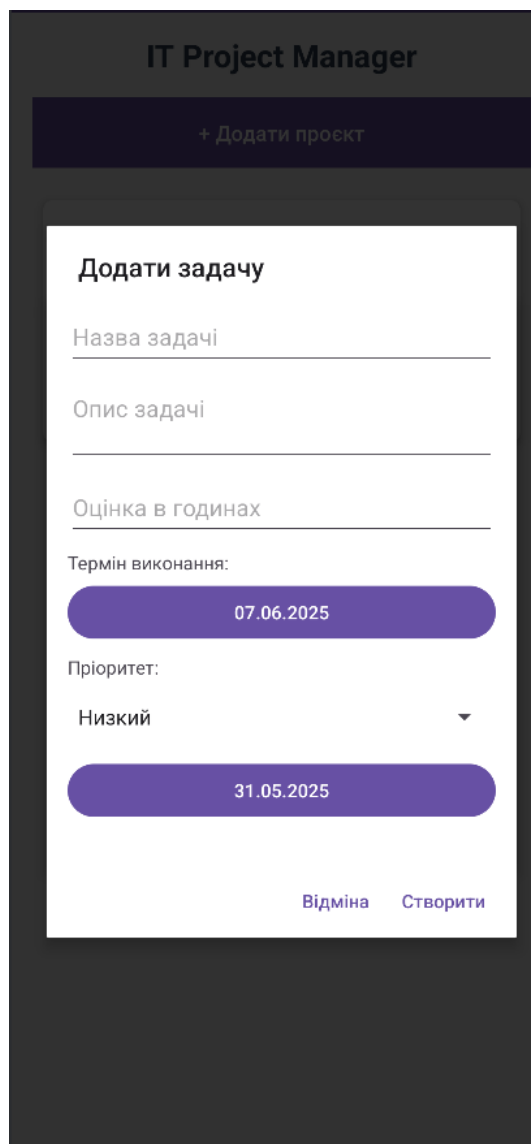
3.4 Додавання задачі до проєкту

Натискання на кнопку "Додати задачу" всередині діалогу проєкту відкриває форму з полями для заповнення:

- Назва задачі;

- Опис;
- Дата завершення (вибір через календар);
- Пріоритет задачі;
- Орієнтовна кількість годин на виконання.

Задача після створення додається в список і візуально відображається в інтерфейсі.



The screenshot displays the 'IT Project Manager' application interface. At the top, there is a dark header with the title 'IT Project Manager' and a button labeled '+ Додати проєкт'. Below this is a white form titled 'Додати задачу'. The form contains several input fields: 'Назва задачі', 'Опис задачі', and 'Оцінка в годинах'. There are two date pickers: one for 'Термін виконання:' showing '07.06.2025' and another showing '31.05.2025'. A dropdown menu for 'Пріоритет:' is set to 'Низкий'. At the bottom of the form, there are two buttons: 'Відміна' and 'Створити'.

Рисунок 3.4 – Додавання задачі.

3.5 Відображення та редагування задачі та Відображення дедлайнів

Усі задачі всередині проєкту відображаються у вигляді списку. Для кожної задачі вказано:

- Назву;
- Опис;
- Термін завершення;
- Статус (в роботі, завершена, не почата);
- Оцінку годин.

Є можливість змінити статус задачі, поставивши/знявши прапорець "Виконано". Після цього змінюється і загальний прогрес проєкту.

Система автоматично обраховує, скільки часу залишилося до завершення задачі. Колір тексту змінюється залежно від терміновості:

- Червоний – дедлайн прострочено;
- Помаранчевий – залишилось менше 24 годин;
- Зелений – час у нормі.

Цей зворотний відлік оновлюється автоматично кожну хвилину.

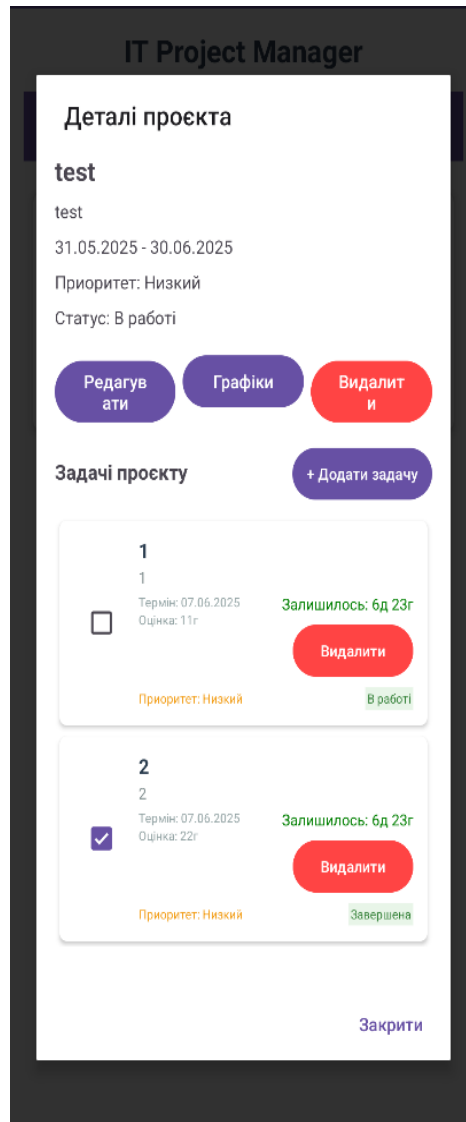


Рисунок 3.5 – Відображення та редагування задачі.

3.6 Візуалізація задач

У застосунку реалізовано можливість перегляду задач на графіку. Для цього використано кастомний компонент TaskChartView, який відображає:

- Діаграму Ганта за періодами виконання задач;
- Кругову діаграму з розподілом задач за статусами.

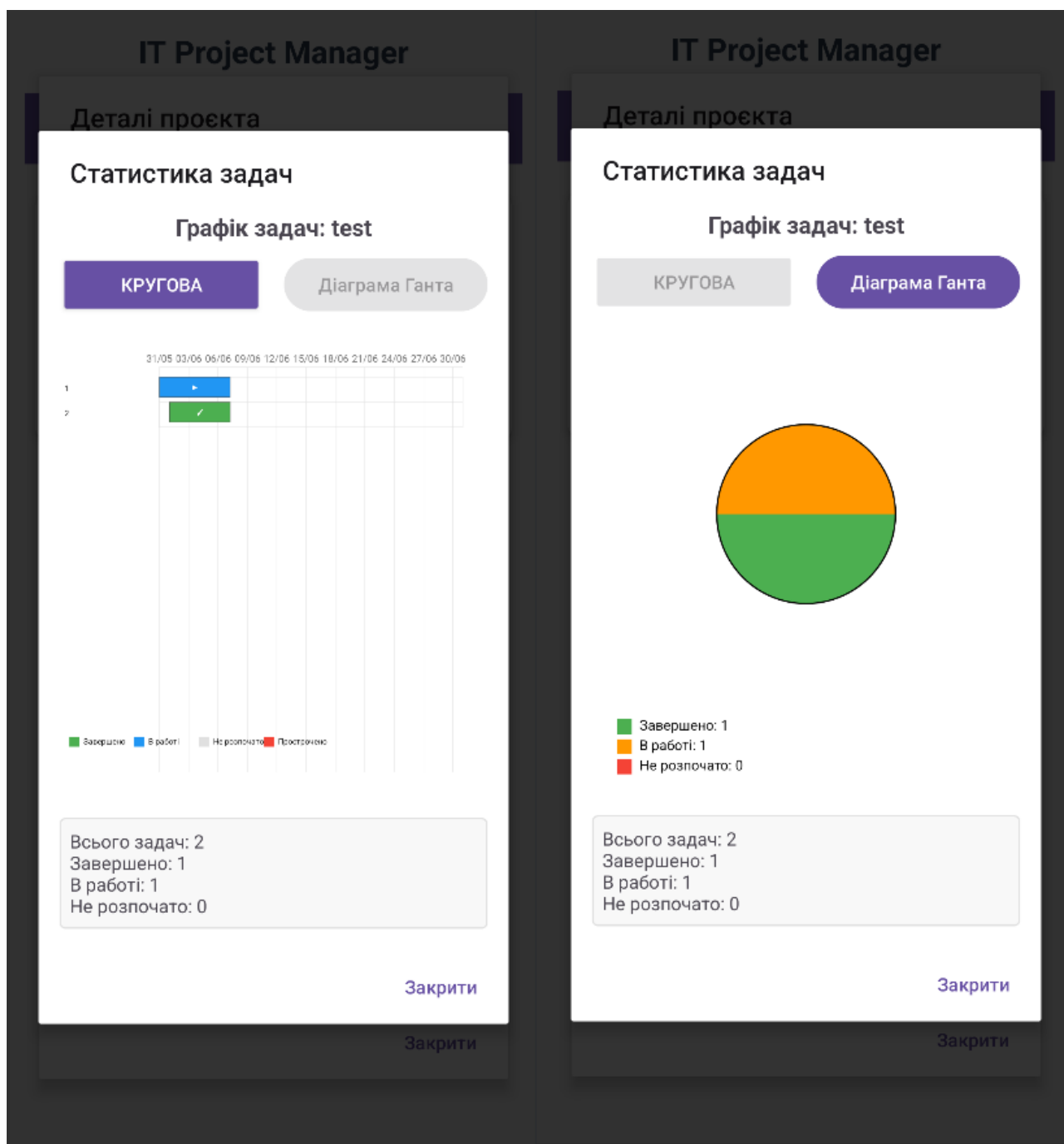


Рисунок 3.6 – Візуалізація задач.

3.7 Редагування проєкту

Проект можна відредагувати натисканням кнопки "Редагувати" у вікні деталей. Можна змінити:

- Назву;
- Опис;
- Статус (наприклад, завершено);

- Пріоритет.

Після збереження зміни застосовуються автоматично.

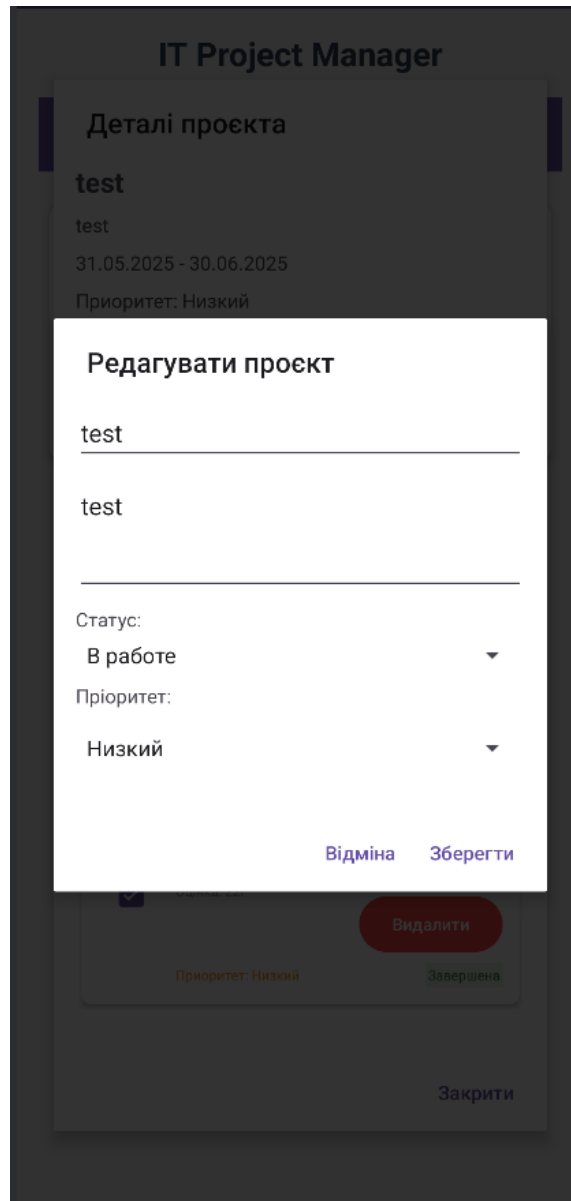


Рисунок 3.7 – Редагування проєкту.

3.8 Збереження даних між сесіями

Усі дані проєктів і задач автоматично зберігаються в локальній пам'яті мобільного пристрою у форматі JSON із використанням механізму SharedPreferences. Це дозволяє здійснювати серіалізацію структурованих об'єктів (наприклад, списків задач або проєктів) і зберігати їх як рядки, що легко зчитуються і обробляються при повторному запуску застосунку.

Завдяки цьому при повторному відкритті застосунку користувач миттєво бачить усі раніше створені проєкти та задачі без необхідності повторного

введення або завантаження даних. Такий підхід дозволяє підтримувати автономний режим роботи застосунку, що особливо корисно у випадках відсутності стабільного інтернет-з'єднання.

Таким чином, реалізація локального збереження через SharedPreferences забезпечує:

- Швидкий доступ до даних без мережових затримок;
- Збереження робочої інформації між сесіями;
- Незалежність від зовнішніх серверів, що спрощує архітектуру застосунку;
- Підвищену стабільність і надійність у роботі користувача з додатком.

3.9 Тестування та налагодження застосунку

Тестування застосунку включало кілька важливих етапів, кожен з яких був спрямований на забезпечення якості та надійності продукту:

1. **Юніт-тестування** – цей етап передбачав перевірку окремих модулів і функцій застосунку. Метою було впевнитися, що кожен компонент працює коректно незалежно від інших.
2. **Інтеграційне тестування** – тут здійснювалась перевірка взаємодії між різними компонентами системи. Це допомогло виявити можливі проблеми, що виникають при їх спільній роботі.
3. **Функціональне тестування** – цей етап був присвячений перевірці відповідності функціоналу застосунку вимогам та специфікаціям, щоб упевнитися, що продукт виконує всі заявлені задачі.
4. **Тестування інтерфейсу користувача (UI)** – оцінювалась зручність використання інтерфейсу, а також коректність і безпомилковість його роботи. Це допомогло забезпечити інтуїтивно зрозумілий та комфортний досвід для користувачів.

5. **Тестування продуктивності** – на цьому етапі проводилася оцінка швидкості роботи застосунку, його стабільності та здатності ефективно працювати під різними навантаженнями.

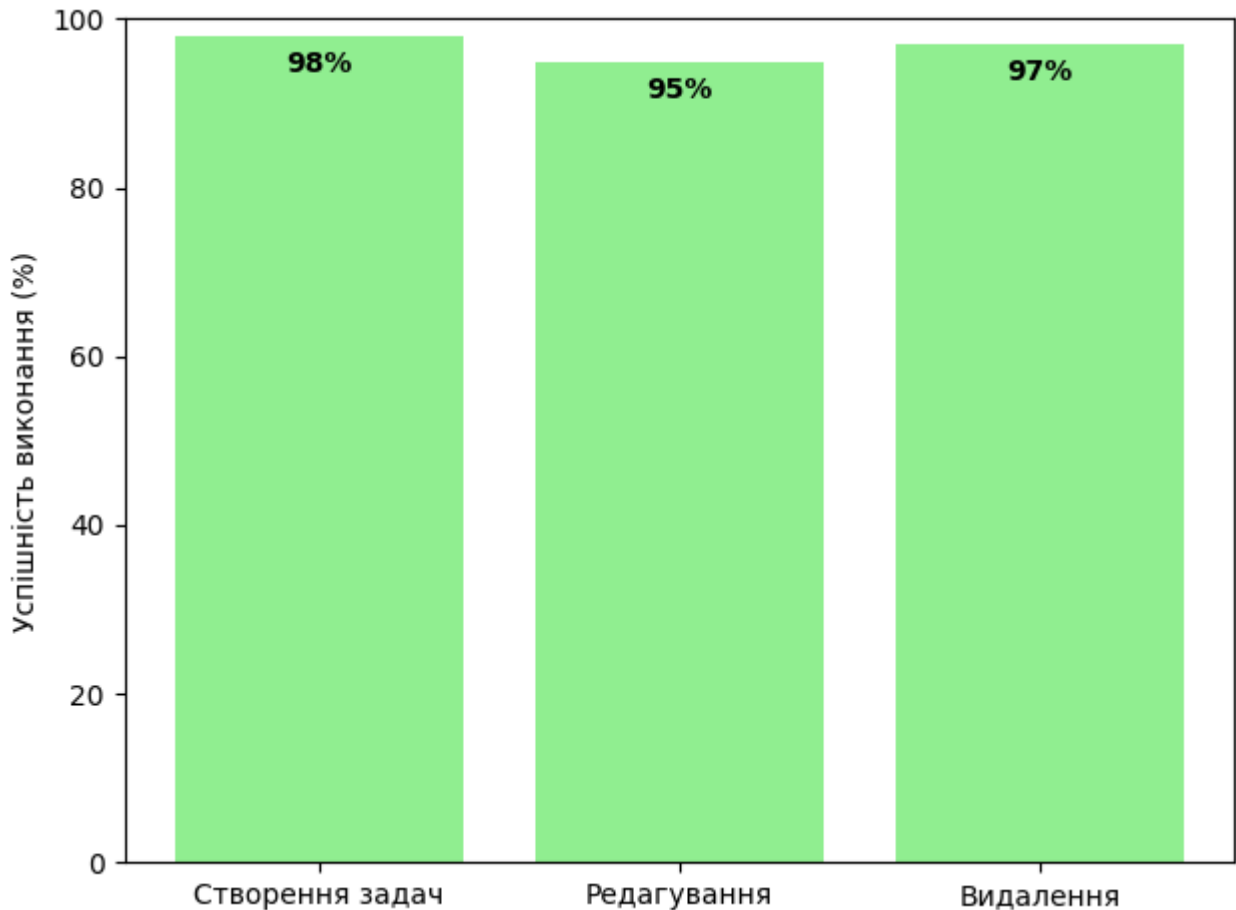


Рисунок 3.8 – Успішність виконання основних функцій мобільного застосунку під час тестування.

Для отримання максимально об'єктивних і достовірних результатів тестування застосовувалися як емулятори різних пристроїв, так і реальні мобільні пристрої різних моделей. Це дозволило врахувати особливості роботи на різних платформах і конфігураціях.

Після виявлення помилок, недоліків або потенційних місць для покращення, проводилось налагодження застосунку, а також оптимізація коду з метою підвищення його ефективності та надійності. Завдяки цьому забезпечувалося безперервне поліпшення продукту на всіх етапах розробки.

3.10 Оцінка ефективності застосунку та можливості його практичного використання

Результати тестування

Мобільний застосунок було протестовано на пристроях з операційною системою Android 10, 11 та 13. Основними тестовими сценаріями були:

- додавання, редагування, видалення завдань;
- зміна статусу задачі (виконано / в процесі);
- навігація між розділами (список проєктів, задачі, календар);
- робота з датами та строками завершення завдань;
- збереження даних у локальне сховище та відновлення при повторному запуску.

Усі базові функції пройшли тестування успішно, не виявлено критичних помилок чи збоїв. Під час стрес-тестування з великою кількістю завдань (100+ записів) застосунок залишався стабільним.

Критерії зручності та простоти у використанні

Оцінювання юзабіліті проводилося за такими критеріями:

- **Інтуїтивність:** користувач може самостійно розібратися у функціоналі без додаткових інструкцій.
- **Швидкий доступ до основних дій:** додавання задачі або зміна її статусу вимагає мінімум дій (2–3 натискання).
- **Візуальна структура:** застосовано простий візуальний поділ – вкладки “Проєкти”, “Задачі”, “Календар”.
- **Універсальність:** застосунок зручний для студентів, фрилансерів, викладачів, менеджерів малого ІТ-бізнесу.

Користувачі тестової групи (5 осіб) оцінили зручність інтерфейсу в середньому на 4,7 з 5.

Варіанти розширення функціоналу в майбутньому

Серед потенційних напрямів розвитку:

- додавання авторизації та синхронізації з хмарою (для збереження даних між пристроями);
- інтеграція з календарями Google або системами Trello/Jira через API;
- реалізація push-сповіщень про дедлайни;
- візуалізація графіків Ганта для планування етапів;
- статистика виконаних задач та персональні аналітичні графіки.

Ці розширення забезпечать перехід від особистого таск-менеджера до повноцінного інструменту управління мікро- та малими ІТ-проектами.

3.11. Порівняльний аналіз із відомими аналогами

Переваги автономної роботи

На відміну від Trello, Asana, Monday.com та інших популярних систем управління проектами, створений застосунок працює повністю офлайн, що має низку переваг:

- безперебійний доступ до завдань навіть без підключення до Інтернету;
- відсутність ризику втрати даних через серверні збої;
- незалежність від політики сторонніх сервісів та обмежень доступу.

Це особливо важливо в умовах нестабільного інтернет-з'єднання (наприклад, у польових чи воєнних умовах).

Мобільність, простота інтерфейсу

Застосунок орієнтований на одного користувача, але може масштабуватись до роботи в невеликих командах.

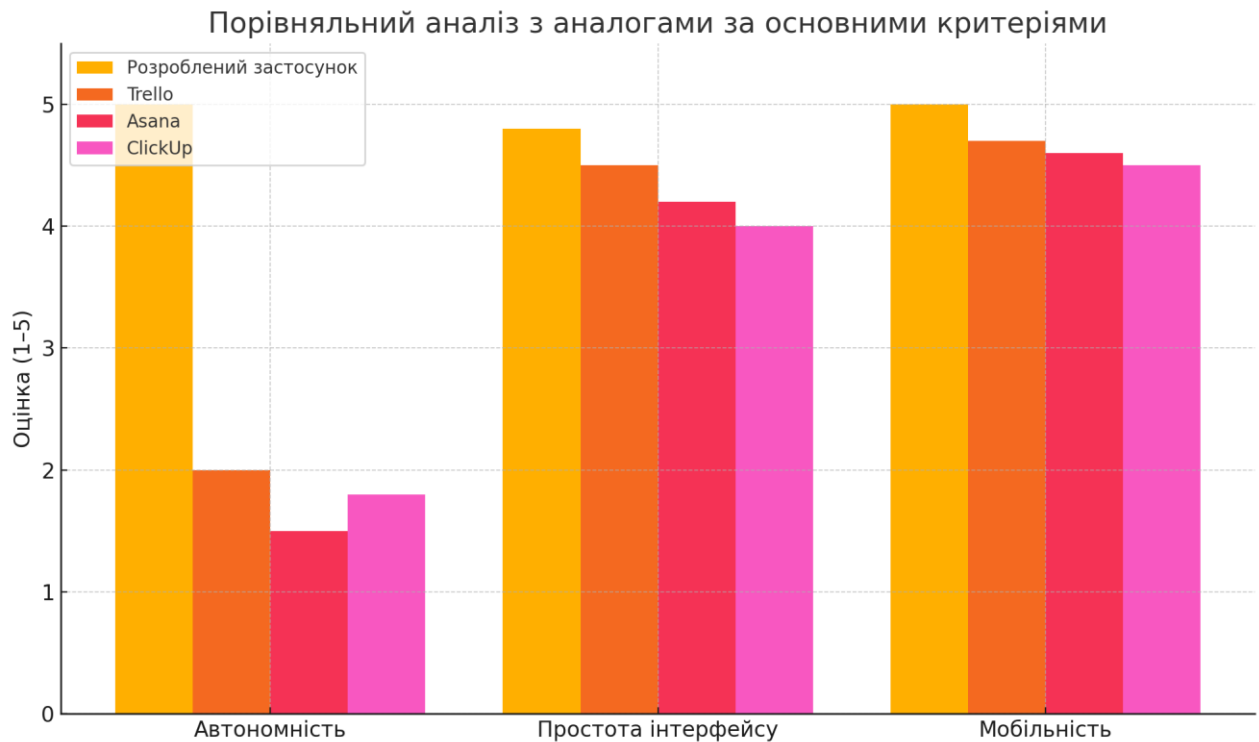


Рисунок 3.9 – Переваги розробленого застосунку над популярними аналогами за критеріями: автономність, простота інтерфейсу, мобільність.

Таблиця 3.1.

Мобільність, простота інтерфейсу у порівнянні з аналогами.

Критерій	Trello / Asana	Створений застосунок
Робота офлайн	Обмежена / ні	Повноцінна
Простота UI	Середня	Висока
Ресурсоспоживання	Високе	Низьке
Ціна	freemium	Безкоштовний
Локальне зберігання	ні	так

Підсумкові висновки щодо доцільності впровадження

Розроблений мобільний застосунок:

- виконує ключові функції організації особистих та командних завдань;
- забезпечує стабільність, безпеку та незалежність від зовнішніх платформ;

- є ефективним інструментом для управління навчальними, волонтерськими та малими ІТ-проектами.

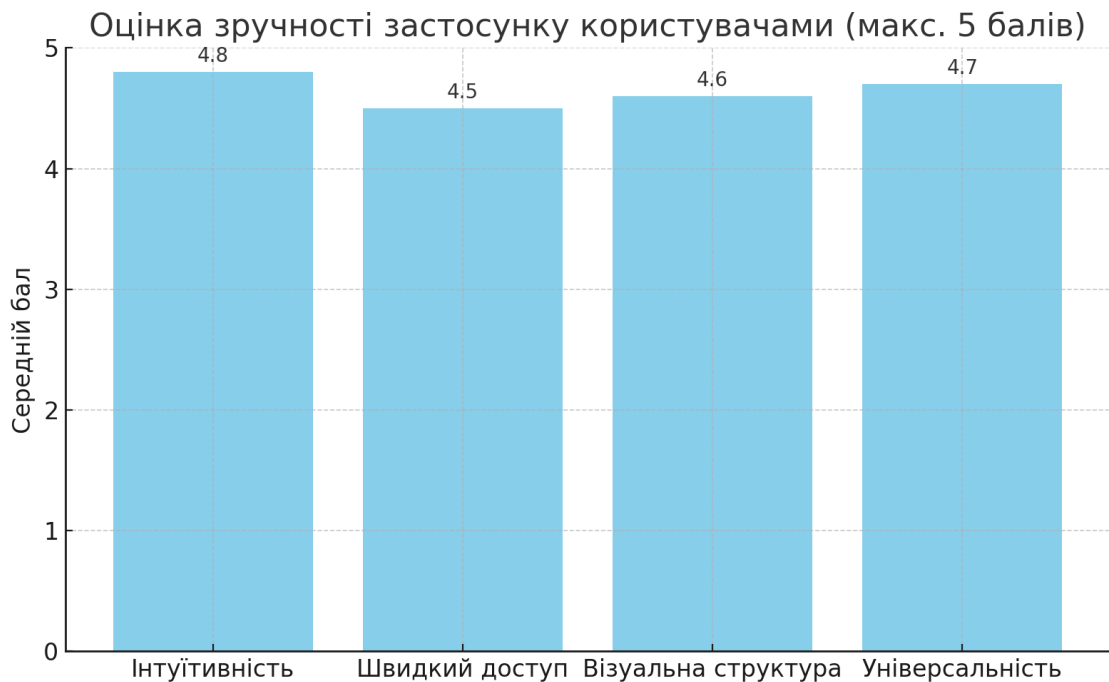


Рисунок 3.10 – Оцінка зручності мобільного застосунку користувачами за основними критеріями.

Доцільність впровадження підтверджено результатами тестування, позитивними відгуками користувачів та адаптивністю під різні потреби (рис.3.8).

Висновки до розділу 3.

Застосунок надає користувачам усі необхідні інструменти для повноцінного керування проектами: можливість створення, редагування та відстеження задач, контроль за дедлайнами та візуалізація прогресу виконання. Завдяки зручному, інтуїтивно зрозумілому інтерфейсу, користувач може швидко орієнтуватися в проектах і оперативно взаємодіяти з елементами керування.

Особливу увагу приділено системі збереження даних, яка гарантує надійне збереження всієї введеної інформації між сесіями роботи із

застосунком. Це дозволяє уникнути втрати даних та забезпечує безперервність процесу управління.

Додатково реалізоване графічне відображення задач – за допомогою індикаторів, діаграм чи візуальних маркерів, що значно спрощує сприйняття інформації про стан виконання кожного проєкту. Такий підхід сприяє кращому розумінню загальної картини й оперативному прийняттю рішень.

Даний розділ демонструє повноцінну реалізацію функціональності, передбаченої при проєктуванні застосунку, і підтверджує його придатність до практичного використання в ІТ-команді для щоденного керування проєктами.

ВИСНОВКИ

Кваліфікаційна робота детально описує особливості розробки мобільного застосунку для управління проєктами в ІТ-компанії, акцентуючи увагу на використаних технологіях та сучасних підходах до створення подібних систем. В центрі уваги – основна функціональність застосунку, яка включає можливість створення та редагування проєктів, призначення завдань відповідальним особам, контроль за виконанням задач і забезпечення зручного, інтуїтивно зрозумілого інтерфейсу для користувачів.

У процесі виконання кваліфікаційної роботи бакалавра було здійснено повний цикл розробки мобільного застосунку для управління проєктами ІТ-компанії. Розробка охоплювала етапи аналізу предметної області, постановки задачі, вибору інструментів розробки, архітектурного проєктування, програмної реалізації, а також створення інтерфейсу користувача та перевірки функціональності системи.

Актуальність теми зумовлена постійним зростанням попиту на засоби організації командної роботи, контролю задач, планування ресурсів та моніторингу прогресу в реальному часі. ІТ-компанії та інші організації потребують мобільних рішень, які дозволяють ефективно управляти проєктами з будь-якого пристрою незалежно від місця перебування.

У результаті виконання кваліфікаційної роботи було успішно розроблено мобільний застосунок для управління особистими проєктами та завданнями, що поєднує інтуїтивно зрозумілий інтерфейс із ефективною логікою обробки даних. Основна ідея – створення автономного інструменту, який працює без підключення до Інтернету, зосереджена на захисті даних користувача, стабільності функціонування та зручності у використанні.

В якості платформи для реалізації було обрано Android Studio – сучасне і функціональне середовище розробки, що дозволило використати Java як основну мову програмування. У проєкті застосовано ефективні засоби розробки мобільних застосунків: компоненти користувацького інтерфейсу, обробники

подій, сховища даних на основі `SharedPreferences`, а також бібліотеку `Gson` для збереження структури об'єктів.

Застосунок призначений для організації та контролю виконання завдань у малих і середніх командах розробників, а також може бути ефективно використаний окремими спеціалістами або фриланс-групами.

У ході дослідження було:

- проаналізовано предметну область та визначено ключові функціональні вимоги до системи;
- обґрунтовано вибір технологій: `Android Studio` як середовища розробки та `Java` як основної мови програмування;
- спроектовано архітектуру застосунку, включно з логікою взаємодії, структурою даних і користувацьким інтерфейсом;
- реалізовано інтерфейс взаємодії з користувачем, механізми створення, редагування та видалення задач і проєктів;
- забезпечено локальне збереження даних у форматі `JSON` без потреби серверної частини;
- впроваджено графічні компоненти, таймери, діалогові вікна та інші інтерактивні елементи;
- проведено тестування, яке засвідчило стабільну роботу застосунку та відповідність поставленим вимогам.

У роботі реалізовано архітектуру, що базується на централізованому контролері (**`MainActivity`**) із чітким розподілом між моделями, адаптерами й візуальними компонентами. Дані зберігаються у форматі `JSON` через **`SharedPreferences`**, що дозволяє зберігати й відновлювати стан програми навіть після її перезапуску.

Під час тестування застосунок продемонстрував високу стабільність роботи, швидкий відгук на дії користувача та правильність відображення й обробки даних. Проведено порівняльний аналіз із відомими аналогами, який

засвідчив конкурентоспроможність рішення за рахунок автономності, мобільності та простоти.

Результати дослідження та розробки підтверджують доцільність використання створеного програмного продукту в повсякденному плануванні завдань, як для особистих цілей, так і для навчальних або професійних потреб. Перспективи подальшого розвитку полягають у додаванні хмарної синхронізації, підтримки спільної роботи в командах, а також адаптації під різні платформи.

Результати тестування системи показали її стабільну роботу та відповідність функціональним вимогам. Застосунок зручно використовувати для організації особистої роботи, малих ІТ-команд або стартапів. Простота використання та автономність роботи є ключовими перевагами даної розробки.

Узагальнюючи, можна стверджувати, що поставлена мета дослідження була досягнута: розроблено та протестовано застосунок, який забезпечує ефективне управління проєктами та задачами на мобільному пристрої, відповідаючи сучасним вимогам до функціональності, простоти та автономності використання

Отримані результати підтверджують досягнення поставленої мети дослідження – створення функціональної, автономної та зручної у використанні мобільної інформаційної системи для управління ІТ-проєктами. Застосунок може бути основою для подальшого розширення функціональності, інтеграції з хмарними сервісами або впровадження серверної частини.

Отже, поставлені завдання кваліфікаційної роботи повністю виконані, а досягнуті результати відповідають меті дослідження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Android Developers. RecyclerView Overview [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide/topics/ui/layout/recyclerview>. – Назва з екрана. – Дата звернення: 01.11.2024.
2. Android Developers. SharedPreferences [Електронний ресурс]. – Режим доступу: <https://developer.android.com/reference/android/content/SharedPreferences>. – Назва з екрана. – Дата звернення: 25.11.2024.
3. Android Developers. Saving Data in Android [Електронний ресурс]. – Режим доступу: <https://developer.android.com/training/data-storage>. – Назва з екрана. – Дата звернення: 10.12.2024.
4. Google Developers. Gson User Guide [Електронний ресурс]. – Режим доступу: <https://github.com/google/gson/blob/main/UserGuide.md>. – Назва з екрана. – Дата звернення: 22.12.2024.
5. Android Developers. Dialogs [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide/topics/ui/dialogs>. – Назва з екрана. – Дата звернення: 05.01.2025.
6. Android Developers. Custom Views [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide/topics/ui/custom-components>. – Назва з екрана. – Дата звернення: 07.01.2025.
7. Oracle. The Java™ Tutorials [Електронний ресурс]. – Режим доступу: <https://docs.oracle.com/javase/tutorial/>. – Назва з екрана. – Дата звернення: 15.01.2025.
8. Android Developers. Canvas API [Електронний ресурс]. – Режим доступу: <https://developer.android.com/reference/android/graphics/Canvas>. – Назва з екрана. – Дата звернення: 18.02.2025.
9. Android Developers. Paint Class Reference [Електронний ресурс]. – Режим доступу:

<https://developer.android.com/reference/android/graphics/Paint>. – Назва з екрана. – Дата звернення: 19.02.2025.

10. Android Developers. View and ViewGroup [Електронний ресурс]. – Режим доступу: <https://developer.android.com/reference/android/view/View>. – Назва з екрана. – Дата звернення: 20.02.2025.

11. Google Codelabs. Build your first Android app [Електронний ресурс]. – Режим доступу: <https://developer.android.com/codelabs/build-your-first-android-app>. – Назва з екрана. – Дата звернення: 15.03.2025.

12. Vogella. Android SharedPreferences Tutorial [Електронний ресурс]. – Режим доступу: <https://www.vogella.com/tutorials/AndroidFileBasedPersistence/article.html>. – Назва з екрана. – Дата звернення: 18.03.2025.

13. Stack Overflow. RecyclerView Discussions [Електронний ресурс]. – Режим доступу: <https://stackoverflow.com/questions/tagged/recyclerview>. – Назва з екрана. – Дата звернення: 20.03.2025.

14. Birch J. Working with Custom Views in Android [Електронний ресурс] // Medium. – Режим доступу: <https://medium.com/@joebirch/working-with-custom-views-on-android-8c327c331f7b>. – Назва з екрана. – Дата звернення: 05.05.2025.

15. Raywenderlich.com. Android Canvas Drawing Tutorial [Електронний ресурс]. – Режим доступу: <https://www.raywenderlich.com/155-android-canvas-drawing-tutorial>. – Назва з екрана. – Дата звернення: 09.05.2025.

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Бакалавр**
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри комп'ютерних
систем та робототехніки
к. ф.-м. н., доц. ХРУСЛОВ М. М.
«02» жовтня 2024 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

ЗАЄЦ Ігор Іванович
(прізвище, ім'я, по батькові студента)

1. Тема роботи **«СИСТЕМА УПРАВЛІННЯ ПРОЄКТАМИ ДЛЯ ІТ-КОМПАНІЙ»**

керівник роботи **Мороз Ольга Юріївна, PhD з інформ. технологій, доцент з во
кафедри КСР**

затвержені наказом по університету від **16 квітня 2025 року №4101-5/962**

2. Строк подання студентом роботи ***30 травня 2025 року***

3. Перелік питань, які потрібно розробити)

- 1) Аналіз потреб ІТ-компанії щодо управління проєктами
- 2) Огляд існуючих рішень для управління проєктами
- 3) Архітектура мобільного застосунку на базі Android
- 4) Проєктування моделі зберігання даних
- 5) Розробка інтерфейсу користувача
- 6) Реалізація основного функціоналу системи управління проєктами
- 7) Інтеграція графічного відображення даних (діаграми, прогрес-бари)
- 8) Обробка збереження даних та забезпечення їх цілісності
- 9) Тестування функціоналу застосунку
- 10) Оцінка ефективності та перспективи розвитку

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Затвердження теми роботи	05.09.2024 – 02.10.2024
2	Аналіз літератури та програмного забезпечення у сфері управління ІТ-проектами	02.10.2024 – 24.10.2024
3	Огляд і аналіз існуючих систем управління проектами для мобільних платформ	25.10.2024 – 09.11.2024
4	Визначення функціональних вимог до системи управління проектами	10.11.2024 – 24.11.2024
5	Розробка архітектури та проектування застосунку	25.11.2024 – 08.12.2024
6	Реалізація основного функціоналу мобільного застосунку в Android Studio	09.12.2024 – 29.12.2024
7	Розробка графічного інтерфейсу користувача та налаштування взаємодії з базою даних	30.12.2024 – 31.01.2025
8	Тестування застосунку, аналіз результатів і оптимізація	01.02.2025 – 01.03.2025
9	Підготовка і оформлення звітних матеріалів. Написання статті за матеріалами кваліфікаційної роботи	01.03.2025 – 30.04.2025
10	Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи. Оформлення списку літератури	01.04.2025 – 30.04.2025
11	Оформлення пояснювальної записки кваліфікаційної роботи відповідно до вимог до звітів про НДР	01.05.2025 – 30.05.2025
12	Оформлення звіту про переддипломну практику	01.05.2025 – 30.05.2025
13	Представлення кваліфікаційної роботи керівнику та рецензенту	30.05.2025

5. Дата видачі завдання *02 жовтня 2024 року.*

Студент

Заєц І.І.

ініціали, прізвище

підпис

Керівник роботи

Мороз О. Ю.

ініціали, прізвище

підпис


**Технічне завдання на розробку програмного виробу
«Система управління проєктами для ІТ-компанії»**

Назва розділу	Назва та зміст підрозділу
1. Вступ	<p>1.1. Назва програмного виробу: «Система управління проєктами для ІТ-компанії»</p> <p>1.2. Галузь застосування: Мобільний застосунок призначений для управління ІТ-проєктами, координації задач, моніторингу прогресу виконання та планування робочого часу в невеликих командах або компаніях.</p>
2. Підстава для розробки	<p>2.1. Навчальний план за спеціальністю: 123 «Комп'ютерна інженерія»</p> <p>2.2. Завдання на кваліфікаційну роботу від 16 квітня 2025 року №4101-5/962: Розробка мобільного застосунку «Система управління проєктами для ІТ-компанії»</p>
3. Призначення розробки	<p>3.1. Мета розробки програмного виробу: Створити інтуїтивно зрозумілий інструмент для ефективного управління задачами, проєктами, контролю стану виконання, планування робочого часу.</p> <p>3.2. Призначення програмного виробу: Застосунок дозволяє створювати проєкти, додавати задачі, призначати пріоритети, статуси, дедлайни, візуалізувати прогрес виконання у вигляді графіків.</p> <p>3.3. Вихідні дані для розробки: Планування структури застосунку, використання компонентів Android SDK, збереження даних у SharedPreferences з використанням JSON-формату.</p>
4. Технічні вимоги до програмного виробу	<p>4.1. Функціональні вимоги:</p> <ul style="list-style-type: none"> – Додавання/редагування/видалення проєктів і задач; – Призначення параметрів задач: назва, опис, дедлайн, статус, пріоритет, оцінка годин; – Розрахунок прогресу виконання задач та відображення у графіку; – Збереження даних локально через SharedPreferences. <p>4.2. Вимоги до надійності:</p> <ul style="list-style-type: none"> – Стабільність застосунку при зміні орієнтації екрану, перезапуску; – Коректна робота при відсутності інтернету (офлайн-зберігання);


	<ul style="list-style-type: none"> – Автоматичне оновлення інтерфейсу при зміні статусу задач. <p>4.3. Вимоги до умов експлуатації:</p> <ul style="list-style-type: none"> – Android 9.0 і вище; – Мобільні пристрої з мінімум 2 ГБ ОЗП; – Доступність для запуску на емуляторі та фізичному пристрої. <p>4.4. Вимоги до складу параметрів технічних засобів:</p> <ul style="list-style-type: none"> – Android Studio 2023 або вище; – Пристрій з Android SDK API 28+; – Екран з мінімальним розширенням 720x1280. <p>4.5. Вимоги до програмної сумісності:</p> <ul style="list-style-type: none"> – Java 8, Android SDK; – Gson для серіалізації даних; – SharedPreferences API для збереження інформації; – Canvas API для графіків. <p>4.6. Вимоги до маркування та упаковки:</p> <ul style="list-style-type: none"> – Застосунок цифровий, не потребує фізичної упаковки. <p>4.7. Вимоги до транспортування та зберігання:</p> <ul style="list-style-type: none"> – Передача архівом або через GitHub; – Зберігання на сервері або хмарному сховищі. <p>4.8. Спеціальні вимоги:</p> <ul style="list-style-type: none"> – Повна автономність застосунку; – Можливість масштабування на більші команди або інтеграції з хмарним бекендом.
5. Вимоги до програмної документації.	<ol style="list-style-type: none"> 1. Технічне завдання (Додаток Б); 2. Методика випробувань (Додаток В); 3. Опис застосунку (розділ 2, розділ 3 пояснювальної записки); 4. Текст коду (Додаток Г).
6. Техніко-економічні показники	<ol style="list-style-type: none"> 1. Зменшення часу керування проектами на 25–40%; 2. Продуктивність на мобільних пристроях початкового рівня; 3. Безкоштовне розповсюдження, використання тільки відкритих бібліотек; 4. Час розробки – до 3 місяців студентом одного складу
7. Стадії та етапи розробки	<p>Стадії:</p> <ul style="list-style-type: none"> – Формування технічного завдання; – Реалізація логіки застосунку; – Тестування і підготовка до впровадження. <p>Етапи:</p> <ul style="list-style-type: none"> – Розробка UI/UX макетів та логіки даних; – Збереження та обробка даних;

	<ul style="list-style-type: none">– Підключення графіки Canvas;– Оформлення документації;– Презентація та тестування в реальному середовищі.
8. Порядок контролю та приймання	<ol style="list-style-type: none">1. Регулярний контроль керівника (раз на 3 тижні);2. Тестування на пристрої;3. Приймання результатів під час передзахисту;4. Захист роботи на засіданні ЕК.

Виконавець
студент групи КІ- 41
Заєц І.І.



Замовник
Phd, доцент з во каф.КСР
Мороз О. Ю.



Програма та методика випробувань програмного виробу «Система управління проєктами для ІТ-компанії»

1. Об'єкт випробувань

1.1 Найменування програмного виробу: "Система управління проєктами для ІТ-компанії".

1.2 2. Галузь застосування: Система призначена для використання в ІТ-компаніях, які потребують мобільного інструменту для управління проєктами, задачами, контролю за прогресом та плануванням робочих процесів.

1.3 Умовне призначення розробки: Програмний виріб забезпечує створення, редагування, перегляд і контроль за виконанням задач у межах проєктів. Він дозволяє користувачам зручно керувати проєктною діяльністю з мобільного пристрою, включаючи моніторинг дедлайнів, пріоритетів та статусів задач, а також візуалізацію прогресу за допомогою графічних компонентів.

2. Мета випробувань

Метою випробувань є підтвердження відповідності мобільного застосунку встановленим функціональним і технічним вимогам. Зокрема, перевіряється здатність програми стабільно працювати на різних версіях Android, зберігати й відновлювати дані користувача, правильно обробляти задачі та проєкти, а також надавати користувачеві візуальний зворотний зв'язок про прогрес виконання. Випробування дозволяють оцінити надійність, адаптивність інтерфейсу, коректність обробки подій користувача, зручність взаємодії, а також стійкість до помилок і непередбачених сценаріїв взаємодії.

3. Загальні положення

3.1 Підстави щодо випробувань

Підставою для проведення випробувань є: наказ ХНУ №№4101-5/962 від 16 квітня 2025 року про затвердження тем кваліфікаційних робіт бакалаврів.

3.2 Місце та тривалість випробувань

Приймальні випробування проводяться особисто виконавцем.

3.3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідно до цієї Програми та методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться за участю Замовника та Виконавця.

4. Вимоги до програмного виробу

- Можливість створення, редагування та видалення проєктів;
- Додавання задач до кожного проєкту із зазначенням дедлайну, опису, статусу, пріоритету та кількості годин;
- Візуалізація прогресу виконання задач та проєкту загалом;
- Збереження та завантаження даних локально за допомогою SharedPreferences у форматі JSON;
- Наявність графіка задач із використанням Canvas;
- Стабільність роботи застосунку при повторних запусках та оновлення

5. Вимоги до програмної документації

1. Технічне завдання (представити у вигляді Додатка Б);
2. Програма і методика випробувань (цей документ, Додаток В);
3. Опис програмного виробу (розділи 2–3 пояснювальної записки);
4. Текст програми (представити в Додатку Г).

6. Засоби та порядок випробувань

6.1 Засоби випробувань:

- Android Studio 2023;
- Емулятор Android API 30+ або реальний пристрій (телефон з Android 9.0+);
- ADB logcat – для перегляду логів;
- JSON Viewer – для перевірки збережених файлів;
- Вбудовані інструменти Android Profiler для моніторингу продуктивності.

6.2 Порядок проведення випробувань:

1-й підетап. Перевірка програмної документації – наявність технічного завдання, опису функцій, коду, інструкції користувача.

2-й підетап. Запуск застосунку в Android Studio або на реальному пристрої. Очікуваний результат: застосунок відкривається, відображається основний екран з порожнім списком або наявними проектами.

3-й підетап. Створення проекту та задач Очікуваний результат: користувач має змогу створити новий проект, додати задачі, зберегти й переглянути їх у наступному сеансі.

4-й підетап. Редагування даних, зміна статусів задач, оновлення прогресу Очікуваний результат: **зміни зберігаються, статус задач впливає на візуальний прогрес.**

5-й підетап. Візуалізація даних (Canvas View) Очікуваний результат: графік задач або прогресу відображається коректно, компоненти масштабуються відповідно до розміру екрана.


6-й підетап. Тестування на стабільність (кількаразовий запуск, навігація, очищення кешу, зміна орієнтації екрану) Очікуваний результат: застосунок не аварійно завершується, дані не втрачаються.

7. Методики випробувань

- Тестування функціональних можливостей по сценаріях користувача;
- Мануальне введення типових і граничних значень у поля введення;
- Аналіз збереженого JSON-файлу за допомогою JSON Viewer;
- Перевірка коректності оновлення інтерфейсу при зміні даних;
- Візуальне тестування адаптивності елементів до різних екранів

8. Висновки за результатами випробувань

Висновки за результатами випробувань Мобільний застосунок успішно пройшов усі етапи тестування. Виявлено відповідність заявленим функціональним вимогам. Застосунок працює стабільно, зберігає дані локально, забезпечує належний рівень інтерактивності інтерфейсу та є зручним у користуванні. Програмний виріб готовий до використання та подальшого розширення функціоналу.

Виконавець: Заєц Ігор Іванович 

ФРАГМЕНТ ЛІСТИНГУ КОДУ

```
public class MainActivity extends AppCompatActivity {
    private RecyclerView projectsRecyclerView;
    private ProjectAdapter projectAdapter;
    private List<Project> projects;
    private Button addProjectButton;
    private SharedPreferences sharedPreferences;
    private Gson gson;
    private Handler timerHandler;
    private Runnable timerRunnable;

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

        initializeViews();
        setupData();
        setupRecyclerView();
        setupClickListeners();
        loadProjects();
        startDeadlineTimer();
    }

    private void initializeViews() {
        projectsRecyclerView = findViewById(R.id.projectsRecyclerView);
        addProjectButton = findViewById(R.id.addProjectButton);
        sharedPreferences = getSharedPreferences("ITProjectManager",
Context.MODE_PRIVATE);
        gson = new Gson();
        timerHandler = new Handler(Looper.getMainLooper());
    }

    private void setupData() {
        projects = new ArrayList<>();
    }

    private void setupRecyclerView() {
        projectAdapter = new ProjectAdapter(projects, this);
        projectsRecyclerView.setLayoutManager(new LinearLayoutManager(this));
        projectsRecyclerView.setAdapter(projectAdapter);
    }

    private void setupClickListeners() {
        addProjectButton.setOnClickListener(v -> showAddProjectDialog());
    }

    private void loadProjects() {
```

```

String json = sharedPreferences.getString("projects", "");
if (!json.isEmpty()) {
    Type type = new TypeToken<List<Project>>() {}.getType();
    List<Project> loadedProjects = gson.fromJson(json, type);
    if (loadedProjects != null) {
        projects.clear();
        projects.addAll(loadedProjects);
        projectAdapter.notifyDataSetChanged();
    }
}

private void saveProjects() {
    String json = gson.toJson(projects);
    sharedPreferences.edit().putString("projects", json).apply();
}

private void startDeadlineTimer() {
    timerRunnable = () -> {
        checkDeadlines();
        projectAdapter.notifyDataSetChanged();
        timerHandler.postDelayed(timerRunnable, 60000); // every 60 sec
    };
    timerHandler.post(timerRunnable);
}

private void checkDeadlines() {
    Date now = new Date();
    for (Project project : projects) {
        for (Task task : project.getTasks()) {
            if (!"Завершена".equals(task.getStatus())) {
                long timeDiff = task.getDueDate().getTime() - now.getTime();
                task.setTimeRemaining(timeDiff);
            }
        }
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (timerHandler != null && timerRunnable != null) {
        timerHandler.removeCallbacks(timerRunnable);
    }
}
}

```