

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
Теплоенергетичний факультет**

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр Коваль

«\_\_» \_\_\_\_\_ 2020 р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інформаційні технології моніторингу  
довкілля»**

**спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

**на тему: «Програмний додаток групового управління проектами на базі  
операційної системи Android»**

Виконав :

студент ІV курсу, групи ТМ-61

Жицький Андрій Михайлович \_\_\_\_\_

Керівник:

д.т.н., професор,

Отрох Сергій Іванович \_\_\_\_\_

Рецензент:

к.т.н., доцент,

Зенів Ірина Онуфріївна \_\_\_\_\_

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

Студент \_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 122 Комп'ютерні науки та інформаційні технології

Спеціалізація Інформаційні технології моніторингу довкілля

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр Коваль

(підпис)

” \_\_\_ ” \_\_\_\_\_ 2020р.

**ЗАВДАННЯ  
на дипломну роботу студенту**

\_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Програмний додаток групового управління проектами на базі операційної системи Android

керівник роботи д.т.н., професор Отрох Сергій Іванович

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

затверджена наказом вищого навчального закладу від "25" травня 2020р.

№ **1168-с**

2. Строк подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи мова програмування Java, мова програмування PHP, середовище розробки Android Studio, система керування реляційними базами даних MySQL

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз задачі створення програмного додатку для реалізації групового управління проектами на базі операційної системи Android, Аналіз проблеми групового управління проектами, Аналіз та оцінка існуючих програмних рішень. Вибір засобів розробки. Розробка програмної системи для реалізації групового управління проектами



## **АНОТАЦІЯ**

Метою роботи було створення програмного додатку для групового управління проектами, а саме для перегляду списку проектів, багаторівневої структури завдань на кожному проекті, додавання та видалення завдань. Проект створено в середовищі розробки Android Studio, з використанням мови програмування Java.

Записка містить 77 сторінок, 45 рисунків, 7 посилань і 3 додатки.

Ключові слова: ПРОЕКТНИЙ МЕНЕДЖЕР, ОС ANDROID, JAVA, УПРАВЛІННЯ ЗАДАЧАМИ, ANDROID STUDIO.

## **ABSTRACT**

The purpose of the work was to create a software application for group project management, namely to view the list of projects, multi-level task structure on each project, add and delete tasks. The project created in a development environment Android Studio, using the programming language Java.

The note contains 77 pages, 45 images, 7 references and 3 applications.

Keywords: PROJECT MANAGER, OS ANDROID, JAVA, TASK MANAGEMENT , ANDROID STUDIO.

# Зміст

Вступ.....	6
1. Задача створення програмного додатку для реалізації групового управління проектами на базі операційної системи Android .....	8
1.1 Призначення та основні функції програмного продукту.....	8
1.2 Потенційні користувачі системи .....	9
2. Аналіз проблеми групового управління проектами .....	10
2.1 Менеджер проектів Microsoft Project.....	10
2.2 Менеджер завдань Trello .....	11
2.3 Менеджер проектів Asana .....	11
3. Засоби розробки.....	13
3.1 Середовище розробки Android Studio.....	13
3.2 Система керування реляційними базами даних MySQL .....	16
3.4 Мова програмування PHP .....	18
3.3 Мова програмування Java .....	19
4. Опис програмної реалізації .....	21
4.1 Фізична модель даних.....	21
4.2 Структура таблиць бази даних .....	20
4.3 Класи і методи програмного продукту .....	26
4.4 Функціональна модель програмного забезпечення.....	35
5. Робота користувача з програмною системою .....	36
5.1 Системні вимоги та інсталяція .....	36
5.2 Робота користувача з програмним продуктом.....	37
Висновки .....	49
Список використаних джерел .....	50

## Вступ

Управління проектом є невід’ємною складовою успішності бізнес процесів. Застосування різноманітних прийомів та інструментів дозволяє досягти поставлених проектних цілей і задовільнити потреби клієнтів. Важко уявити сучасну організацію, котра не використовує проектне планування та менеджмент задач на проекті.

Більшість проектних менеджерів зосередженні на максимальній підтримці конкретного проекту і надають його учасникам широкий набір функцій для організації проектної роботи. У свою чергу менеджери проектів отримують певний перелік необхідних інструментів для організації розподілення завдань та делегування ролей серед учасників команди.

Проте все частіше у зв’язку з розширенням ринку і значним зростанням кількості проектів які виконують лідируючі світові компанії, все частіше виникає потреба у груповому управлінні проектами.

Окрім зовсім очевидної функції підвищення продуктивності керівників проекту, а також створення комфортних умов задля забезпечення максимального управління часом; групове управління також сприяє більш ефективному проектному планування, а також дозволяє заощадити значну кількість різноманітних ресурсів, як людських, так і матеріальних.

Основним завданням було створити ефективний програмний додаток для групового управління проектами, який би надавав необхідний набір функцій як звичайним учасникам, так і менеджерам проектів.

Для розробки системи було використано середовище розробки Android Studio, мову програмування Java. Серверна частина виконана на мові програмування PHP. База даних створена за допомогою системи управління MySQL, а для полегшення роботи з нею було застосовано сервіс адміністрування phpMyAdmin.

Додаток працює на пристроях з операційною системою Android версією не нижче ніж Jelly Bean 4.3.

Записка містить п'ять розділів. У першому розділі описано задачу створення програмного додатку для реалізації групового управління проектами.

У другому розділі проведено аналіз проблеми групового управління проектами та аналіз існуючих аналогічних систем.

У третьому розділі показані основні засоби розробки системи, їх призначення в програмних модулях.

У четвертому розділі описано розроблений програмний продукт та його компоненти.

У п'ятому розділі зазначені вимоги до апаратного забезпечення та операційної системи користувача, а також показано його роботу з програмою.

# **1. Задача створення програмного додатку для реалізації групового управління проектами на базі операційної системи Android**

Ознайомившись з літературою та дослідивши аналогічні системи, розробити програмний продукт для реалізації групового управління проектами на базі операційної системи Android. Система повинна надавати користувачам доступ до поставлених їм завдань у проектах, у яких вони приймають участь, а також надавати менеджерам проектів функції управління завданнями і розподілення людських ресурсів.

## **1.1. Призначення та основні функції програмного продукту**

Програмний продукт повинен враховувати багаторівневі завдання проекту і надавати одночасний доступ для багатьох учасників і керівників проекту на кожному рівні завдань. Кожен користувач системи може бути задіяний у декількох проектах одночасно. Менеджери проектів повинні мати змогу керувати завданнями багатьох проектів, а також бути учасниками інших проектів без загальних функцій управління.

Основні функції системи:

- автентифікація задля забезпечення інформаційної безпеки;
- розподілення прав користувачів і рівнів доступу до проектів і завдань;
- доступ користувачів до особистої інформації в системі;
- надання даних про поточні проекти кожному користувачу;
- надання даних про поточні завдання кожного проекту відповідно;
- забезпечення вищого рівня доступу для менеджерів проектів;
- реалізація управління завданнями на кожному проекті;
- редагування статусів завдань.

Основною метою є створення комфортного і доступного програмного продукту на базі операційної системи Android, щоб забезпечити користувачам зручний

перегляд всіх своїх завдань на кожному проєкті, а також підвищити продуктивність менеджерів проєктів і покращити рівень управління груповими задачами.

## **1.2. Потенційні користувачі системи**

Система переважно розрахована на ІТ компанії, які зацікавлені в управлінні проєктними завданнями, а також мають необхідність розподілення великої кількості людських ресурсів на декількох проєктах одночасно. Проте програмний продукт може бути також ефективним і для невеликих компаній, задля підвищення продуктивності праці і рівня зосередженості і концентрації працівників.

Поза межами ІТ ринку такий додаток теж має значну актуальність, адже він спрямований на бізнес-проєкти і розростання ринку в цілому. Тож така система може бути впроваджена у будь-які організації задля корегування поточної системи управління або ж встановлення нової.

Продукт може бути застосовано в закладах освіти та науки з метою делегування зобов'язань і завдань для викладачів та працівників, а також як програмне рішення для оптимізації своїх ресурсів, як часових так і матеріальних.

Наявність додатку на кожному новітньому гаджеті повинна сприяти підвищенню ефективності менеджменту проєктів і широкому розповсюдженні серед сучасних провідних організацій. Саме операційна система Android і сумісність з багатьма пристроями будуть забезпечувати велику кількість користувачів системи необхідними функціональними інструментами, без використання персональних комп'ютерів і ноутбуків, що дозволить використовувати додаток також і на виробництвах, промислових підприємствах і компаніях у яких неможливо або ж проблематично надати всім працівникам окремий стаціонарний комп'ютер.

## 2. Аналіз проблеми групового управління проектами

Наразі існує багато програмних рішень для управління проектами на базі операційних систем Windows та Linux, але окрім того що користувач не може отримати доступ до таких систем зі свого мобільного чи планшету, ці програмні продукти також у більшості випадків доступні лише у варіанті дуже дорогих Premium пакетів, розрахованих на великі корпорації з великою кількістю потенційних користувачів системи, що робить їх недоступними для організацій середнього і малого бізнесу.

Розглянемо найбільш відому і популярну систему MS Project від світового лідера програмних продуктів компанії Microsoft.

### 2.1. Менеджер проектів Microsoft Project

Microsoft Project (MS Project) — це комплексне програмне забезпечення — система управління проектами і спосіб оптимізації управління портфелями проектів, який дозволяє планувати і контролювати проектну діяльність організацій. Для цього застосовуються вбудовані шаблони, інструменти для різних рівнів аналітики і статистики, засоби управління робочим часом. MS Project найчастіше використовується як програмне рішення для середнього та крупного бізнесу.

Microsoft пропонує своїм користувачам декілька варіантів ліцензії для отримання цього продукту. Варто також зазначити, що до стандартної підписки Office 365 (від недавно Microsoft 365) MS Project не входить навіть на рівні Business преміум, і корпорації змушені купувати ліцензію окремо.

Варіанти ліцензування:

- Project Plan 3 — \$30.00 з користувача за місяць;
- Project Plan 5 (розширені функції) — \$55.00 з користувача за місяць;
- Project Standard 2019 — постійна ліцензія за \$620.00 з користувача;

— Project Professional 2019 — постійна розширена ліцензія за \$1030.00 з користувача;

Наразі MS Project не має свого програмного рішення для мобільних пристроїв на базі операційної системи Android, проте користувачі можуть скористатися веб-версією системи з обмеженим функціоналом.

## **2.2. Менеджер завдань Trello**

Якщо ж говорити про системи для мобільних пристроїв та планшетів, варто розглянути Trello — менеджер завдань для організації власного часу, а також створення колективних задач. Даний програмний продукт розробила компанія Fog Creek Software, позиціонувавши його як багатоплатформне рішення для управління проектами і задачами. Trello доступний для Android, iOS і Windows, також можна скористатися веб-версією.

Більшість функцій спрямована саме на особисте користування, тоді як робота в проектах навіть для платних версій має досить обмежений функціонал. Окрім того відсутні рівні задач і розподілення ролей, менеджери проектів не мають окремого рівня доступу, а робота в декількох проектах одночасно не передбачена.

Trello передбачає як безкоштовний варіант для особистого користування і користування в командах до 10 користувачів, так і платні версії, а саме:

— Business Class — \$ 9.99 з користувача за місяць;

— Enterprise — \$ 20.83 з користувача за місяць;

Trello ідеально підходить для одного користувача, або ж для розробки певних продуктів у невеликих колективах для оптимізації часу.

## **2.3. Менеджер проектів Asana**

Ще один з програмних додатків для управління проектами на базі операційних систем Android та iOS це Asana від однойменної компанії-розробника. Система була

створена колишніми працівниками компанії Facebook Дастином Московіцем і Джастіном Розенштейном.

Asana позиціонується як розширений планувальник завдань для їх виконання в командах. На відміну від Trello, тут присутні певний функціонал для забезпечення зв'язку між користувачами. Окрім того додаток пропонує і особисті списки завдань. Попри все рівні доступу все ще погано організовані, тоді як багаторівневі завдання недоступні.

Asana присутня як у безкоштовному варіанті до 15 користувачів, так і в розширених варіантах, зокрема із функцією пріоритетів завдань і проекту.

Ліцензії Asana:

- Premium Plan — \$ 9.99 з користувача за місяць;
- Business Plan — \$ 19.99 з користувача за місяць;
- Enterprise Plan — \$ 29.99 з користувача за місяць;

Система не передбачає групового управління проектами, але надає певні функції для менеджерів проектів, за допомогою яких вони можуть управляти діяльністю одного проекту з делегуванням зобов'язань і розподіленням робочого часу, до того ж модель Kanban (Канбан) тут реалізована на значному рівні, що надає змогу ефективно організувати дошки завдань і розроблювати проектні цілі та розставляти пріоритети.

### 3. Засоби розробки

Під час розробки системи було використано середовище розробки Android Studio, як універсальне середовище для створення програмних продуктів для операційної системи Android. Для програмування клієнтського програмного модуля використано об'єктно-орієнтовану мову програмування Java. Користувацький інтерфейс створено за допомогою Розширюваної Мови Розмітки (XML). Для налаштування запитів до серверу використано бібліотеку `org.apache.http.legacy`.

Серверна частина написана за допомогою мови програмування PHP. Для створення та керування реляційною базою даних було використано систему керування MySQL на базі сервісу адміністрування phpMyAdmin. Керування локальним сервером здійснюється за допомогою універсальної збірки веб-серверу XAMPP, з використанням програмного забезпечення Apache.

Для забезпечення стабільного підключення до серверу з боку клієнта використано бібліотеку `android.os.AsyncTask`. Анімації створені з використанням бібліотеки `animation.LinearInterpolator`. Для побудови дерев інтегровано бібліотеки `atv.model.TreeNode` і `atv.view.AndroidTreeView`.

Передача даних між сервером і клієнтом відбувається за допомогою файлів формату JSON, з використанням JSON бібліотеки для Android — `org.json.JSONObject`, яка дозволяє організувати роботу з JSON об'єктами на стороні клієнта.

#### 3.1. Середовище розробки Android Studio

Android Studio — це інтегроване середовище ( IDE ) для розробки додатків на базі операційної системи Android. Воно створене на основі IntelliJ IDEA, інтегрованому Java-середовищі та включає в себе різноманітні інструменти для користувача, за допомогою яких реалізовано функції редагування коду та розробки.

Загальний вигляд середовища показано на рисунку 3.1.1.

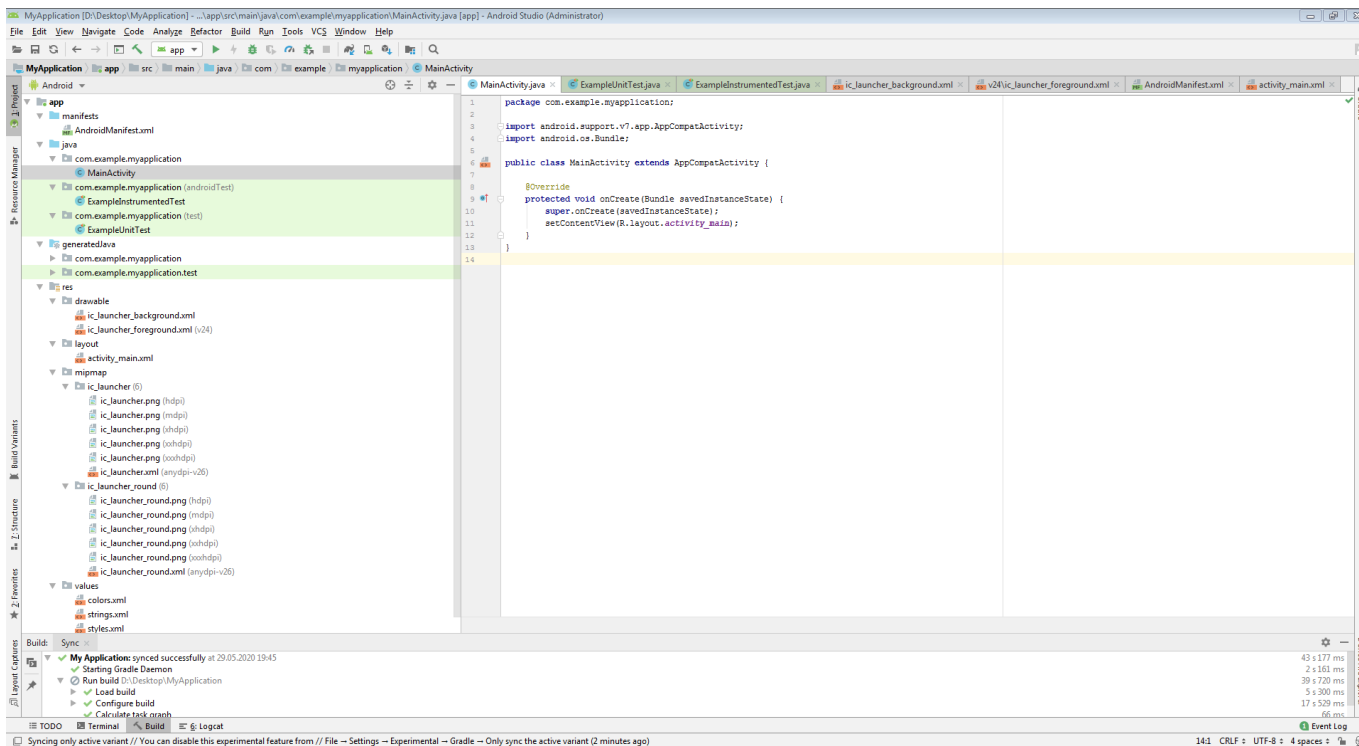


Рисунок 3.1.1 — Загальний вигляд Android Studio

Для забезпечення підтримки комфортної розробки додатків для операційної системи Android, Android Studio використовує універсальну систему побудови додатку на основі модуля Gradle, велику кількість емуляторів, надає широкий вибір шаблонів коду для полегшення процесу розробки та інтеграцію до сервісу Github. Кожен проект, що створюється в Android Studio має один або декілька модулів із вихідним кодом та файлами ресурсів. Ці модулі включають модулі додатків, модулі бібліотек та модулі Google App Engine.

Android Studio використовує функцію Instant Push, щоб передавати будь-які зміни коду, програмних модулів та ресурсів до працюючої програми. Редактор коду допомагає розробнику в написанні коду та надає доступ до компіляції, рефракції та аналізу коду. Програми, вбудовані в Android Studio, компілюються у формат APK для подальшого завантаження до Google Play Store.

Програмне забезпечення вперше було оголошено в Google I/O в травні 2013 року, а перша стабільна збірка була випущена в грудні 2014 року. Android Studio доступний для настільних платформ Mac, Windows та Linux. Він замінив Eclipse Android Development Tools (ADT) як основний IDE для розробки програм Android.

Android Studio та комплект розробки програмного забезпечення можна завантажити безпосередньо з Google.

У Android Studio використовується підхід до побудови додатків за допомогою Activities. Activity — це клас, який призначений для створення візуального екрану, що буде відображений на екрані пристрою користувача. Activities мають так званий стек, який забезпечує життєвий цикл програмного додатку, відповідно з відкриттям нового Activity, попереднє не закривається повністю, а залишається в оперативній пам'яті пристрою, тоді як картинка на екрані замінюється. Такий підхід дозволяє передавати дані між Activities, зберігати вибори користувача, дає змогу повертатися до попередній екранів і тому подібне.

Саме специфічна побудова програмного додатку і зумовила використання в середовищі розробки клас Activity. При створенні додатків розробники наслідують всі свої класи або від класу Activity, або від класів-фрагментів. Activity дозволяє додатку завжди враховувати поточні особливості пристрою, такі як мова пристрою, орієнтація екрану, версія програмного забезпечення, сенсори і служби.

Android Studio пропонує розробку як на мові програмування Java, так і на мові Kotlin — одній з найбільш сучасних і прогресивних мов для операційної системи Android, яка враховує всі недоліки Java і створює нові можливості у проектуванні сучасних і безпечних додатків. Основними методами для розробки користувацького інтерфейсу є View і Widget. За допомогою цих інструментів робота з дизайном і стилізацією програмних продуктів значно спрощується, тоді як конструювання і налаштування функцій користувача стає набагато легшим завданням для розробника.

Використання модуля Firebase реалізує прямі оновлення і безпосереднє підключення до різноманітних баз даних, що підвищує якість програмних продуктів і розширює функціонал всіх створених у Android Studio додатків.

Android Studio надає розробникам широкий функціонал для тестування та компіляції своїх систем, зокрема й віртуальні емулятори мобільних пристроїв та планшетів (рисунки 3.2.2), за допомогою яких можна протестувати роботу свого додатку на будь-якій доступній версії операційної системи Android, а також на будь-якій з найпоширеніших моделей пристроїв.

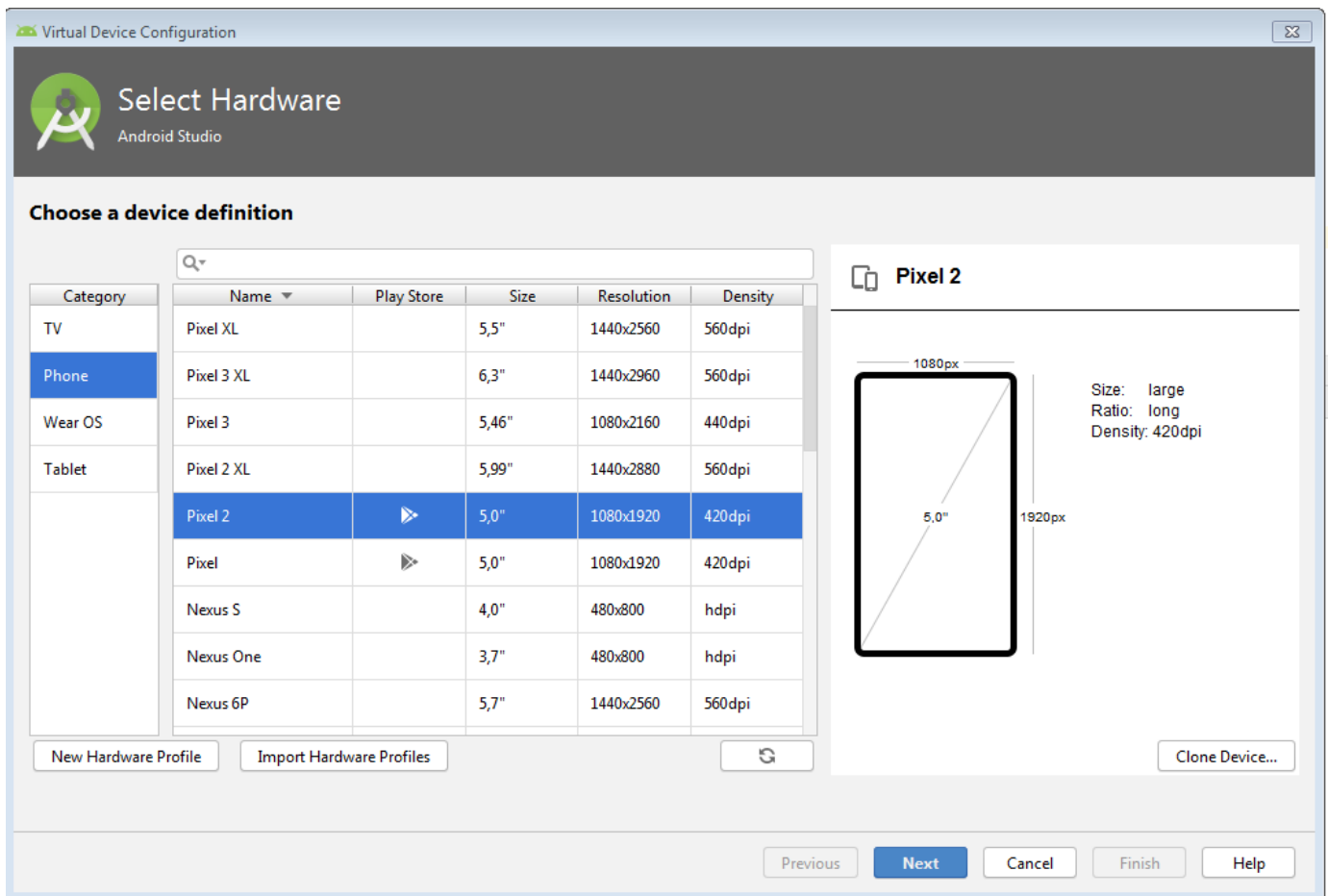


Рисунок 3.1.2 — Перелік віртуальних емуляторів

Тестування за допомогою таких інструментів дозволяє ще на стадії розробки виявити та усунути всі можливі помилки та неточності, що підвищує якість продуктів, які створені за допомогою Android Studio. Також такий підхід дає можливість програмувати без використання пристрою на базі операційної системи Android.

### 3.2. Система керування реляційними базами даних MySQL

Система управління базами даних MySQL була створена компанією ТсХ, для спрощення керування об'ємними базами даних, так підвищення швидкості доступу до них.

Основні процеси, що відбуваються в середовищі: MySQL створює сховище для зберігання та подальшої обробки даних, визначаючи внутрішні та зовнішні

взаємозв'язки таблиць між собою; клієнти можуть робити запити, шляхом введення спеціальних конкретних операторів на мові запитів SQL, потім ці операції надсилаються до MySQL; серверна програма обробляє подані запити і відправляє запитувану інформацію, яка з'являється на стороні клієнтів. MySQL підтримує широкий масив кластерних серверів. Незалежно від того, чи зберігаєте ви велику кількість великих електронних комерційних даних або займаєтесь великими бізнес-проектами, MySQL може допомогти вам безперебійно з оптимальною швидкістю. Промисловості вже багато років використовують MySQL, а це означає, що є достатньо ресурсів для кваліфікованих розробників. Користувачі MySQL можуть сподіватися на швидку розробку програмного забезпечення та експертів-фрілансерів, які бажають працювати за меншу заробітну плату, якщо вони коли-небудь потребуватимуть їх. Захищеність ваших даних повинні бути головним пріоритетом при виборі потрібного програмного забезпечення RDBMS. Завдяки системі з розподіленням прав доступу та керуванню обліковими записами користувачів, MySQL встановлює високу планку безпеки. Доступна перевірка на основі хоста та шифрування паролем.

Задля полегшення взаємодії з MySQL часто використовують різноманітні автоматизовані сервіси з адміністрування ваших баз даних. Одним з таких популярних сервісів є phpMyAdmin — безкоштовний програмний інструмент, написаний на PHP, призначений для управління та адміністрування MySQL через Інтернет. phpMyAdmin підтримує широкий спектр операцій на MySQL та MariaDB. Часто використовувані операції (керування користувачами, дозволами тощо) можуть виконуватися через користувацький інтерфейс.

phpMyAdmin постачається з широким спектром документації, і користувачі можуть оновлювати наші вікі-сторінки, щоб ділитися ідеями та практичними рекомендаціями для різних операцій. Команда phpMyAdmin спробує допомогти вам, якщо ви зіткнулися з будь-якою проблемою; ви можете використовувати різні канали підтримки, щоб отримати допомогу. Багатофункціональність і простота є основними перевагами для користувачів, що створено за рахунок великої кількості серверів у різних регіонах світу та наявності гнучкого набору індивідуальних налаштувань для

кожного окремого адміністратора або розробника. Саме тому сервіс phpMyAdmin так часто використовується розробниками для спрощення роботи з базою даних.

### 3.3. Мова програмування PHP

PHP стартував як невеликий проект з відкритим кодом, який розвивався, коли все більше людей дізнавалися, наскільки це корисно. Расмус Лердорф випустив першу версію PHP ще в 1994 році. PHP - рекурсивна аббревіатура для “PHP: Hypertext Preprocessor”. PHP - це сценарна мова на сервері, яка вбудована в HTML. Він використовується для управління динамічним вмістом, базами даних, відстеженням сеансів, навіть для створення цілих сайтів електронної комерції. Він інтегрований з низкою популярних баз даних, включаючи MySQL, PostgreSQL, Oracle, Sybase, Informix та Microsoft SQL Server. PHP дуже швидко виконує свої завдання, особливо у вигляді модуля Apache на стороні Unix. Після запуску сервер MySQL виконує навіть дуже складні запити з величезними наборами результатів за час встановлення записів. PHP підтримує велику кількість основних протоколів, таких як POP3, IMAP та LDAP. PHP4 додав підтримку Java та архітектури розподілених об'єктів (COM та CORBA), що зробило можливість розвитку n-ярусу.

Наразі PHP все ще знаходиться на вершині світу веб-розробок. Значна частка всіх веб-сайтів працює на цій мові програмування, а простий у користуванні CMS Wordpress продовжує набирати популярність. Це дозволяє користувачам розробляти та розміщувати власні веб-сайти на безкоштовних або платних хостингових платформах лише за кілька простих кроків.

Основними перевагами PHP є:

- Простота;
- Ефективність;
- Безпека;
- Гнучкість;
- Функціональність.

RНР виконує системні функції, тобто може створювати, відкривати, читати, писати та закривати файли в системі. RНР може обробляти форми, тобто збирати дані з файлів, зберігати дані у файл, через електронну пошту можна надсилати дані, повертати дані користувачеві. Функції додавання, видалення, зміни елементів всередині вашої бази даних через RНР. Доступ до змінних файлів cookie та встановлення файлів cookie. Використовуючи RНР, ви можете обмежити доступ користувачів до деяких сторінок вашого веб-сайту. Також RНР дозволяє шифрувати дані. Широкий набір функцій зумовлює високу популярність RНР серед розробників клієнт-серверних застосунків, особливо коли йдеться про веб-застосунки.

Зручність у використанні та досить проста побудова коду, дає можливість створювати широкий спектр різноманітних серверних додатків для будь-якого використання.

### **3.4. Мова програмування Java**

Java — мова програмування, розроблена Джеймсом Гослінгом разом з іншими членами команди, які також були названі Зеленою командою в 1995 році для Sun Microsystems, а саме для цифрових пристроїв, таких як телеприймачі, телевізори тощо. Java безкоштовна для доступу та може працювати на всіх платформах. Java — класова та об'єктно-орієнтована мова програмування. Java полегшила життя, видаливши всі складності, такі як покажчики, перевантаження оператора, які ви бачите на C ++ або будь-якій іншій мові програмування.

Java код перетворюється в байт-код після компіляції, цей код не може бути прочитаний людиною, що забезпечує відокремлення розробника від рівня компіляції. Також java не використовує явного вказівника і запускає програми всередині так званої пісочної скриньки (пісочниці), щоб запобігти будь-якій діяльності з ненадійних джерел. Це дає змогу розробляти системи та програми, що не містять вірусів.

Java надає функцію, яка допомагає створювати розподілені програми. Використовуючи віддалений виклик методу (RMI), програма може викликати метод

іншої програми в мережі та отримати вихід. Ви можете отримати доступ до файлів, викликавши методи з будь-якої машини в Інтернеті. Java має потужну систему управління пам'яттю. Це допомагає усунути помилки, оскільки код перевіряється під час компіляції та виконання.

Є три основні компоненти Java — JVM, JDK та JRE. JDK або Java Development Kit — це те, де розробники записують свій код і запускають його через JRE або Java Runtime Environment. Як перекладається код? Через віртуальну машину Java (JVM). За допомогою JVM будь-який код, написаний на Java (або будь-якою іншою мовою), може бути переведений на байт-код Java. Тоді будь-яка машина може реалізувати цей код на основі Операційної системи. JVM знаходиться всередині JRE разом з пакетами java (бібліотеками).

Java підтримує багатопотоковість, що забезпечує одночасне виконання багатьох процесів і пришвидшує програмні продукти, які створює розробник. Java вже давно вважається однією з найкращих мов для мультиплатформерних додатків, адже вона не прив'язана до певних характеристик вашої операційної системи. У зв'язку з цим Java часто використовується як основна мова для програмування програмних продуктів на базі операційної системи Android. Також часто її використовують в комплексі з новітньою мовою Kotlin.

## 4. Опис програмної реалізації

### 4.1. Фізична модель даних

Враховуючи те, що було обрано реляційну систему керування базами даних, фізична модель (рисунок 4.1.1) побудована за принципом реляційної моделі даних.

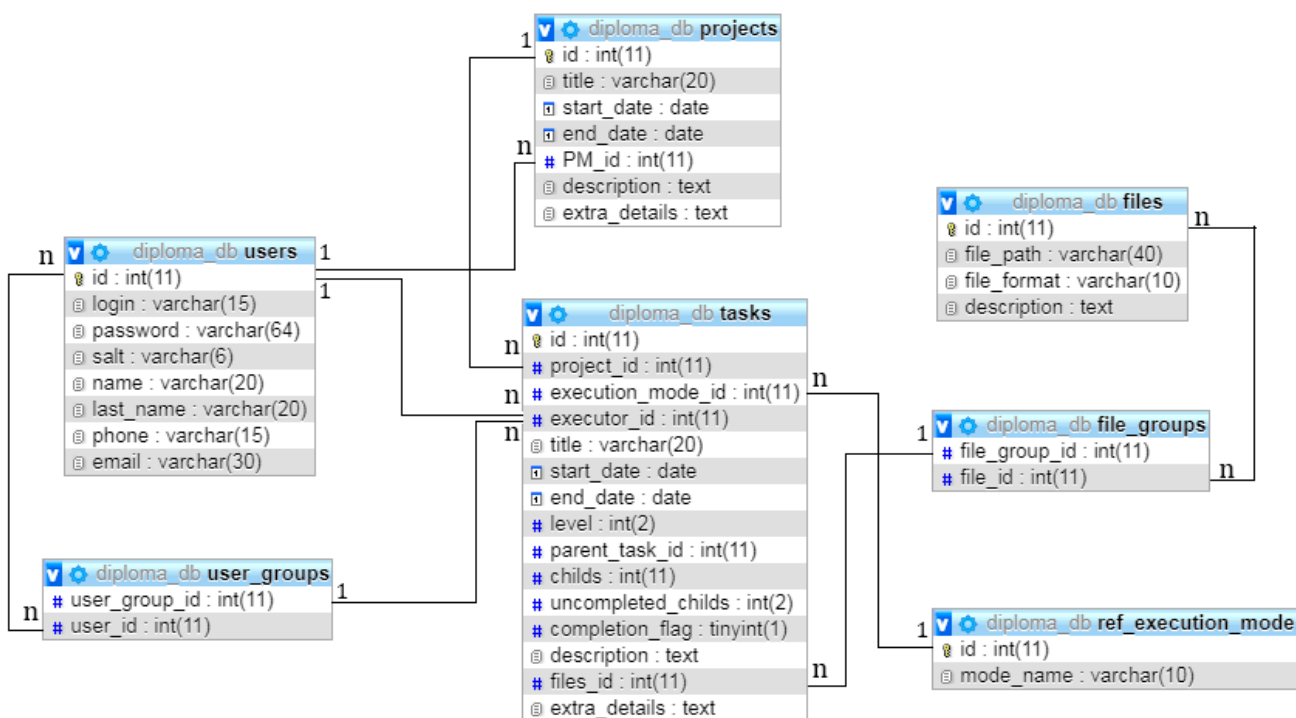


Рисунок 4.1.1 — Фізична модель даних

На рисунку показано розподіл даних програмного продукту в таблицях. Для дотримання цілісності бази даних проведена нормалізація.

### 4.2. Структура таблиць бази даних

База даних складається з семи таблиць.

В таблиці “Users” (рисунок 4.2.1) зберігається інформація про користувача. В таблиці містяться наступні поля:

- `id` — `id` користувача,
- `login` — логін користувача,
- `password` — пароль користувача в зашифрованому вигляді,
- `salt` — додаткова комбінація випадкових символів, яка використовується під час шифрування паролю,
- `name` — ім’я користувача
- `last_name` — прізвище користувача,
- `phone` — телефонний номер користувача,
- `email` — адреса електронної пошти користувача.

#	Ім'я	Тип
1	<code>id</code> 	<code>int(11)</code>
2	<code>login</code>	<code>varchar(15)</code>
3	<code>password</code>	<code>varchar(64)</code>
4	<code>salt</code>	<code>varchar(6)</code>
5	<code>name</code>	<code>varchar(20)</code>
6	<code>last_name</code>	<code>varchar(20)</code>
7	<code>phone</code>	<code>varchar(15)</code>
8	<code>email</code>	<code>varchar(30)</code>

Рисунок 4.2.1 — Структура таблиці Users

В таблиці “Tasks” (рисунок 4.2.2) зберігається інформація про завдання проектів. В таблиці містяться наступні поля:

- `id` — `id` завдання,
- `project_id` — `id` проекту з яким пов’язано це завдання,
- `execution_mode_id` — `id` варіанту виконання завдання,
- `executor_id` — `id` виконавця,
- `title` — назва завдання,
- `start_date` — дата початку завдання,

- `end_date` — дата завершення завдання,
- `level` — рівень завдання,
- `parent_task_id` — `id` батьківського завдання (якщо завдання першого рівня, то поле може приймати `null`),
- `childs` — кількість підзавдань,
- `uncompleted_childs` — кількість ще не завершених підзавдань,
- `description` — опис завдання,
- `files_id` — `id` групи файлів пов'язаних з цим завданням,
- `extra_details` — додаткові деталі завдання.


#	Ім'я	Тип
1	<b>id</b> 	int(11)
2	<b>project_id</b>	int(11)
3	<b>execution_mode_id</b>	int(11)
4	<b>executor_id</b>	int(11)
5	<b>title</b>	varchar(20)
6	<b>start_date</b>	date
7	<b>end_date</b>	date
8	<b>level</b>	int(2)
9	<b>parent_task_id</b>	int(11)
10	<b>childs</b>	int(11)
11	<b>uncompleted_childs</b>	int(2)
12	<b>completion_flag</b>	tinyint(1)
13	<b>description</b>	text
14	<b>files_id</b>	int(11)
15	<b>extra_details</b>	text

Рисунок 4.2.2 — Структура таблиці Tasks

В таблиці “Projects” (рисунок 4.2.3) зберігається інформація про проект. В таблиці містяться наступні поля:

- `id` — `id` проекту,
- `title` — назва проекту,
- `start_date` — дата початку проекту,

- end\_date — дата завершення проекту,
- PM\_id — id менеджера проекту,
- description — опис проекту,
- extra\_details — додаткові деталі проекту.

#	Ім'я	Тип
1	id 	int(11)
2	title	varchar(20)
3	start_date	date
4	end_date	date
5	PM_id	int(11)
6	description	text
7	extra_details	text

Рисунок 4.2.3 — Структура таблиці Projects

В таблиці “Files” (рисунок 4.2.4) зберігається інформація про файл. В таблиці містяться наступні поля:

- id — id файлу,
- file\_path — шлях до файлу,
- file\_format — формат файлу,
- description — опис файлу.


#	Ім'я	Тип
1	id 	int(11)
2	file_path	varchar(40)
3	file_format	varchar(10)
4	description	text

Рисунок 4.2.4 — Структура таблиці Files

В таблиці “File\_groups” (рисунок 4.2.5) зберігається інформація про групи файлів проекту. В таблиці містяться наступні поля:

- file\_group\_id — id групи файлів,
- file\_id — id файлу.

#	Ім'я	Тип
1	file_group_id	int(11)
2	file_id	int(11)

Рисунок 4.2.5 — Структура таблиці File\_groups

В таблиці “User\_groups” (рисунок 4.2.6) зберігається інформація про групи користувачів у групових завданнях проекту. В таблиці містяться наступні поля:

- user\_group\_id — id групи користувачів,
- user\_id — id користувача.

#	Ім'я	Тип
1	user_group_id	int(11)
2	user_id	int(11)

Рисунок 4.2.6 — Структура таблиці User\_groups

В таблиці “Ref\_execution\_mode” (рисунок 4.2.7) зберігається інформація про варіанти виконання завдань проекту. В таблиці містяться наступні поля:

- id — id варіанту виконання,
- mode\_name — назва варіанту виконання.


#	Ім'я	Тип
1	id 	int(11)
2	mode_name	varchar(10)

Рисунок 4.2.7 — Структура таблиці Ref\_execution\_mode

Між усіма таблицями встановлено всі необхідні зв'язки, таблиці проіндексовано, щоб забезпечити ефективну і зручну взаємодію між сервером і базою даних, а також полегшити пошук інформації в основних таблицях і зв'язаних з ними таблицях. Таблиці дозволяють реалізувати весь функціонал програмного продукту, а спосіб побудови бази даних надає можливість до розширення системи.

### 4.3. Класи і методи програмного продукту

Програмний продукт складається з шістнадцяти класів (рисунок 4.3.1). Розглянемо основний функціонал та призначення кожного з них.

- AddTaskActivity
- ContextClass
- DeleteDialogFragment
- DeleteDialogUkrFragment
- HomeActivity
- MainActivity
- MultiSpinner
- my\_context\_class
- MyHolder
- ProfileActivity
- ProjectInfoActivity
- ProjectsActivity
- SelectableItemHolderNoChilds
- SelectableItemHolderWithChilds
- TaskInfoActivity
- TasksActivity

Рисунок 4.3.1 — Список класів програмного продукту

Більшість класів мають забезпечують з'єднання з сервером для роботи з інформацією з бази даних, задля цього використано бібліотеку `android.os.AsyncTask`, яка забезпечує довготривалі запити до серверу, а також очікування відповіді від нього. Основною причиною використання такої методики є затримка при взаємодії з серверною частиною додатку.

Клас `MainActivity` призначений для автентифікації. Цей клас взаємодіє з сервером для підтвердження особи користувача і подальшої авторизації. Методи класу `MainActivity` показано на рисунку 4.3.2. За допомогою внутрішнього класу `MyAsyncTask` відбувається процедура автентифікації.

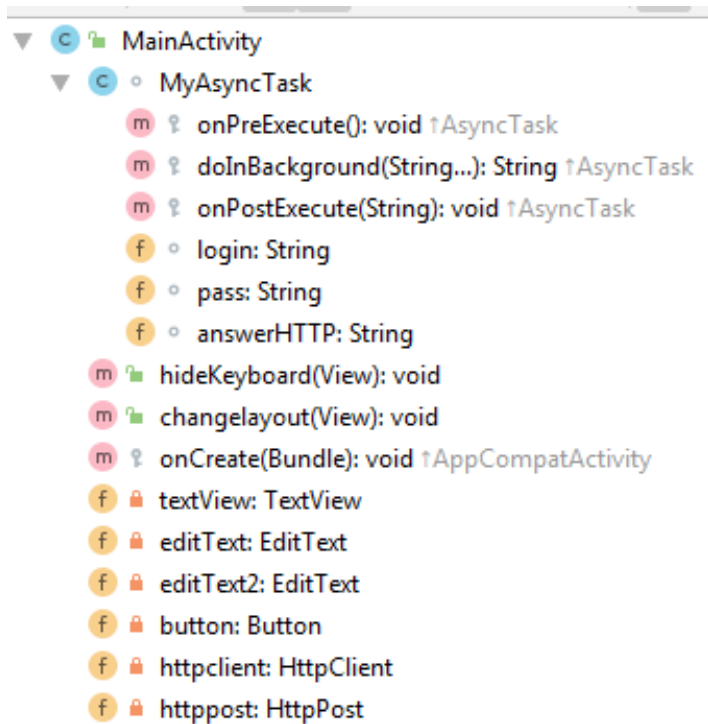


Рисунок 4.3.2 — Методи класу MainActivity

Клас ContextClass призначений для передачі інформації між класами програми. Цей клас зберігає та передає рядок з'єднання з сервером, об'єкт клієнту з'єднання, а також містить інформацію про поточну мову додатку. Методи класу ContextClass показано на рисунку 4.3.3.

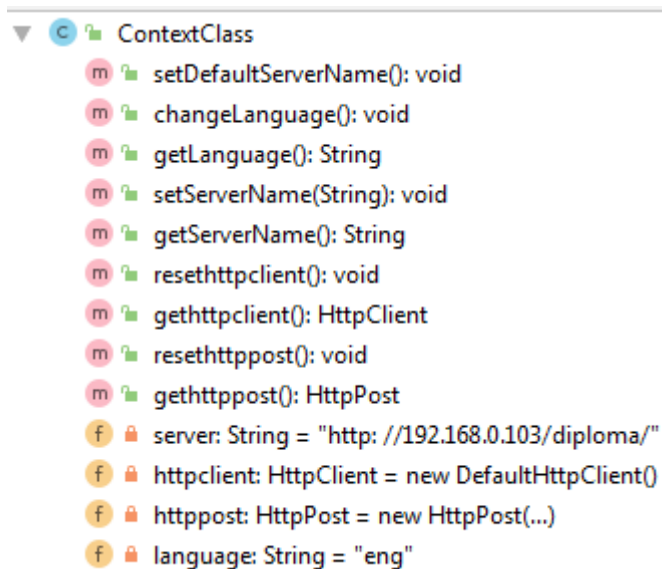


Рисунок 4.3.3 — Методи класу ContextClass

Необхідність створення такого класу пов'язана з проблемою використання глобальних змінних в Android-додатках. Таке рішення є найбільш оптимальним з боку використання оперативної пам'яті пристрою.

Клас HomeActivity необхідний для створення головного меню програми. Цей клас дозволяє користувачу перейти до одного з основних розділів програмного додатку. Методи класу HomeActivity показано на рисунку 4.3.4.

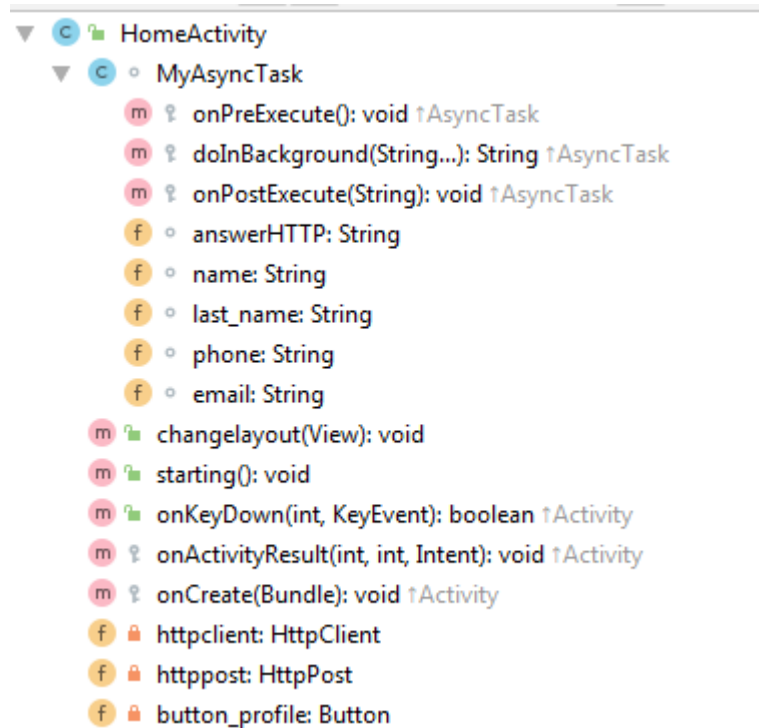


Рисунок 4.3.4 — Методи класу HomeActivity

Клас ProfileActivity призначений для перегляду особистої інформації користувача. Цей клас з'єднується з сервером і виводить на екран користувача його особисту інформацію, яка зберігається в базі даних. Методи класу ProfileActivity показано на рисунку 4.3.5.

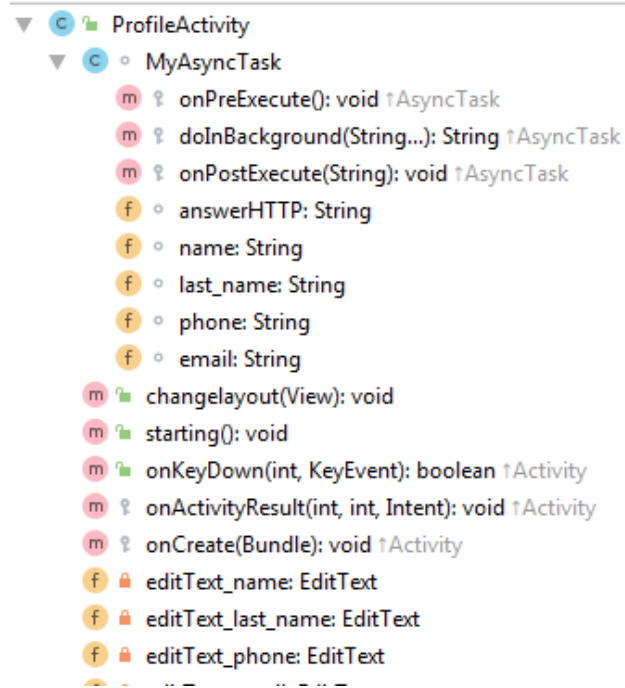


Рисунок 4.3.5 — Методи класу ProfileActivity

Клас ProjectsActivity призначений для перегляду списку проектів, які доступні користувачу. Цей клас з'єднується з сервером і виводить на екран користувача всі доступні для нього проекти у вигляді списку. Методи класу ProjectsActivity показано на рисунку 4.3.6.

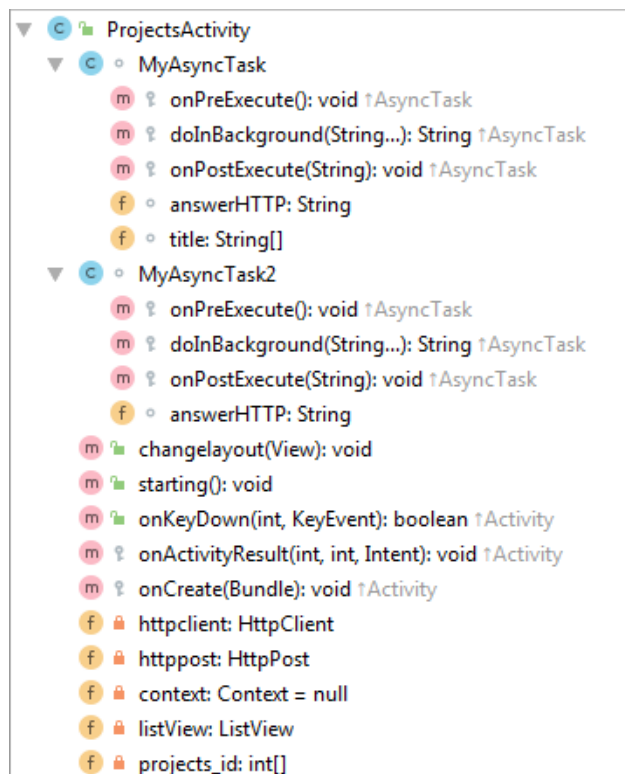


Рисунок 4.3.6 — Методи класу ProjectsActivity

Клас `ProjectInfoActivity` призначений для перегляду інформації про проект. Цей клас з'єднується з сервером і виводить на екран користувача інформацію про обраний проект. Методи класу `ProjectInfoActivity` показано на рисунку 4.3.7.

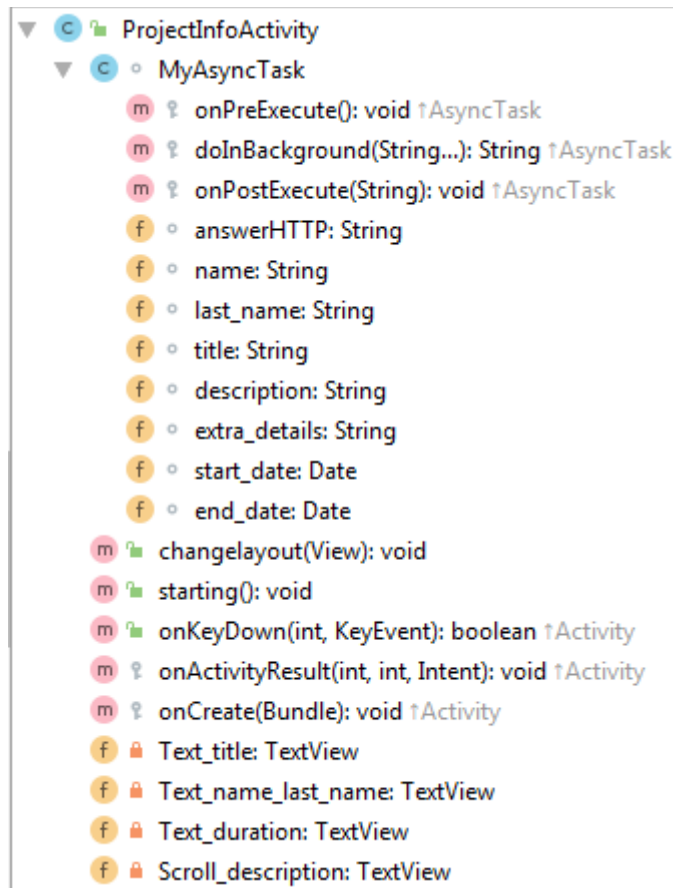


Рисунок 4.3.7 — Методи класу `ProjectInfoActivity`

Клас `TaskActivity` призначений для перегляду завдань проекту. Цей клас з'єднується з сервером і виводить на екран користувача про доступні для нього проектні завдання у вигляді багаторівневого дерева. Методи класу `TaskActivity` показано на рисунку 4.3.8.

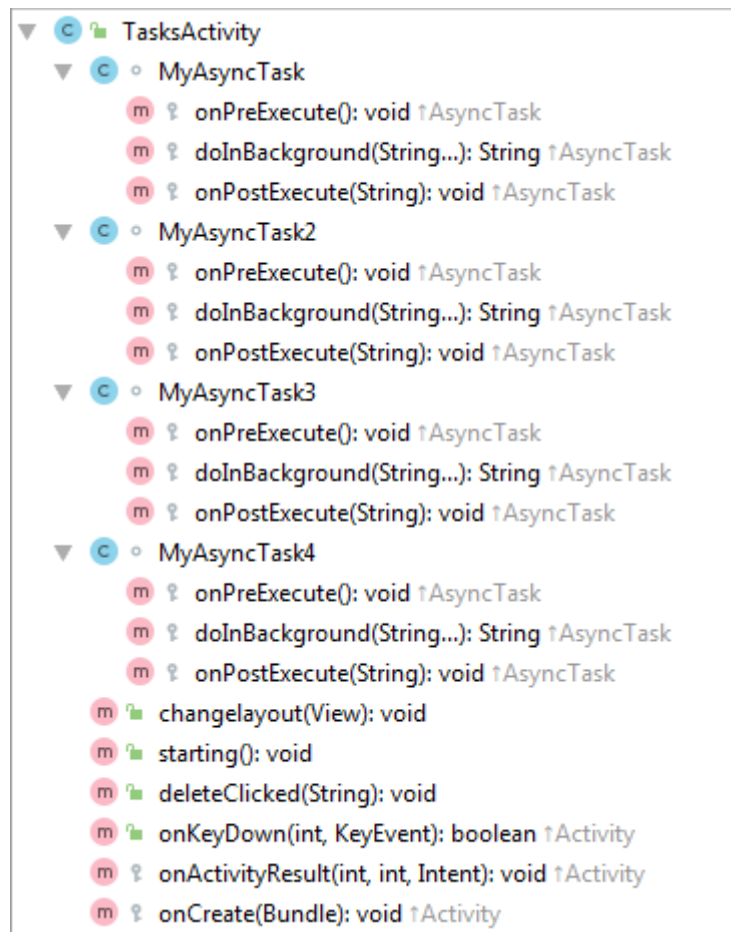


Рисунок 4.3.8 — Методи класу TaskActivity

Класи MyHolder, SelectableItemHolderNoChilds, SelectableItemHolderWithChilds є допоміжними класами, призначеними для побудови дерева завдань. Методи класу MyHolder показано на рисунку 4.3.9, SelectableItemHolderNoChilds — на рисунку 4.3.10, SelectableItemHolderWithChilds — на рисунку 4.3.11.

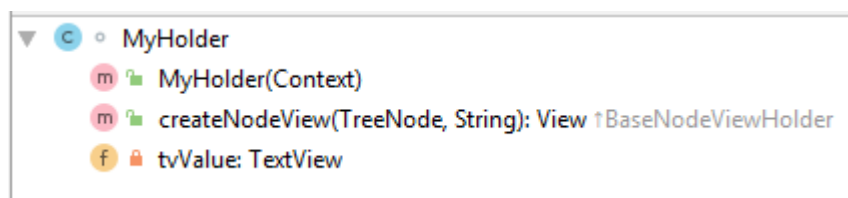


Рисунок 4.3.9 — Методи класу MyHolder

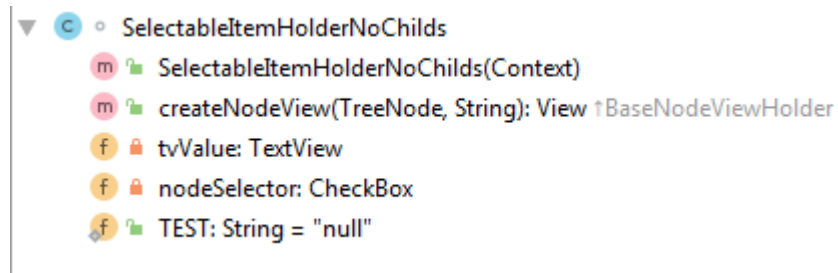


Рисунок 4.3.10 — Методи класу SelectableItemHolderNoChilds

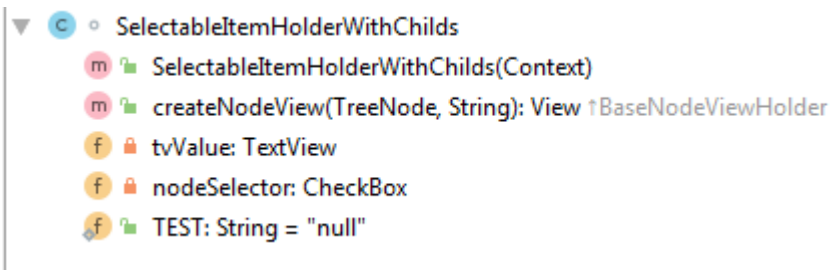


Рисунок 4.3.11 — Методи класу SelectableItemHolderWithChilds

Клас `my_context_class` призначений для збереження контексту в дереві завдань. Методи класу `my_context_class` показано на рисунку 4.3.12.

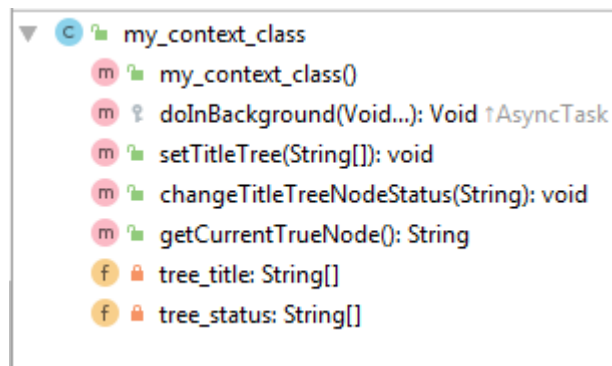


Рисунок 4.3.12 — Методи класу my\_context\_class

Клас `TaskInfoActivity` призначений для перегляду інформації про завдання. Цей клас з'єднується з сервером і виводить на екран користувача всю інформацію про вибране завдання, також клас реалізує зміну статусу завдання із не виконаного на виконане і навпаки. Методи класу `TaskInfoActivity` показано на рисунку 4.3.13.

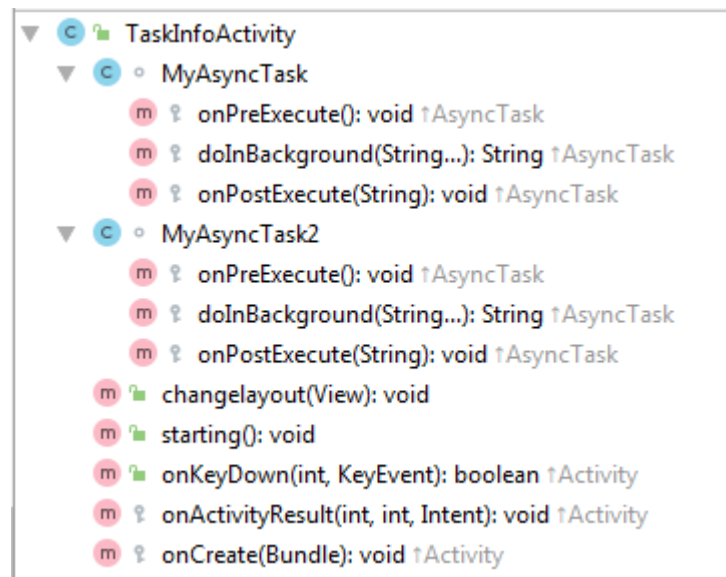


Рисунок 4.3.13 — Методи класу TaskInfoActivity

Класи DeleteDialogFragment і DeleteDialogUkrFragment це допоміжні класи-фрагменти, які призначені для видалення завдання, кожен клас працює з певною мовою додатку. Методи класу DeleteDialogFragment показано на рисунку 4.3.14, DeleteDialogUkrFragment — на рисунку 4.3.15

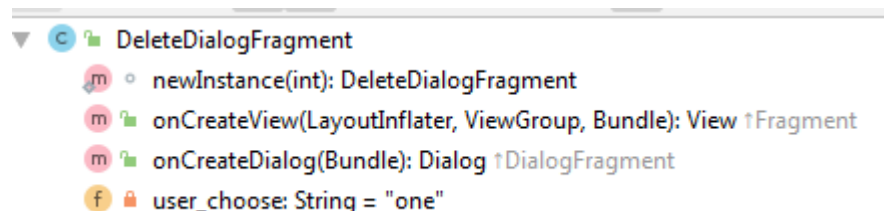


Рисунок 4.3.14 — Методи класу DeleteDialogFragment

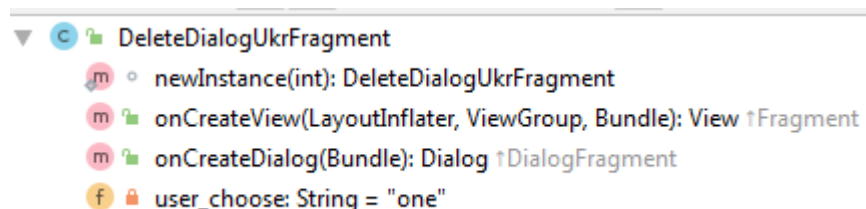


Рисунок 4.3.15 — Методи класу DeleteDialogUkrFragment

Клас `AddTaskActivity` призначений для створення нового завдання. Цей клас зчитує введені користувачем дані, з'єднується з сервером і додає нове завдання з цією інформацією до бази даних. Методи класу `AddTaskActivity` показано на рисунку 4.3.16.

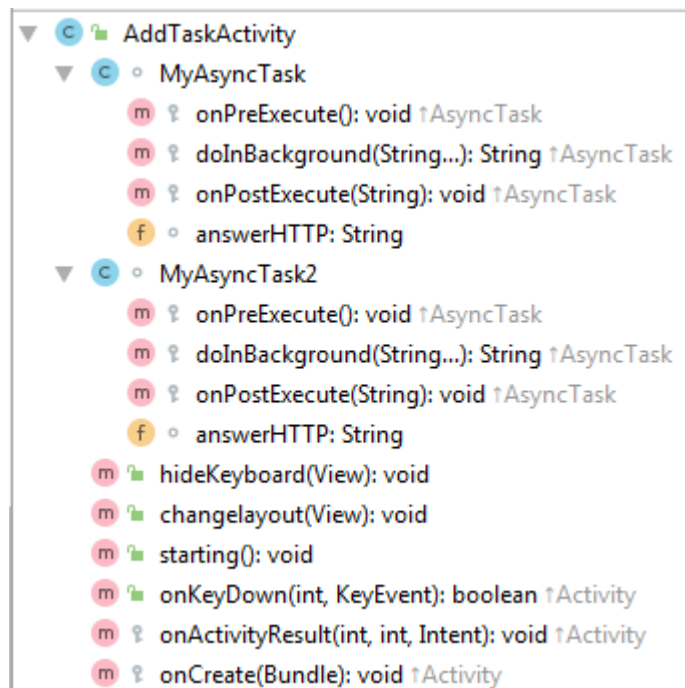


Рисунок 4.3.16 — Методи класу `AddTaskActivity`

Клас `multi_spinner` є дочірнім класом стандартного класу `Spinner`. Цей клас призначений для створення випадаючого списку із можливістю вибору декількох варіантів. Клас використовується під час створення нового завдання. Методи класу `multi_spinner` показано на рисунку 4.3.17.

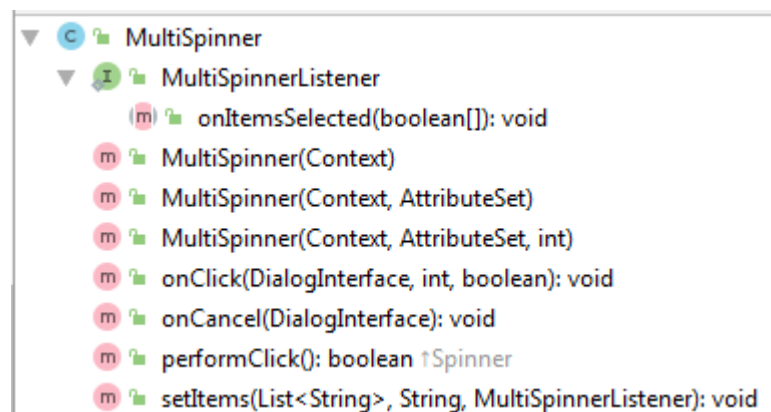


Рисунок 4.3.17 — Методи класу `multi_spinner`

#### 4.4. Функціональна модель програмного забезпечення

На рисунку (рисунок 4.4.1) зображено UML діаграму сценаріїв використання. На діаграмі показано усі варіанти взаємодії користувачів з програмним продуктом.

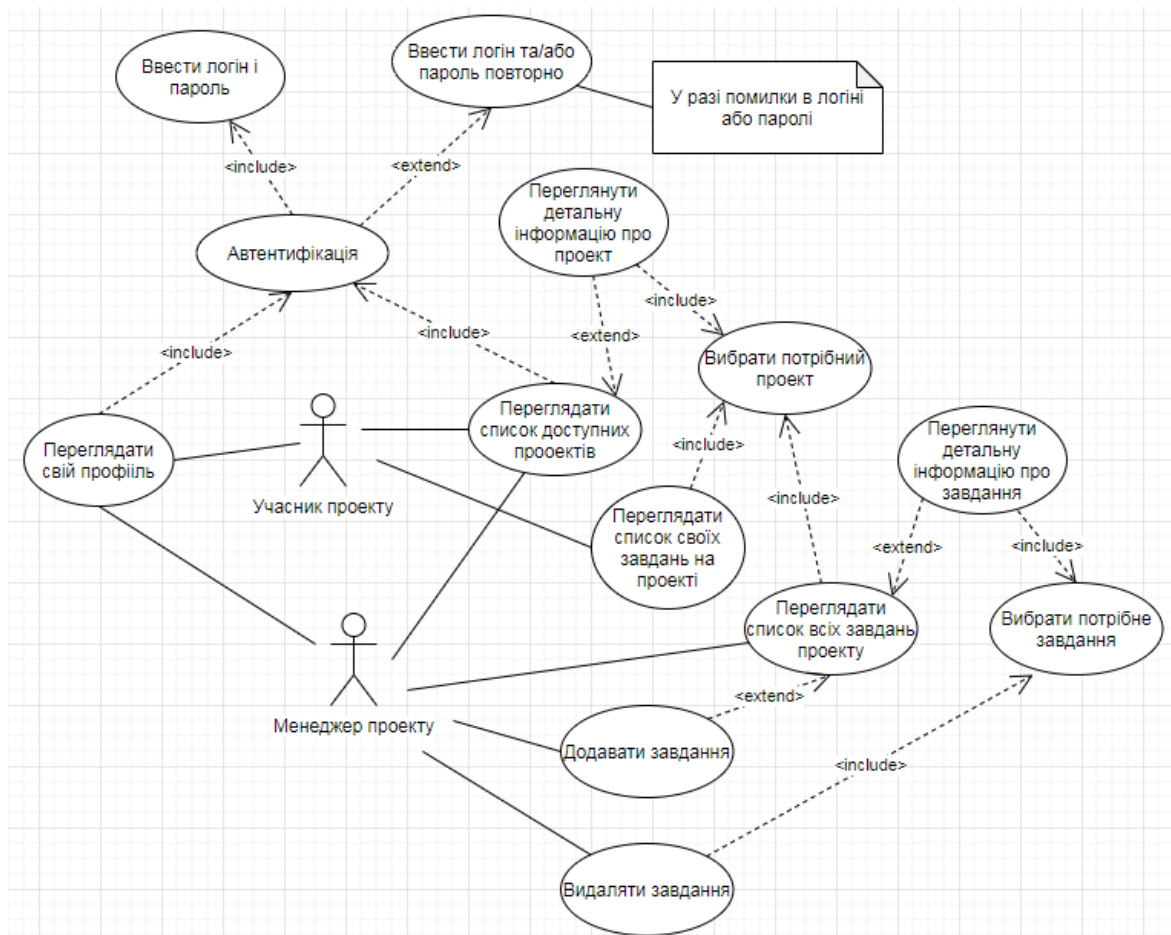


Рисунок 4.4.1 — Діаграма сценаріїв використання

В системі присутні два типи користувачів: учасник проекту і менеджер проекту. Менеджер проекту окрім усіх функцій учасника, також має змогу переглянути всі завдання проекту, додавати нові завдання, видаляти завдання. Визначення рівня доступу користувача відбувається після проходження процедури автентифікації, тоді система визначає проекти у яких задіяний цей користувач і його права на цих проектах.

## 5. Робота користувача з програмною системою

Для забезпечення стабільної роботи додатку групового управління проектами варто дотримуватися основних правил під час інсталяції і використанні.

### 5.1. Системні вимоги та інсталяція

Для встановлення розробленої програмної системи пристрій користувача повинен мати процесор не гірше ніж Qualcomm Snapdragon 425 або ж MediaTek MT6750 з тактовою частотою не менше 1,3 GHz та відеооядром не гірше ніж ARM Mali-T860 MP2, діагональ екрану не менше ніж 5.4 дюйми, обсяг оперативної пам'яті не менше ніж 2 Gb. На пристрої повинна бути встановлена операційна система Android версією не нижче ніж Jelly Bean 4.3.

Для інсталяції додатку необхідно надати дозвіл на встановлення додатків із невідомих джерел (рисунок 5.1.1).

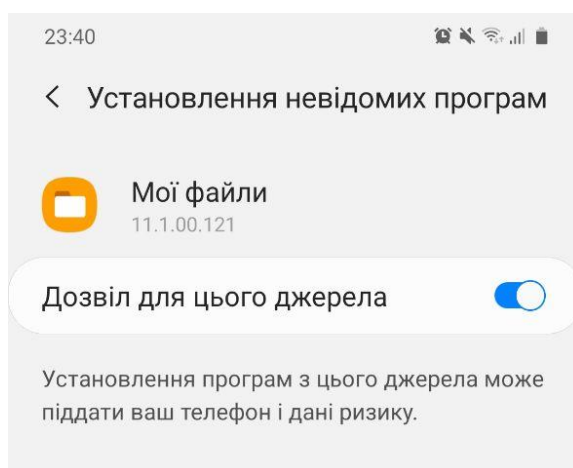


Рисунок 5.1.1 — Надання дозволу невідомим джерелам

Після цього обираємо файл інсталяції Project.apk і натиснути кнопку “Установити” (рисунок 5.1.2).

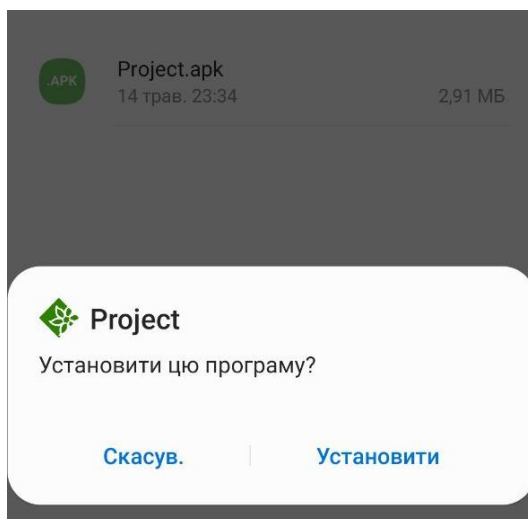


Рисунок 5.1.2 — Початок інсталяції

## 5.2. Робота користувача з програмним продуктом

Після того як додаток встановиться, з'явиться перший екран програми з формою автентифікації користувача (рисунок 5.2.1).

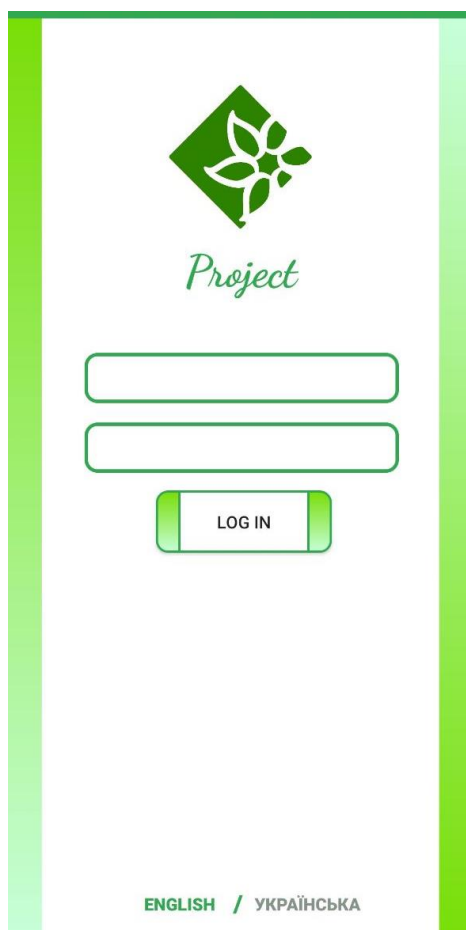


Рисунок 5.2.1 — Форма автентифікації

На цьому і всіх подальших екранах кнопки “English” та “Українська” змінюють після натискання мову на додатку з англійської на українську і навпаки (рисунок 5.1.4).

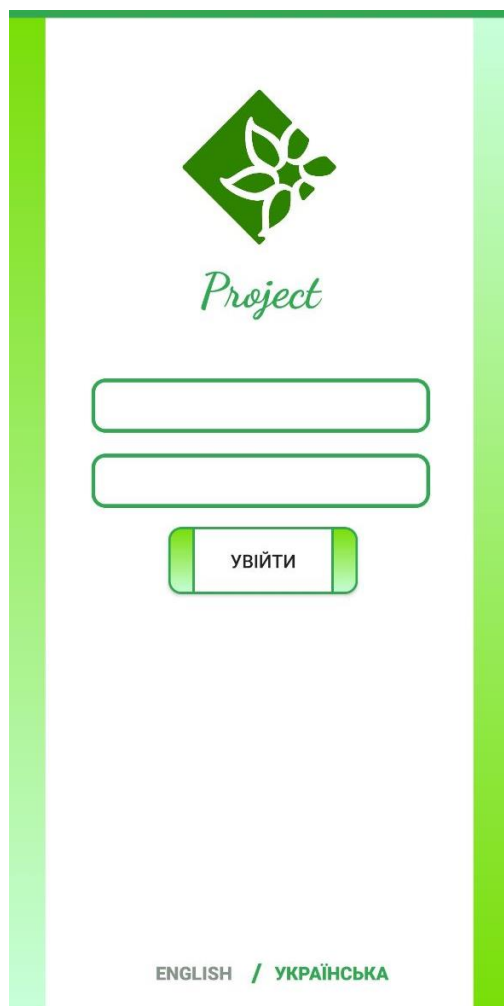


Рисунок 5.2.2 — Зміна мови додатку

Введемо логін і пароль користувача і натиснемо кнопку “Увійти”. У випадку помилки в логіні або паролі ми отримаємо відповідне повідомлення на екрані (рисунок 5.2.3).

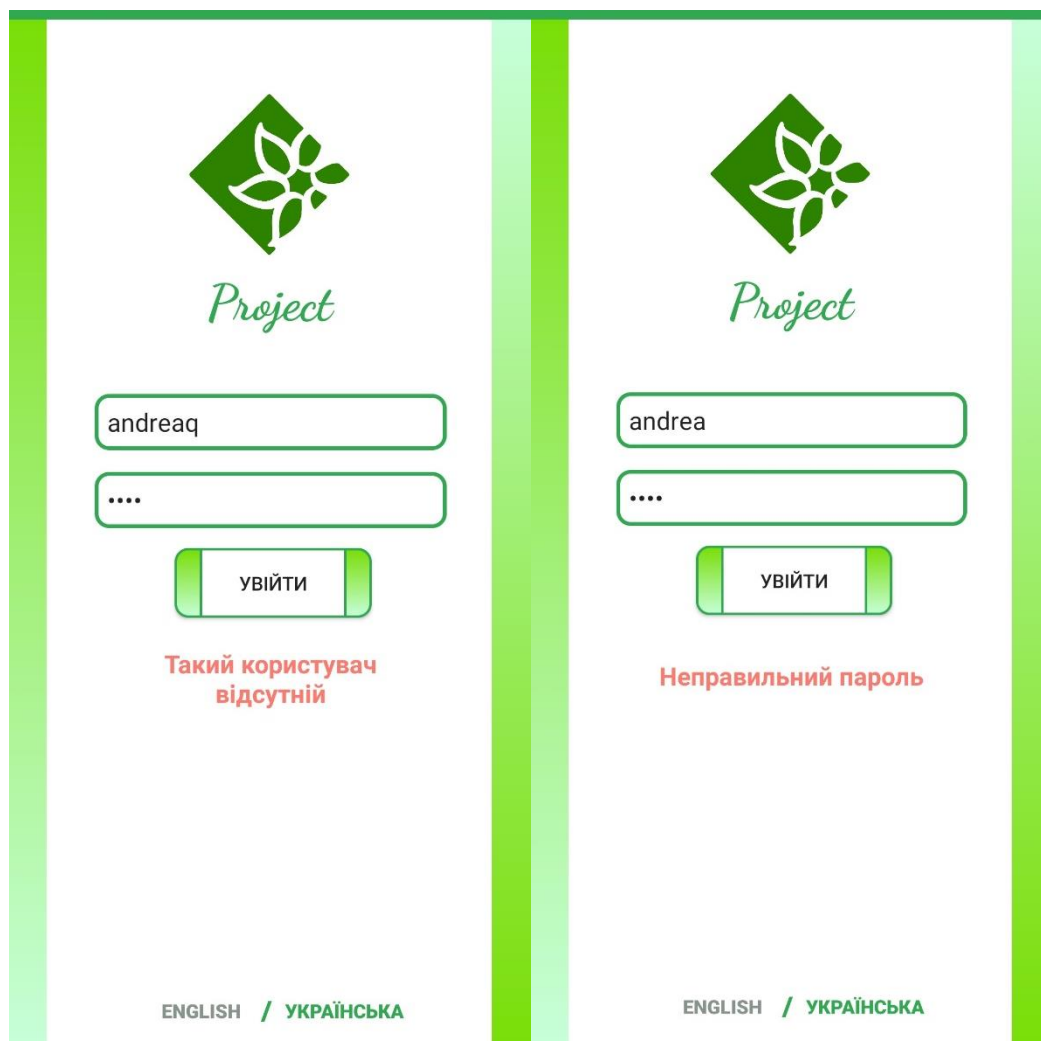


Рисунок 5.2.3 — Повідомлення з помилкою в логіні або паролі

Якщо логін і пароль введено правильно і дані користувача однакові із даними в базі системи, то він отримає повідомлення що все пройшло успішно і перейде на наступний екран (рисунок 5.2.4).

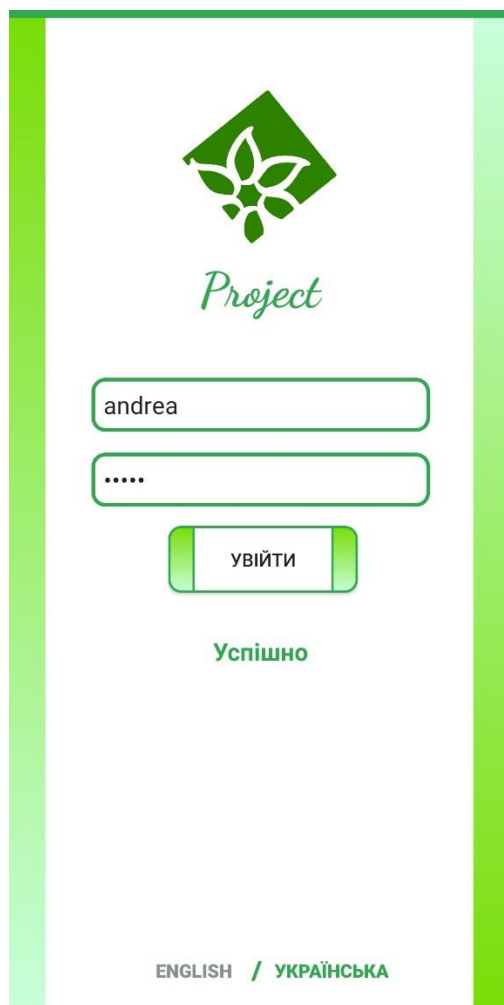


Рисунок 5.2.4 — Повідомлення про успішну автентифікацію

На наступному екрані відображено меню програми (рисунок 5.2.5) у якому при натисканні кнопки “Мій профіль” користувач перейде до розділу “Особистий кабінет” (рисунок 5.2.6), а при натисканні кнопки “Мої проекти” перейде до розділу “Мої проекти” зі списком доступних для нього проектів (рисунок 5.2.7).



Рисунок 5.2.5 — Меню програми

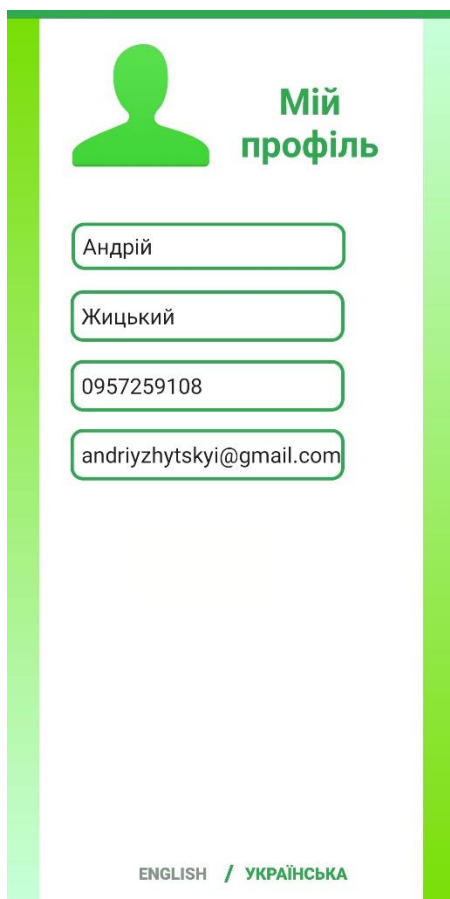


Рисунок 5.2.6 — Особистий кабінет користувача

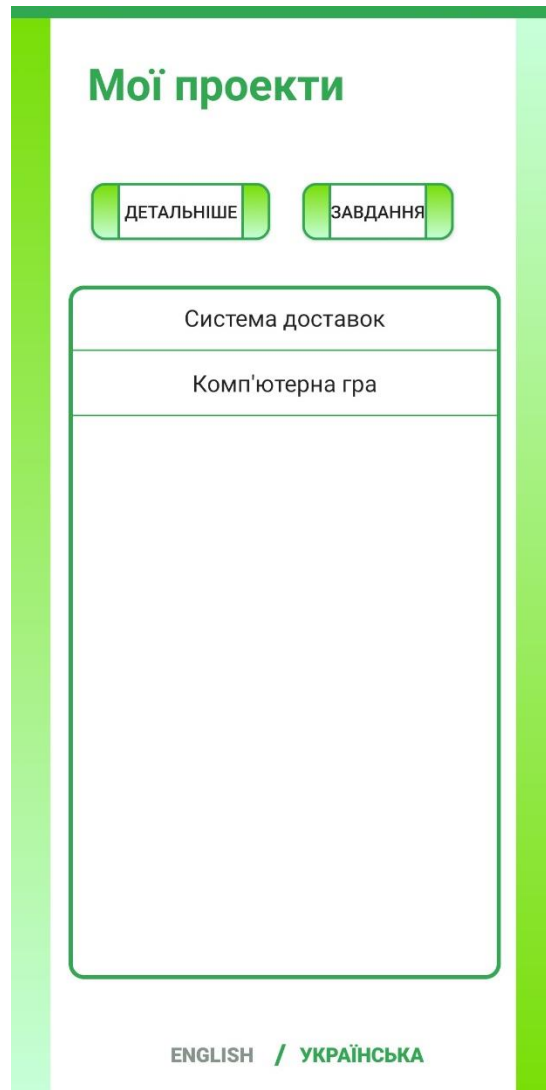


Рисунок 5.2.7 — Список проектів

У розділі “Мої проекти” оберемо один з проектів клацнувши на нього (рисунок 5.2.8).

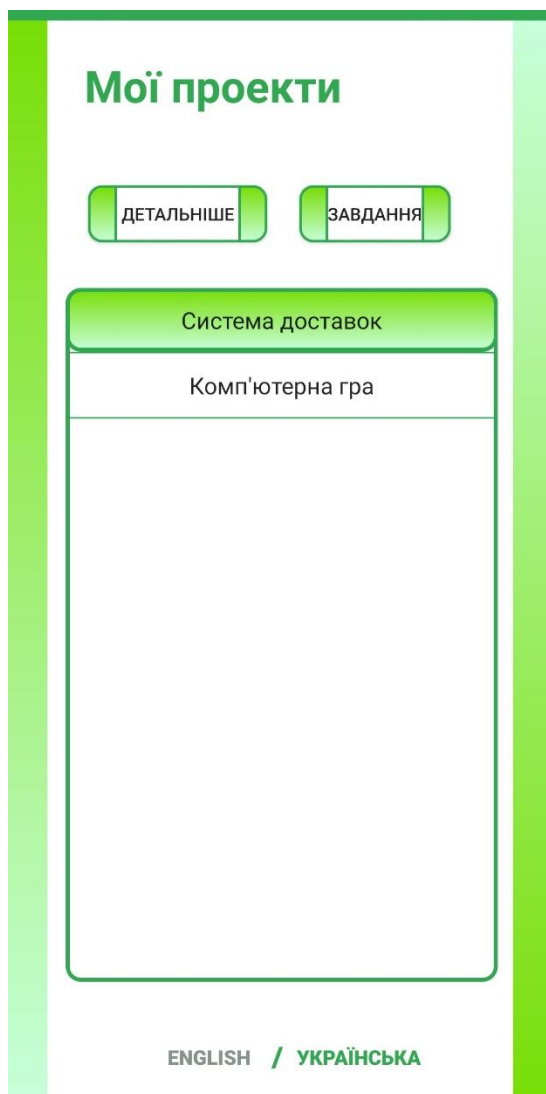


Рисунок 5.2.8 — Вибір проекту

Натискання кнопки “Детальніше” відобразить інформацію про вибраний проект (рисунок 5.2.9), а натискання кнопки “Завдання” виведе список всіх задач обраного проекту, які доступні користувачу (рисунок 5.2.10).

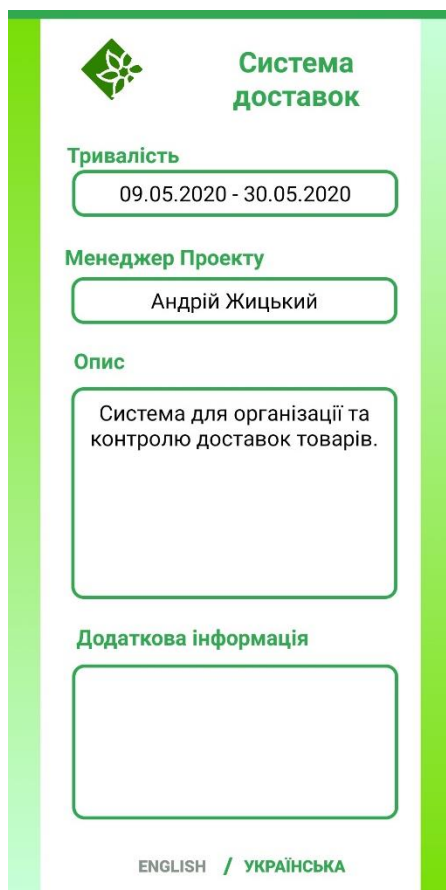


Рисунок 5.2.9 — Інформація про проект



Рисунок 5.2.10 — Інформація про завдання проекту

Далі при натисканні на назву проекту можна розгортати багаторівневий список завдань (рисунок 5.2.11). Також при натисканні на назву завдання можна обрати його (рисунок 5.2.12) для перегляду детальної інформації або для видалення з проекту (якщо роль користувача дозволяє це зробити). Натискання кнопки “Детальніше” відобразить інформацію про вибране завдання. Кнопки “Додати” та “Видалити” доступні лише менеджеру даного проекту. При натисканні кнопки “Додати” користувач додає нове завдання до проекту, а при натисканні кнопки “Видалити” видаляє обране завдання.

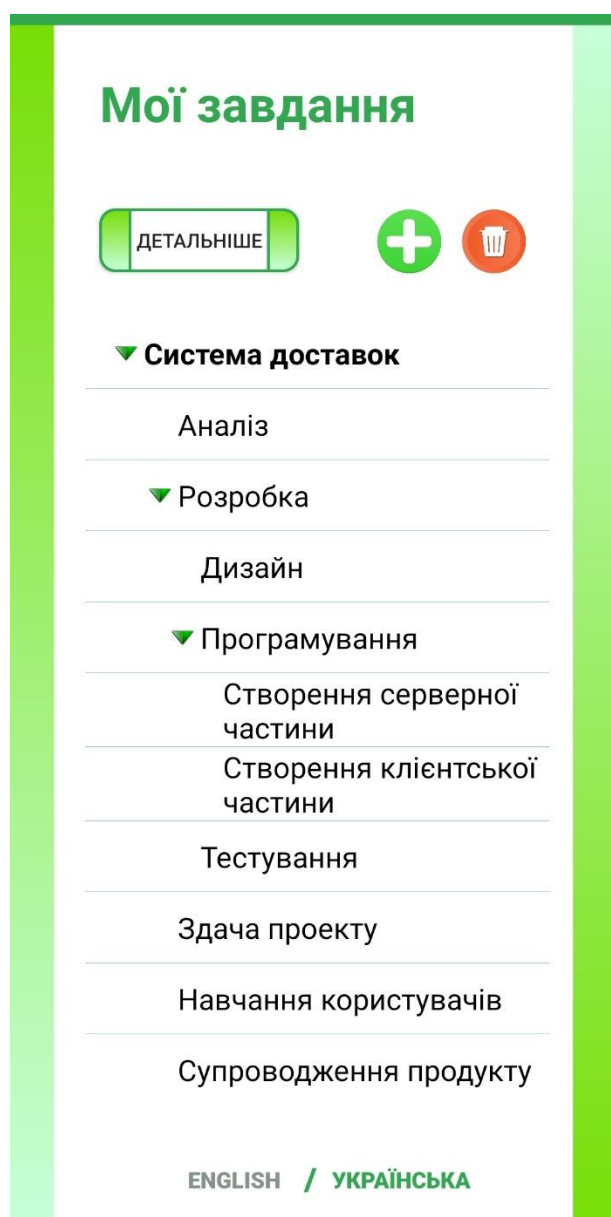


Рисунок 5.2.11 — Багаторівневий список завдань

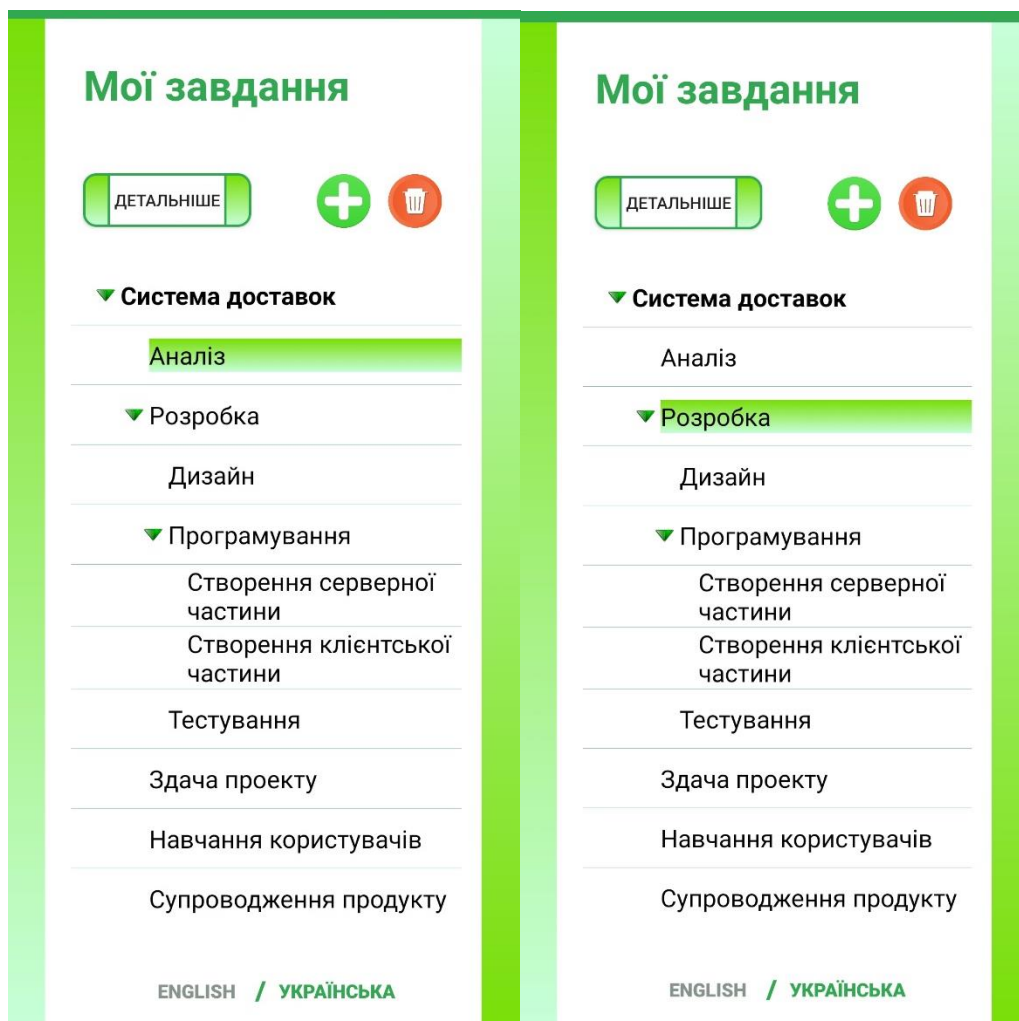


Рисунок 5.2.12 — Вибір завдання

Картинка у верхньому лівому куті екрану відображає статус завдання. Якщо завдання та всі його підзавдання виконані, то користувач побачить зелену картинку, а якщо ж саме завдання або хоча одне з його підзавдань не виконане, то система відобразить червоний варіант картинки (рисунок 5.2.13).

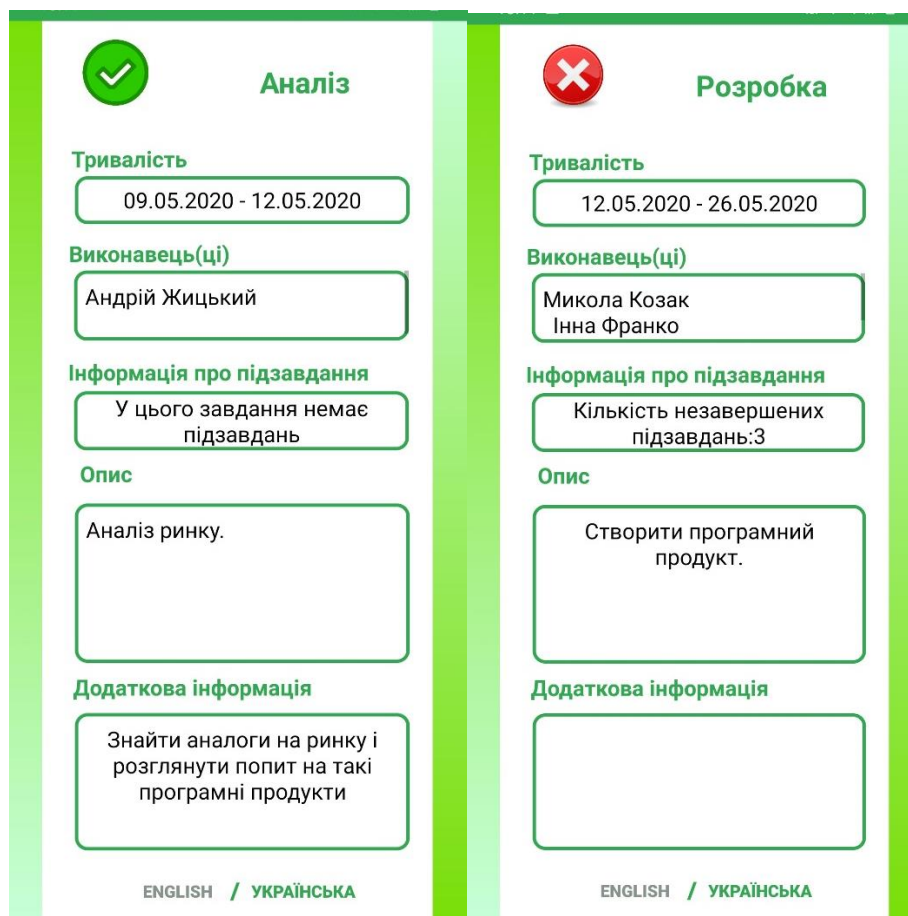


Рисунок 5.2.13 — Відображення інформації про завдання

У випадку якщо користувач є менеджером поточного проекту і він хоче видалити одне з завдань, яке при цьому має одне чи декілька підзавдань, то після натискання кнопки “Видалити” відкриється діалогове вікно з вибором варіанту видалення(рисунок 5.2.14).

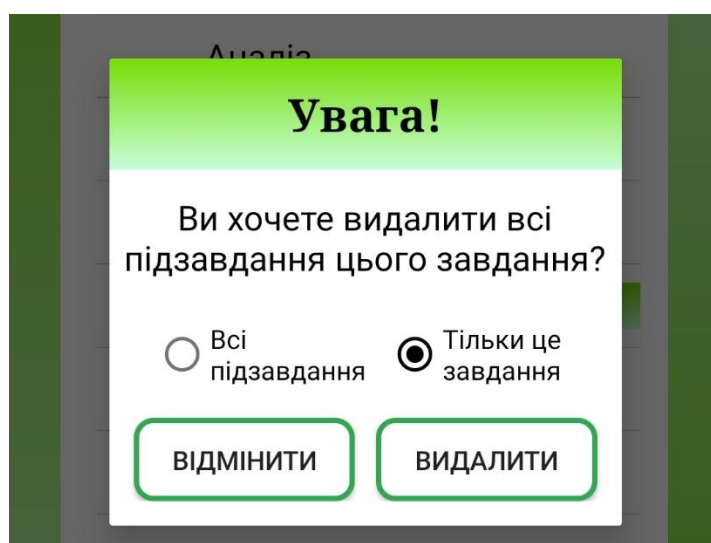
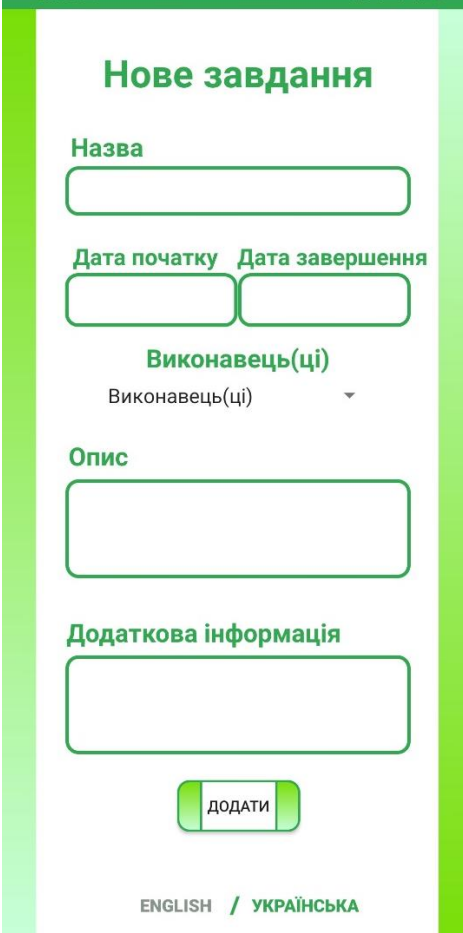


Рисунок 5.2.14 — Вибір варіанту видалення

У випадку якщо користувач є менеджером поточного проекту і він хоче додати нове завдання, то після натискання кнопки “Додати” відкриється сторінка для створення нового завдання(рисунок 5.2.15).



The screenshot shows a web form titled "Нове завдання" (New Task) with a light green background. The form contains the following fields and elements:

- Назва** (Name): A single-line text input field.
- Дата початку** (Start Date) and **Дата завершення** (End Date): Two date selection fields.
- Виконавець(ці)** (Assignee): A dropdown menu with "Виконавець(ці)" (Assignee) as the selected option.
- Опис** (Description): A multi-line text input field.
- Додаткова інформація** (Additional Information): A multi-line text input field.
- ДОДАТИ** (ADD): A green button with white text.
- At the bottom, there is a language selector: **ENGLISH / УКРАЇНСЬКА**.

Рисунок 5.2.15 — Сторінка для створення нового завдання

У разі дотримання користувачем всіх рекомендацій щодо програмного та апаратного забезпечення, програмний продукт буде стабільно і безперервно працювати. У разі недостатньої кількості оперативної пам'яті пристрою швидкість роботи додатку може бути знижена, у такому випадку користувачу варто закрити інші програми, які можуть працювати у фоні.

## Висновки

Під час розроблення програмного додатку було проаналізовано аналогічні програмні продукти та їх функції, а також виявлено основні недоліки цих продуктів.

У ході виконання роботи було створено програмний додаток групового управління проектами на базі операційної системи Android.

Сервер програми написано на мові програмування PHP, а клієнт на мові програмування Java. Під час розробки використано середовище розробки Android Studio та систему управління базами даних MySQL.

Створений програмний продукт реалізує основні функції для групового управління проектами, а також надає доступ до проектних задач різних рівнів.

У системі присутні процеси автентифікації та авторизації, паролі користувачів зберігаються у захищеному вигляді.

Користувач має змогу переглядати та змінювати свої особисті дані, переглядати всі свої поточні проекти, а також списки всіх своїх проектних завдань і статус їх виконання.

Менеджери проектів за рахунок розподілення ролей в системі, мають змогу переглядати всі завдання на тих проектах, на яких вони мають управлінські функції, а також додавати нові завдання і видаляти вже існуючі.

Додаток стабільно і швидко працює на більшості сучасних пристроїв, тож його легко і зручно використовувати. Для користування програмним продуктом не потрібно мати доступ до персонального комп'ютеру, тому він значно підвищує ефективність працівників.

## Список використаних джерел

1. Бондар, Олександр InterBase та Firebird. Практичне керівництво для розумних користувачів і починаючих розробників (+ CD-ROM) / Олександр Бондар. - М.: БХВ-Петербург, 2012. — 592 с.
2. Гонтарева, И. В. Управление проектами / И.В. Гонтарева, Р.М. Нижегородцев, Д.А. Новиков. — М.: Либроком, 2014. — 384 с
3. Ерік, Редмонд Сім баз даних за сім тижнів. Введення в сучасні бази даних та ідеологію NoSQL / Редмонд Ерік. — М.: ДМК Пресс, 2018. — 667 с.
4. Машнин, Т. С. Eclipse. Разработка RCP-, Web-, Ajax- и Android-приложений на Java / Т.С. Машнин. — М.: БХВ-Петербург, 2013. — 384 с.
5. Frame Davidson J. The New Project Management. San Francisco: Jossey-Bass Publishers, 1994. — 328 p.
6. Kerzner, H. Project Management: a systems approach to planning, scheduling, and controlling. 4th ed., New York, Van Nostrand Reinhold, 1992. — 1028 p.
7. Project Management Body of Knowledge (PM BOK), Project Management Institute, Drexel Hill, Pennsylvania, 1987. — 756 p.

# ДОДАТОК 1

Програмний додаток групового управління проектами на базі  
операційної системи Android

Специфікація

Аркушів 2

Київ 2020

Позначення	Найменування	Примітки
Документація		
Компоненти		

## ДОДАТОК 2

Програмний додаток групового управління проектами на базі  
операційної системи Android

Текст програми

Аркушів 20

Київ 2020

- 2 -

**MainActivity.java**

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import android.view.animation.RotateAnimation;
import android.view.animation.Animation;
import android.view.animation.LinearInterpolator;
import android.os.CountDownTimer;
import android.view.inputmethod.InputMethodManager;

public class MainActivity extends AppCompatActivity {

    private TextView textView;
    private EditText editText;
    private EditText editText2;
    private Button button;
```

- 3 -

```
private HttpClient httpclient;
private HttpPost httppost;

public void hideKeyboard(View v) {
    InputMethodManager inputMethodManager =
(InputMethodManager) getSystemService(INPUT_METHOD_SERVICE);
    inputMethodManager.hideSoftInputFromWindow(v.getWindowToken(),0);
}

public void changelayout(View view){
    final ContextClass current_language = (ContextClass) getApplicationContext();
    current_language.changeLanguage();
    if( current_language.getLanguage().equals("ukr")) setContentView(R.layout.activity_main_ukr);
    else setContentView(R.layout.activity_main_eng);

    textView = (TextView)findViewById(R.id.textView);
    editText = (EditText)findViewById(R.id.editText_login);
    editText2 = (EditText)findViewById(R.id.editText_pass);
    button = (Button)findViewById(R.id.button);

    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            hideKeyboard(v);
            new MyAsyncTask().execute();
        }
    });

    editText.setOnFocusChangeListener(new View.OnFocusChangeListener() {
        @Override
        public void onFocusChange(View v, boolean hasFocus) {
            if (!hasFocus) {
                hideKeyboard(v);
            }
        }
    });
};
```

- 4 -

```

editText2.setOnFocusChangeListener(new View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View v, boolean hasFocus) {
        if (!hasFocus) {
            hideKeyboard(v);
        }
    }
});
}

```

```

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

    final ContextClass serverconnect = (ContextClass) getApplicationContext();
    //httpost = serverconnect.gethttpost();
    String server = serverconnect.getServerName()+"log_in.php";
    httpost = new HttpPost(server);
    httpclient = serverconnect.gethttpclient();
    serverconnect.setDefaultServerName();

    textView = (TextView)findViewById(R.id.textView);
    editText = (EditText)findViewById(R.id.editText_login);
    editText2 = (EditText)findViewById(R.id.editText_pass);
    button = (Button)findViewById(R.id.button);

    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            hideKeyboard(v);
            new MyAsyncTask().execute();
        }
    });
}

```

- 5 -

```

editText.setOnFocusChangeListener(new View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View v, boolean hasFocus) {
        if (!hasFocus) {
            hideKeyboard(v);
        }
    }
});

```

```

editText2.setOnFocusChangeListener(new View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View v, boolean hasFocus) {
        if (!hasFocus) {
            hideKeyboard(v);
        }
    }
});

```

```

}

```

```

class MyAsyncTask extends AsyncTask<String, String, String> {

```

```

    String login, pass, answerHTTP;

```

```

    @Override
    protected void onPreExecute() {
        login = editText.getText().toString();
        pass = editText2.getText().toString();
        super.onPreExecute();
    }

```

```

    @Override
    protected String doInBackground(String... params) {

        try {
            List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);

```

- 6 -

```
nameValuePairs.add(new BasicNameValuePair("login", login));
nameValuePairs.add(new BasicNameValuePair("pass", pass));
httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs, "UTF-8"));
```

```
HttpResponse response = httpClient.execute(httppost);
```

```
if (response.getStatusLine().getStatusCode() == 200) {
    HttpEntity entity = response.getEntity();
    answerHTTP = EntityUtils.toString(entity);
}
```

```
}
catch (ClientProtocolException e) {
}
```

```
catch (IOException e) {
}
```

```
return null;
```

```
}
```

```
@Override
```

```
protected void onPostExecute(String result) {
```

```
    super.onPostExecute(result);
```

```
    final ContextClass current_language = (ContextClass) getApplicationContext();
```

```
    if(answerHTTP.equals("Sucess")) {
```

```
        textView.setTextColor(Color.parseColor("#32a852"));
```

```
        if(current_language.getLanguage().equals("eng")) textView.setText(answerHTTP);
```

```
        else textView.setText("Успішно");
```

```
        RotateAnimation rotate = new RotateAnimation(0, 180, Animation.RELATIVE_TO_SELF, 0.5f,
Animation.RELATIVE_TO_SELF, 0.5f);
```

```
        rotate.setDuration(1000);
```

```
        rotate.setInterpolator(new LinearInterpolator());
```

```
        ImageView image= (ImageView) findViewById(R.id.imageView);
```

```
        image.startAnimation(rotate);
```

```
        new CountdownTimer(1000, 1000) {
```



- 8 -

```
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import org.json.JSONException;
import org.json.JSONObject;

public class HomeActivity extends Activity {

    private HttpClient httpclient;
    private HttpPost httppost;

    private Button button_profile;
    private Button button_projects;
    private Button button_schedule;

    public void changelayout(View view){
        final ContextClass app_language = (ContextClass) getApplicationContext();
        app_language.changeLanguage();

        starting();
    }

    public void starting(){

        final ContextClass app_lang = (ContextClass) getApplicationContext();
        if (app_lang.getLanguage().equals("ukr")) setContentView(R.layout.activity_home_ukr);
        else setContentView(R.layout.activity_home_eng);

        button_profile = (Button)findViewById(R.id.goto_my_profile);
        button_projects = (Button)findViewById(R.id.goto_my_projects);
```

- 9 -

```
button_schedule = (Button)findViewById(R.id.goto_my_schedule);
```

```
button_profile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(HomeActivity.this, ProfileActivity.class);
        startActivityForResult(intent,1);
    }
});
```

```
button_projects.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(HomeActivity.this, ProjectsActivity.class);
        startActivityForResult(intent,1);
    }
});
```

```
button_schedule.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

    }
});
}
```

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK ) {

        Intent startMain = new Intent(Intent.ACTION_MAIN);
        startMain.addCategory(Intent.CATEGORY_HOME);
        startMain.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(startMain);

        return true;
    }
}
```

- 10 -

```

    }

    return super.onKeyDown(keyCode, event);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    starting();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    final ContextClass serverconnect = (ContextClass) getApplicationContext();
    String server = serverconnect.getServerName()+"my_profile.php";
    httppost = new HttpPost(server);
    httpclient = serverconnect.getHttpClient();

    starting();
}
}

```

### **ProjectsActivity.java**

```

import android.app.Activity;
import android.os.Bundle;
import android.os.AsyncTask;
import android.content.Intent;
import android.view.KeyEvent;
import android.content.Context;
import android.view.View;
import android.widget.ListView;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.Toast;
import org.apache.http.HttpEntity;

```

- 11 -

```
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import android.widget.Button;
import org.json.JSONException;
import org.json.JSONObject;

public class ProjectsActivity extends Activity {

    private HttpClient httpclient;
    private HttpPost httppost;
    private Context context = null;
    private ListView listView;
    private int[] projects_id;
    private int current_selected_project = 0;
    private Button tasks_button;
    private Button details_button;

    public void changelayout(View view){
        final ContextClass app_language = (ContextClass) getApplicationContext();
        app_language.changeLanguage();

        starting();

        new MyAsyncTask().execute();
    }

    public void starting(){
```

- 12 -

```
final ContextClass app_lang = (ContextClass) getApplicationContext();
if (app_lang.getLanguage().equals("ukr")) setContentView(R.layout.activity_projects_ukr);
else setContentView(R.layout.activity_projects_eng);

current_selected_project = 0;

tasks_button = (Button)findViewById(R.id.tasks);

tasks_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (current_selected_project != 0) {
            new MyAsyncTask2().execute();
            Intent intent = new Intent(ProjectsActivity.this, TasksActivity.class);
            startActivityForResult(intent, 1);
        }
    }
});

details_button = (Button)findViewById(R.id.details);

details_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (current_selected_project != 0) {
            new MyAsyncTask2().execute();
            Intent intent = new Intent(ProjectsActivity.this, ProjectInfoActivity.class);
            startActivityForResult(intent, 1);
        }
    }
});

listView = findViewById(R.id.listView);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
```

- 13 -

```
public void onItemClick(AdapterView<?> parent, View itemClicked, int position,
                        long id) {
    current_selected_project = projects_id[position];
}
});
}
```

@Override

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK ) {

        Intent intent = new Intent();
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        setResult(RESULT_OK, intent);
        finish();

        return true;
    }

    return super.onKeyDown(keyCode, event);
}
```

@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    starting();
    new MyAsyncTask().execute();
}
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    final ContextClass serverconnect = (ContextClass) getApplicationContext();
    String server = serverconnect.getServerName()+"my_projects.php";
    httpPost = new HttpPost(server);
    httpClient = serverconnect.getHttpClient();
}
```

- 14 -

```
starting();

context = this;

new MyAsyncTask().execute();
}

//-----
class MyAsyncTask extends AsyncTask<String, String, String> {

    String answerHTTP;

    String[] title;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected String doInBackground(String... params) {

        try {
            List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
            nameValuePairs.add(new BasicNameValuePair("projects", "get_projects"));
            httpclient.setEntity(new UrlEncodedFormEntity(nameValuePairs, "UTF-8"));

            HttpResponse response = httpclient.execute(httpclient);

            if (response.getStatusLine().getStatusCode() == 200) {
                HttpEntity entity = response.getEntity();
                answerHTTP = EntityUtils.toString(entity);
                String JSON_STRING = "{}";
                try {
                    JSONObject obj = new JSONObject(answerHTTP);
                    int i = 1;
                }
            }
        }
    }
}
```

- 15 -

```
while( !obj.isNull("project "+i) ){
    i++;
}

title = new String[i-1];

projects_id = new int[i-1];

i = 1;
while( !obj.isNull("project "+i) ){
    JSONObject user = obj.getJSONObject("project "+i);
    title[i-1] = user.getString("title");

    projects_id[i-1] = user.getInt("id");
    i++;
}

} catch (JSONException e){
    e.printStackTrace();
}
}
}
catch (ClientProtocolException e) {
}
catch (IOException e) {
}

return null;
}

@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    ArrayAdapter<String> adapter = new ArrayAdapter<>(context,
        R.layout.list_item, title);
```

- 16 -

```

        listView.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
        listView.setAdapter(adapter);
    }
}
//-----

class MyAsyncTask2 extends AsyncTask<String, String, String> {

    String answerHTTP;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected String doInBackground(String... params) {

        try {
            final ContextClass serverconnect = (ContextClass) getApplicationContext();
            String server = serverconnect.getServerName()+"set_project.php";
            HttpPost httppost = new HttpPost(server);
            HttpClient httpclient = serverconnect.getHttpClient();

            List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
            nameValuePairs.add(new BasicNameValuePair("current_project",
String.valueOf(current_selected_project)));
            httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs, "UTF-8"));

            HttpResponse response = httpclient.execute(httppost);

            if (response.getStatusLine().getStatusCode() == 200) {
                HttpEntity entity = response.getEntity();
                answerHTTP = EntityUtils.toString(entity);
            }
        }
    }
}

```

- 17 -

```
        catch (ClientProtocolException e) {
        }
        catch (IOException e) {
        }

        return null;
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
    }
}
```

### **ProfileActivity.java**

```
import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.os.AsyncTask;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import org.json.JSONException;
```

- 18 -

```
import org.json.JSONObject;

public class ProfileActivity extends Activity {

    private EditText editText_name;
    private EditText editText_last_name;
    private EditText editText_phone;
    private EditText editText_email;
    private Button change_button;

    private HttpClient httpclient;
    private HttpPost httppost;

    public void changelayout(View view){
        final ContextClass app_language = (ContextClass) getApplicationContext();
        app_language.changeLanguage();

        starting();
        new MyAsyncTask().execute();
    }

    public void starting(){
        final ContextClass app_lang = (ContextClass) getApplicationContext();
        if (app_lang.getLanguage().equals("ukr")) setContentView(R.layout.activity_profile_ukr);
        else setContentView(R.layout.activity_profile_eng);

        editText_name = (EditText)findViewById(R.id.editText_name);
        editText_last_name = (EditText)findViewById(R.id.editText_last_name);
        editText_phone = (EditText)findViewById(R.id.editText_phone);
        editText_email = (EditText)findViewById(R.id.editText_email);
        change_button = (Button)findViewById(R.id.change);
    }

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_BACK ) {
```

- 19 -

```

Intent intent = new Intent();
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
setResult(RESULT_OK, intent);
finish();
return true;
}
return super.onKeyDown(keyCode, event);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    starting();
    new MyAsyncTask().execute();
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    final ContextClass serverconnect = (ContextClass) getApplicationContext();
    String server = serverconnect.getServerName()+"my_profile.php";
    httppost = new HttpPost(server);
    httpclient = serverconnect.getHttpClient();
    starting();
    new MyAsyncTask().execute();
}
class MyAsyncTask extends AsyncTask<String, String, String> {
    String answerHTTP, name, last_name, phone, email;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
    @Override
    protected String doInBackground(String... params) {
        try {
            List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
            nameValuePairs.add(new BasicNameValuePair("profile", "get_profile"));
            httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs, "UTF-8"));

```

- 20 -

```

    HttpResponse response = httpClient.execute(httpPost);
    if (response.getStatusLine().getStatusCode() == 200) {
        HttpEntity entity = response.getEntity();
        answerHTTP = EntityUtils.toString(entity);
        String JSON_STRING = "{";
        try {
            JSONObject obj = new JSONObject(answerHTTP);
            JSONObject user = obj.getJSONObject("user");
            name = user.getString("name");
            last_name = user.getString("last_name");
            phone = user.getString("phone");
            email = user.getString("email");
        } catch (JSONException e){
            e.printStackTrace();
        }
    }
}
catch (ClientProtocolException e) {
}
catch (IOException e) {
}

return null;
}
@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    editText_name.setText(name);
    editText_last_name.setText(last_name);
    editText_phone.setText(phone);
    editText_email.setText(email);
}
}
}

```

## ДОДАТОК 3

Програмний додаток групового управління проектами на базі  
операційної системи Android

Опис програми

Аркушів 5

Київ 2020

## АНОТАЦІЯ

Розроблений програмний модуль забезпечує групове управління програмними проектами.

Програмний модуль призначений для з'єднання з сервером і отримання інформації з бази даних, виведення на екран користувача списку проектів, інформації про проект, багаторівневого дерева завдань, інформації про завдання. Модуль також зчитує введену менеджером проектів інформацію про нове завдання та вносить їх до бази даних. Мова програмної реалізації — Java. Середовище розробки — Android Studio.

## ЗМІСТ

1. Відомості про програмний модуль .....	4
1.1 Опис логічної структури .....	4
1.2 Вхідні та вихідні дані.....	4
Використані технічні засоби .....	5

# 1. ВІДОМОСТІ ПРО ПРОГРАМНИЙ МОДУЛЬ

Клієнт-додаток називається Project. Даний продукт розроблено у середовищі Android Studio, використовуючи мову програмування Java.

Програма призначена групового управління проектами.

## 1.1. Опис логічної структури

Програмний продукт було розроблено у вигляді мобільного додатку за допомогою універсального набору класів Activity і Fragment.

Мобільний додаток з'єднується з сервером за допомогою широкого набору запитів, що дозволяє йому як отримувати інформацію з бази даних, так і вносити нові дані.

Розроблений модуль подає отриману інформацію у різних варіантах графічного відображення, в залежності від поточного запиту від користувача.

## 1.2. Вхідні та вихідні дані

Вхідними даними для модулю є дані з бази даних, дані внесені користувачем з клавіатури мобільного додатку, вибори користувача.

Вихідними даними, які повертає модуль є сторінки зі списком проектів, багаторівневим деревом завдань, особистою інформацією користувача.

## **ВИКОРИСТАНІ ТЕХНІЧНІ ЗАСОБИ**

Програмний модуль було протестовано на мобільному пристрої на базі операційної системи Android, з процесором Samsung Exynos 7904 (2x1.8 ГГц + 6x1.6 ГГц) та з 3 Гб оперативної пам'яті. Розроблене програмне забезпечення не потребує значних апаратних витрат, тому може використовуватись на менш потужних пристроях, оскільки всі дії відбуваються на сервері.