

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

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Наталія АУШЕВА

«\_\_\_» \_\_\_\_\_ 2022 р.

**Дипломна робота**  
**на здобуття ступеня бакалавра**  
**спеціальності 122 «Комп'ютерні науки»**  
**освітня програма «Комп'ютерний моніторинг та геометричне**  
**моделювання процесів і систем»**  
**на тему: «Мобільний додаток для організації навчання студента»**

Виконав:

Студент IV курсу, групи ТР-81  
Дійнега Денис Олександрович \_\_\_\_\_

Керівник:

Ст. викладач,  
Колумбет Вадим Петрович \_\_\_\_\_

Рецензент:

Доцент,  
Ларіна Катерина Юріївна \_\_\_\_\_

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

Студент \_\_\_\_\_

Київ — 2022 року

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

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший

спеціальність: 122 «Комп’ютерні науки»

освітня програма «Комп’ютерний моніторинг та геометричне моделювання процесів і систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Наталія АУШЕВА

(підпис)

” \_\_\_\_ ” \_\_\_\_\_ 2022р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Дійнега Денис Олександрович

(прізвище, ім’я, по батькові)

1. Тема роботи: Мобільний додаток для організації навчання студента

керівник роботи: Колумбет Вадим Петрович, ст. викладач

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

затверджена наказом вищого навчального закладу від “8” червня 2022 р № 965-с.

2. Строк подання студентом роботи: 10 червня 2022 р.

3. Вихідні дані до роботи: мова програмування Java, мова програмування XML середовище розробки Android Studio.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): розробити мобільний додаток, за допомогою якого користувач зможе зберігати нотатки, список справ, предметів, розклад та ділитися інформацією.

5. Перелік ілюстративного матеріалу: Компоненти, аналіз існуючих систем, діаграма прецедентів, діаграма станів, дані представлені у БД, ілюстрація вікон додатку

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «10» вересня 2021 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	25 травня 2022 р.	
2.	Вивчення та аналіз задачі	02-04 травня 2022 р.	
3.	Розробка архітектури та загальної структури системи	05-06 травня 2022 р.	
4.	Розробка інтерфейсу	07-09 травня 2022 р.	
5.	Програмна реалізація системи	10-27 травня 2022 р.	
6.	Оформлення пояснювальної записки	28 травня 2022 р.	
7.	Захист програмного продукту	29 травня 2022 р.	
8.	Передзахист	09 червня 2022 р.	
9.	Захист	20 червня 2022 р.	

Студент \_\_\_\_\_ Дійнега Д.О. \_\_\_\_\_  
(підпис) (прізвище та ініціали,)

Керівник роботи \_\_\_\_\_ Колумбет В.П. \_\_\_\_\_  
(підпис) (прізвище та ініціали,)

## **АНОТАЦІЯ**

Метою даної дипломної роботи є створення мобільного додатку для організації навчання студента.

Додаток реалізований за допомогою мови програмування Java, мови розмітки XML, мови SQL та середовища розробки Android Studio.

Дипломна записка складається зі вступу, 6 розділів, висновків. Записка містить 64 сторінки, 36 рисунків.

Ключові слова: Android, мобільний додаток, органайзер, Java.

# **ANNOTATION**

The method of giving a diploma work is the creation of a mobile supplement for organizing a student's education.

Implementation add-on for additional Java programming, XML layout, SQL layout and Android Studio development environment.

Graduation note is folded in the beginning, 6 divisions, conclusions. Note to revenge 64 sides, 36 drawings.

Key words: Android, mobile add-on, organizer, Java.

# ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 ПОСТАНОВКА ЗАДАЧІ.....	10
1.1 Призначення додатку .....	10
1.2 Створення моделі БД та додатку .....	10
1.3 Створення програмного інтерфейсу .....	11
1.4 Створення алгоритму для імпорту та експорту даних .....	11
2 ОПИС СИСТЕМИ .....	12
2.1 Діаграма прецедентів .....	13
2.2 Діаграма компонентів .....	14
2.3 Діаграма станів .....	15
ВИСНОВКИ ДО РОЗДІЛУ 2 .....	16
3 ОПИС ІСНУЮЧИХ РІШЕНЬ .....	17
ВИСНОВКИ ДО РОЗДІЛУ 3 .....	29
4 ЗАСОБИ РОЗРОБКИ .....	30
4.1 Операційна система Android .....	30
4.2 Мова програмування Java .....	31
4.3 Бази даних SQL .....	31
4.4 Середовище розробки Android Studio .....	31
4.5 Мова XML.....	34
4.6 Вимоги до обладнання.....	34
ВИСНОВКИ ДО РОЗДІЛУ 4 .....	35
5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ .....	36
5.1 Створення структури проекту .....	36
5.2 Створення моделі БД.....	37
5.3 Створення інтерфейсу .....	41
5.4 Створення Activity .....	43
5.5 Створення алгоритму для імпорту і експорту інформації.....	44

5.6 Створення алгоритму для автоматичного додавання списку предметів та розкладу.....	45
ВИСНОВКИ ДО РОЗДІЛУ 5 .....	46
6 ІНСТРУКЦІЯ РОБОТИ З ПРОГРАМОЮ .....	47
ВИСНОВКИ ДО РОЗДІЛУ 6 .....	62
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	64
ДОДАТОК А.....	65

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

**БД** – бази даних.

**ОС** – операційна система.

**SQL** – Structured Query Language

**XML** – eXtensible Markup Language. Мова розмітки.

**Мобільний додаток** – програма, яка була розроблена для мобільних телефонів або планшетів.

**Органайзер** – блокнот, щоденник для записів про планування подій, зустрічей, заходів, завдань.

**Android** – операційна система для мобільних телефонів.

**API** – Application Programming Interface.



## ВСТУП

У зв'язку з пандемією навчання в усьому світі змінилося. Більшість начальних закладів перейшло на дистанційне навчання. В такому режимі навчання студенти зазвичай отримують всю необхідну інформацію через веб-застосунки в комп'ютері. Я вважаю, що це не вірний підхід до організації свого навчання.

В наш час люди студентського віку багато часу проводять дивлячись у свій смартфон. За статистикою цей час приблизно дорівнює 3 годинам на день. Це складає майже 20% від усього дня, не включаючи час відведений на сон. Тому організація свого навчання через мобільний девайс є більш зручнішою.

Студент може просто забути про крайній час виконання заданого завдання або про пару, яка розпочнеться за декілька хвилин, адже щоб переглянути цю інформацію, він повинен подивитися її у браузері. Цю проблему можуть вирішити сповіщення які будуть надходити на його смартфон, та нагадувати про пару або завдання, яке потрібно здати завтра.

Тому я вирішив розробити додаток, за допомогою якого користувач буде мати можливість контролювати своє навчання: створювати список предметів, розкладу, додавати завдання та отримувати нагадування про початок зайнять та наближення терміну здачі робіт, ділитися цією інформацією з іншими студентами. Результатом роботи буде додаток для платформи Android який буде виконувати вищезазначені функції.

# **1 ПОСТАНОВКА ЗАДАЧІ**

Була поставлена задача розробки мобільного додатку, що буде допомагати організувати навчання студента. Користувач повинен мати можливість створювати список предметів, розклад, добавляти завдання та нотатки, отримувати сповіщення про початок пари та наближення терміну здачі завдань, екзамену, ділитися інформацією з іншими студентами. Користувач може додати розклад та список предметів автоматично.

## **1.1 Призначення додатку**

Додаток орієнтований на учнів вищих та середньо освітніх закладів. Учні та студенти зможуть власноруч організовувати власне навчання шляхом додавання інформації в додаток, перегляду її та отримання нагадувань.

Додаток буде актуальним як при дистанційному так і при очному навчанні. Але він має функціонал, щоб користуватися їм і не на протязі навчального процесу.

## **1.2 Створення моделі БД та додатку**

Додаток повинен мати можливість зберігати інформацію, сортувати данні, робити вибірку по полям БД. База даних повинна зберігатися в пам'яті пристрою, щоб користувач не втратив своїх записів. Додаток повинен мати можливість додавати предмети та розклад автоматично.

### **1.3 Створення програмного інтерфейсу**

Додаток повинен мати зручний інтерфейс, який надає можливість переглядати списки завдань, нотаток, розкладу, отримувати сповіщення, редагувати та добавляти інформацію. Програма повинна мати вікно з налаштуваннями.

### **1.4 Створення алгоритму для імпорту та експорту даних**

Додаток повинен надавати можливість конвертувати поля з БД в текстову строку та зчитувати її для імпорту інформації, створеної іншим користувачем. Користувач додатку повинен мати можливість загрузити список предметів та розклад автоматично за допомогою API.

## 2 ОПИС СИСТЕМИ

Система повинна надавати користувачу можливість додавати, зберігати та редагувати інформацію про нотатки, список справ, предмети і розклад та ділитися або отримувати цю інформацію.

Будь який мобільний додаток складається з користувацького інтерфейсу та програмної частини. Основні елементи проекту це Java класи та .xml файли з розміткою інтерфейсу.

Спочатку створюється модель системи: сутності для БД та методи звернення до таблиць, основні класи. Сама база даних зберігається локально на пристрої користувача.

Після цього будується сам інтерфейс, додаються його основні елементи: кнопки, поля для вводу інформації тощо.

Інтерфейс додатку містить списки для виводу інформації. Зображення окремого елемента списку розмічається в окремому файлі. Кожен список повинен містити окремий клас, який відповідає за розташування та зміну інформації в окремому елементі.

Коли розмітка інтерфейсу завершена створюються класи, в яких містяться методи для взаємодії з інтерфейсом: зчитування натискання кнопок, введення тексту тощо. Отримавши інформацію про взаємодію з інтерфейсом, методи роблять відповідні дії. Зазвичай вони роблять запити до БД щоб змінити або додати інформацію.

## 2.1 Діаграма прецедентів

Діаграма прецедентів зображена на рисунку 2.1



Рисунок 2.1 – Діаграма прецедентів

Діаграма прецедентів демонструє, які можливості має користувач користуючись додатком.

## 2.2 Діаграма компонентів

Діаграма зображена на рисунку 2.2

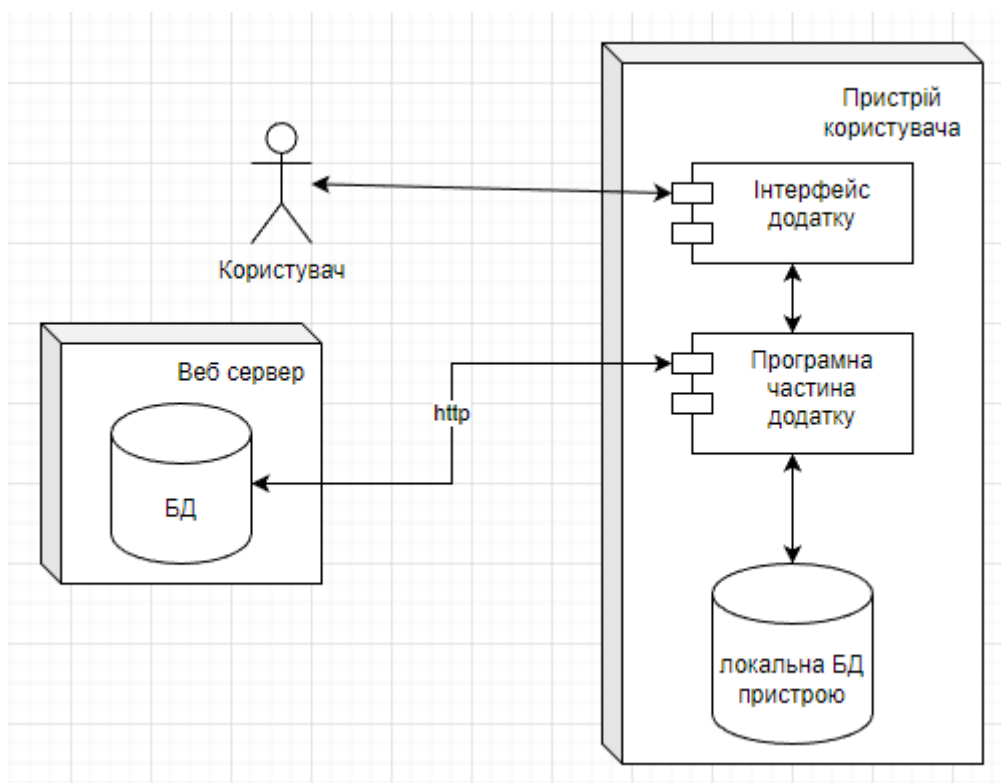


Рисунок 2.2 – Діаграма компонентів

Діаграма демонструє основні компоненти додатку.

Користувач взаємодіє з додатком за допомогою мобільного пристрою. Інтерфейс надає можливість користувачу через програмну частину робити запити до БД: зберігати, редагувати, видаляти інформацію. Додаток може робити запити до API щоб додавати інформацію автоматично.

## 2.3 Діаграма станів

Розглянемо декілька можливостей додатку. Користувач має можливість додавати завдання для предмету. Розглянемо діаграму станів для додавання завдання. Діаграма зображена на рисунку 2.3

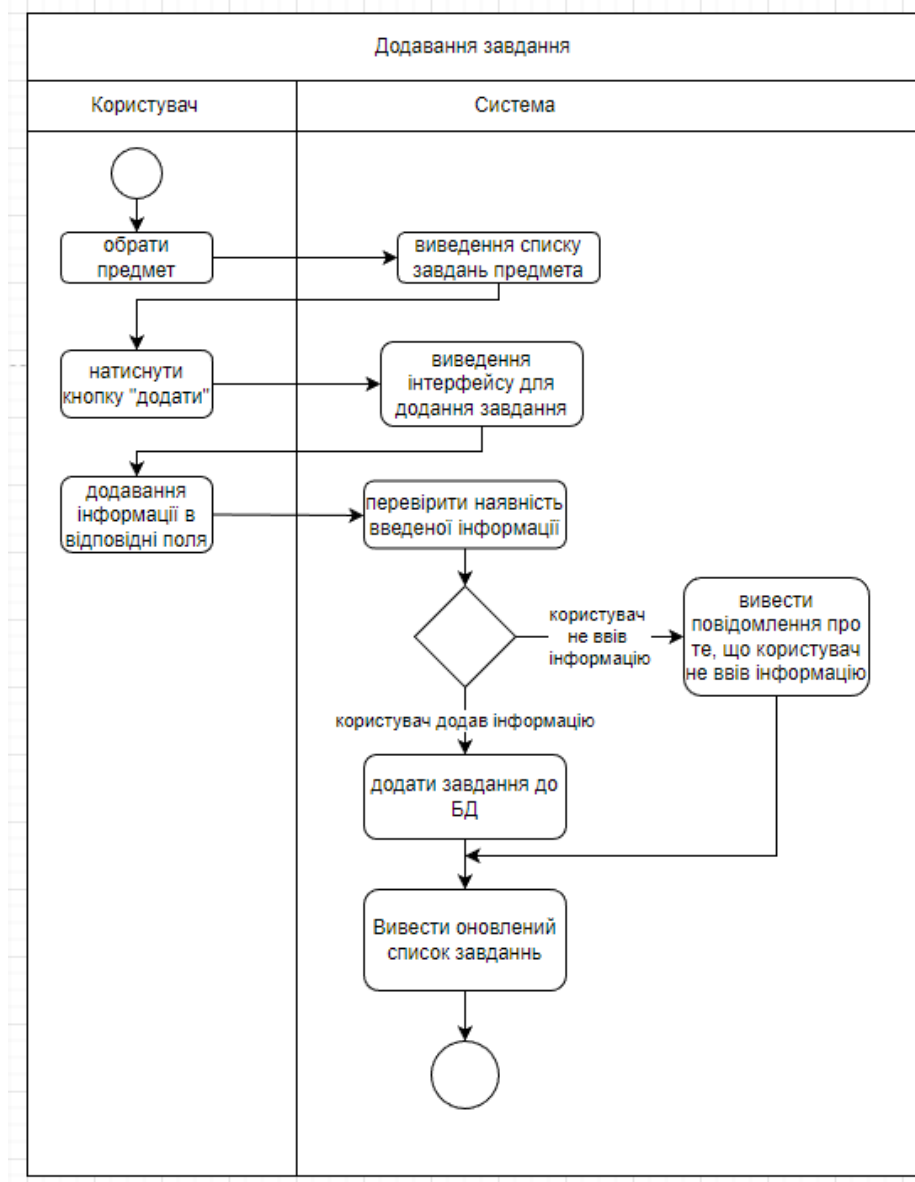


Рисунок 2.3 – Діаграма станів

На діаграмі зображується процес створення завдання

## **ВИСНОВКИ ДО РОЗДІЛУ 2**

У цьому розділі було визначено з чого повинна складатись система, які функції виконувати.

Було розглянуто функції, якими володіє користувач та зображено їх на діаграмі прецедентів.

Було визначено основні компоненти додатку то зображено їх на діаграмі компонентів.

Було продемонстровано взаємодію користувача з додатком на прикладі додавання завдання та створено діаграму станів.



### 3 ОПИС ІСНУЮЧИХ РІШЕНЬ

На сьогоднішній день існує велика кількість додатків органайзерів. Розглянемо деякі з них.

1. Пре інстальований додаток “Нотатки” на смартфонах Xiaomi:

Додаток пре інстальований в смартфонах як стандартний органайзер.

Головне вікно додатку зображене на рисунку 3.1.

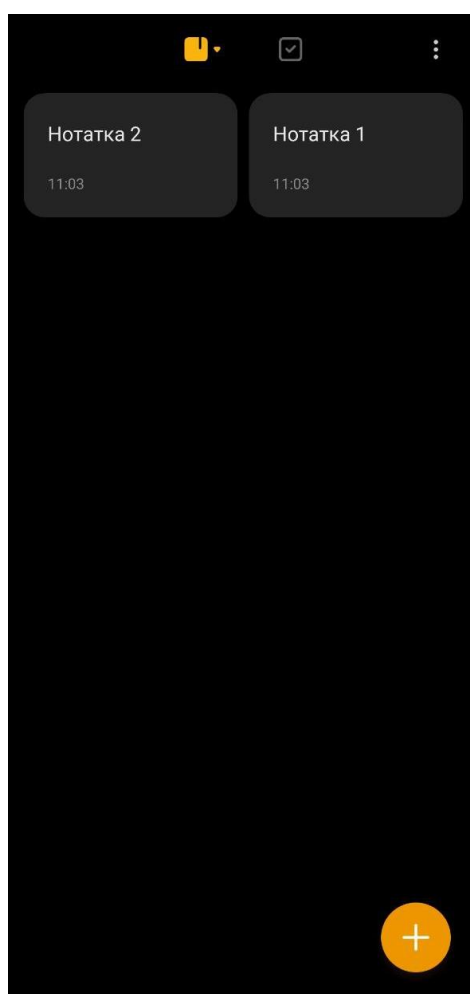


Рисунок 3.1 – Інтерфейс додатку “Нотатки”

На головному вікні зображено список нотаток. Нотатки сортуються за часом створення. Реалізовано пошук нотаток за словами.

Вікно створення нотатки має вигляд, зображений на рисунку 3.2.

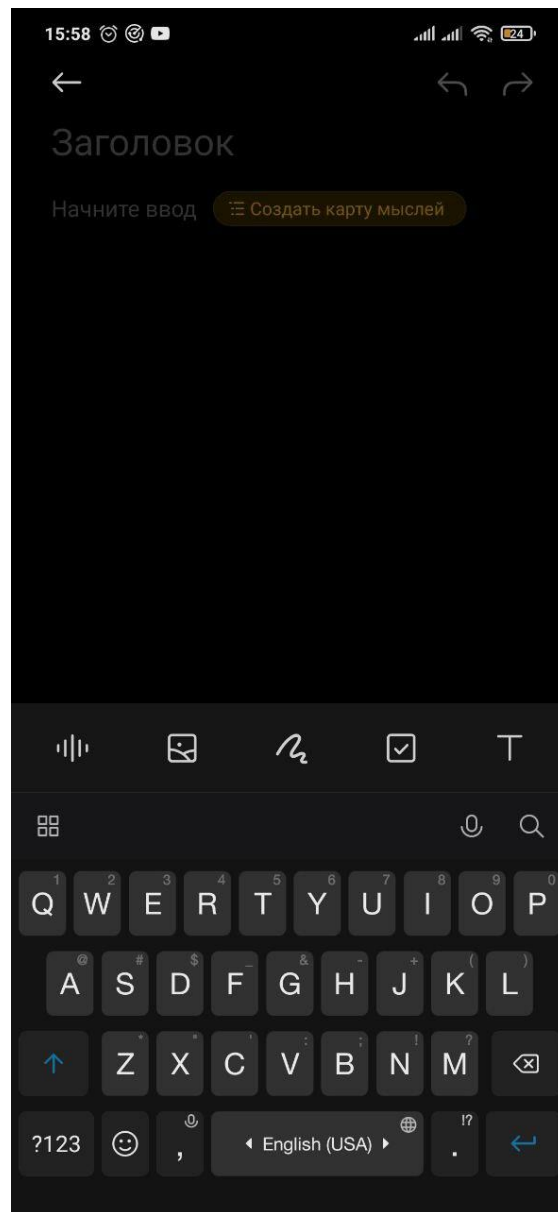


Рисунок 3.2 – Вікно створення нотатки

Користувач має можливість обрати стиль тексту нотатки, додати зображення. Вікно зі списком справ має вигляд, зображений на рисунку 3.3.

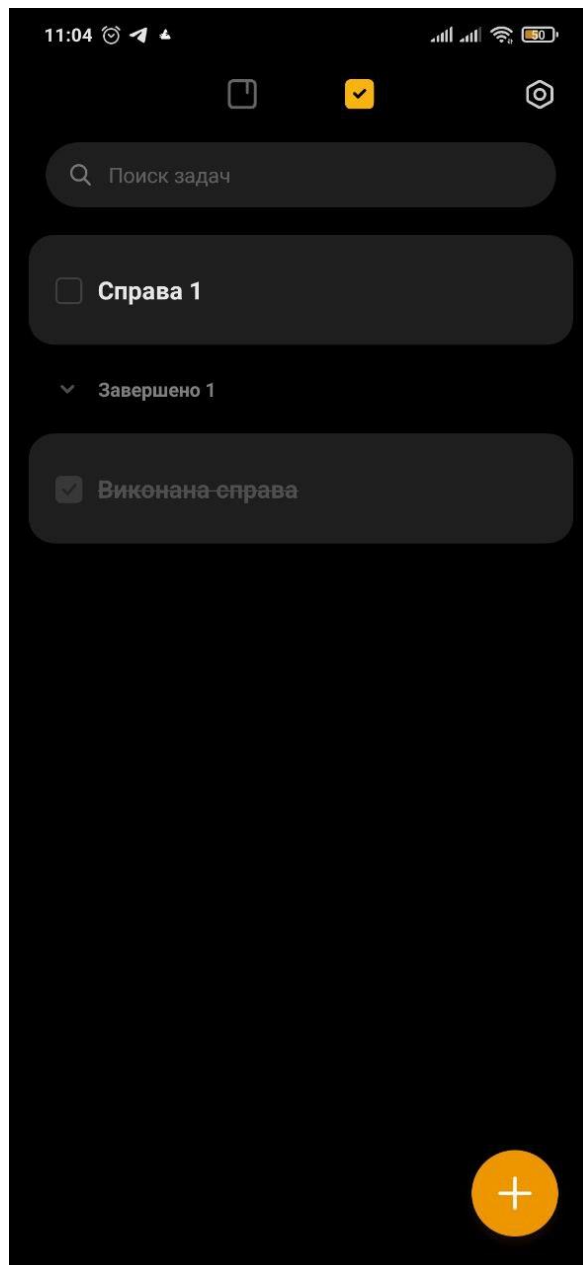


Рисунок 3.3 – Список справ

В цьому вікні зображений список справ, розділений на виконані та невиконані справи.

В користувача є такі можливості налаштування: обрати розмір шрифту, встановити за яким параметром будуть сортуватися нотатки та вимкнути нагадування. Вікно налаштувань додатку має вигляд, зображений на рисунку 3.4.

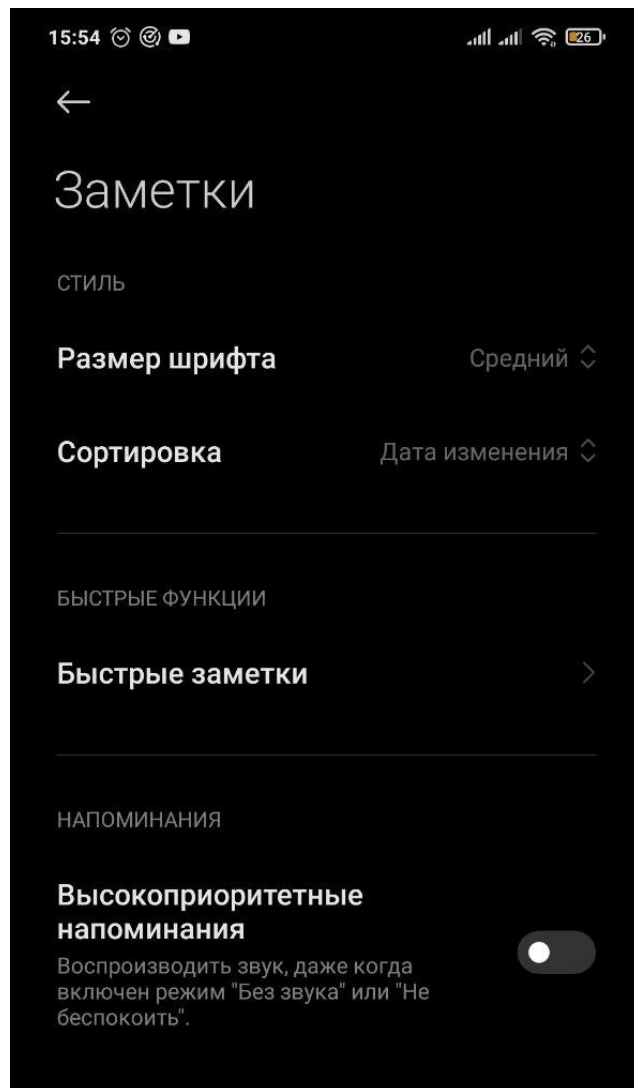


Рисунок 3.4 – Налаштування додатку

Недоліки: немає можливості групувати список справ, створювати розклад, передавати інформацію на інший пристрій.

## 2. Додаток “Список задач”

Наступну програму можна інсталиювати з офіційного каталогу в Android – Play Market.

Головне вікно програми зображено на рисунку 3.5.



Рисунок 3.5 – Головне вікно додатку “Список задач”

Ця програма має більший функціонал, ніж попередня. В додатку є можливість створювати окремі списки справ та сортувати їх.

На рисунку 3.6 зображено вікно, з вибором списку справ.

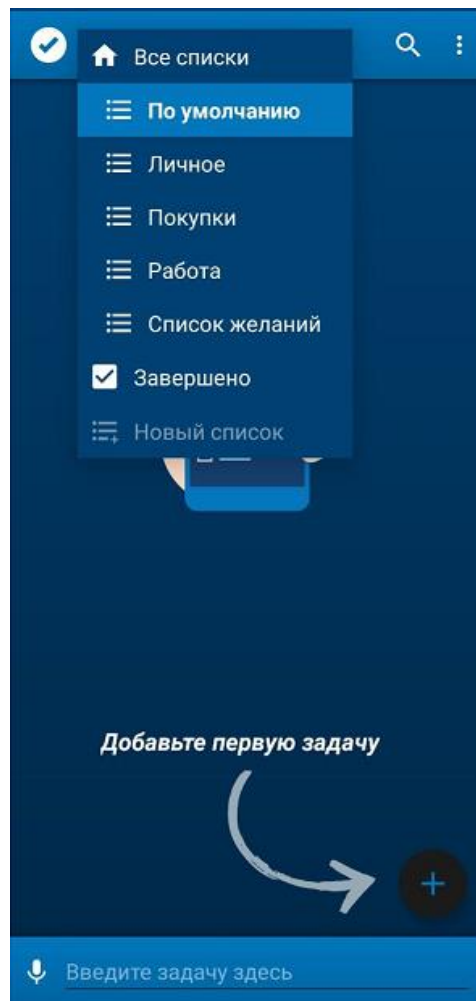


Рисунок 3.6 – Вибір списку справ

При створенні нової задачі, користувач має можливість обрати групу для неї та встановити час нагадування. Також додаток має підтримку голосового вводу тексту. Вікно створення задачі зображено на рисунку 3.7.

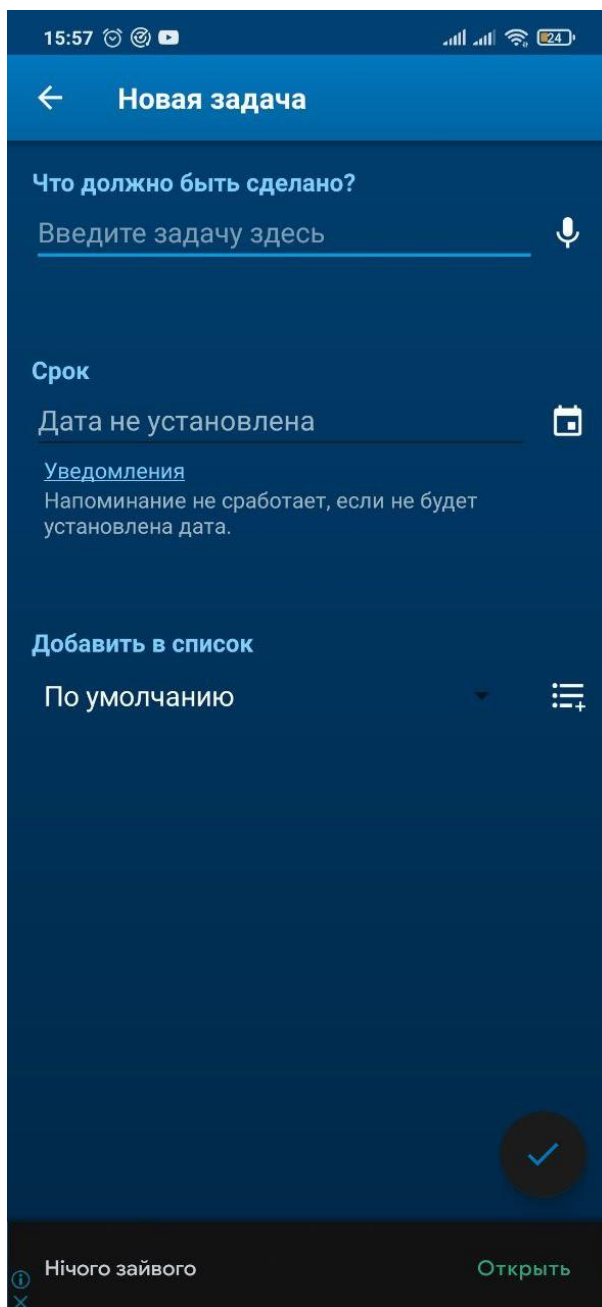


Рисунок 3.7 – Вибір списку справ

Додаток надає можливість синхронізувати нотатки з обліковим записом Google.

Користувач має такі можливості налаштування: встановити список справ за замовчуванням, налаштувати формат часу та вимкнути повідомлення. Вікно налаштувань має вигляд, зображений на рисунку 3.8.

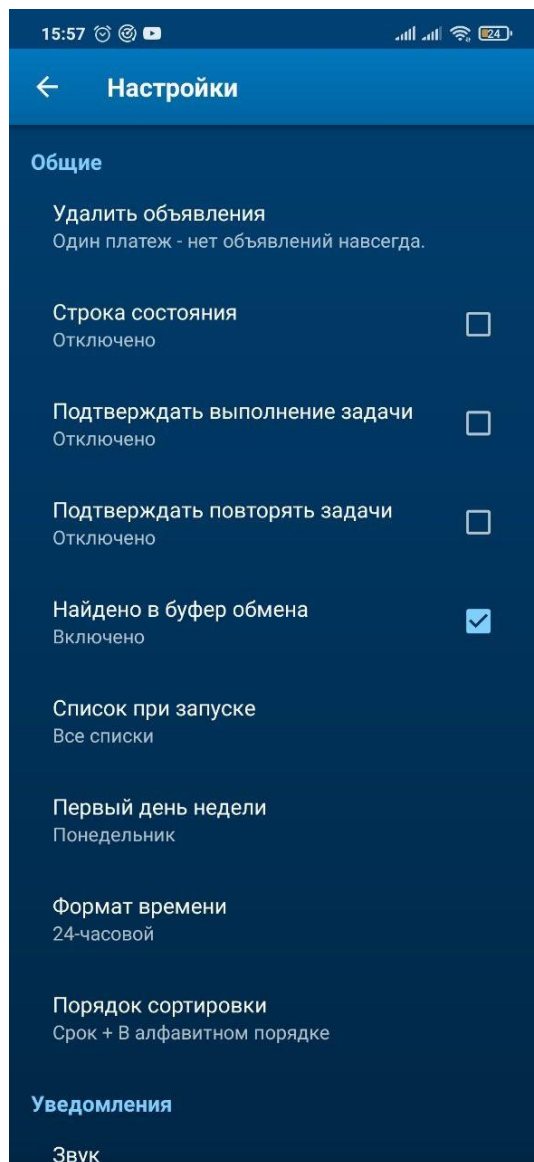


Рисунок 3.8 – Налаштування в додатку

Недоліки: не має можливості створення розкладу та звичайних нотаток. Передавати данні між пристроями можливо лише через авторизацію на іншому пристрої. Також в додатку присутня реклама.

Побудова власного додатку буде враховувати недоліки аналогів: можливість створення розкладу, різних видів нотаток, передача інформації без авторизації на пристрої.



### 3. Мобільний додаток “Розклад КПП”

Головне вікно додатку має вигляд, зображений на рисунку 3.9.

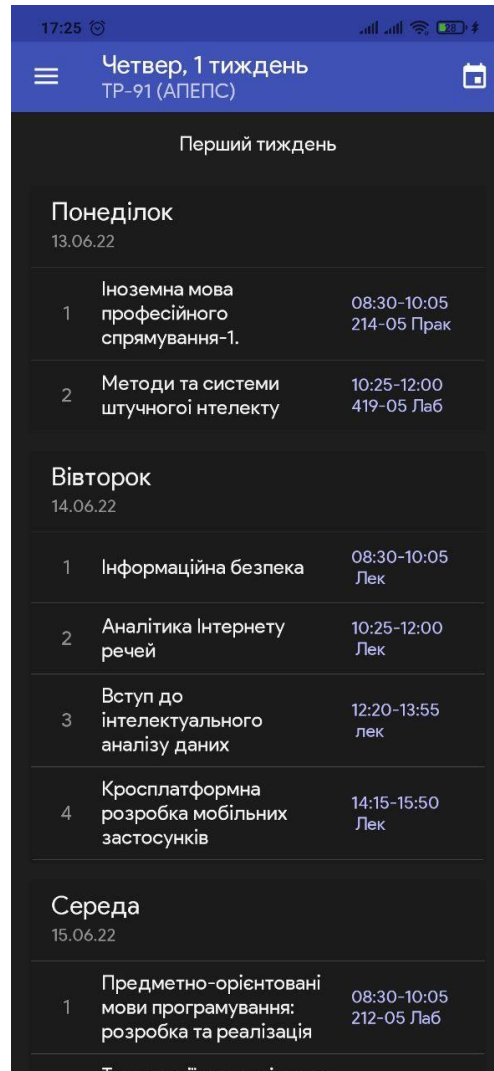


Рисунок 3.9 – Головне вікно додатку

На головному вікні додатку зображено розклад групи. Розклад завантажується з бази даних університету.

Меню додатку має вигляд, зображений на рисунку 3.10.

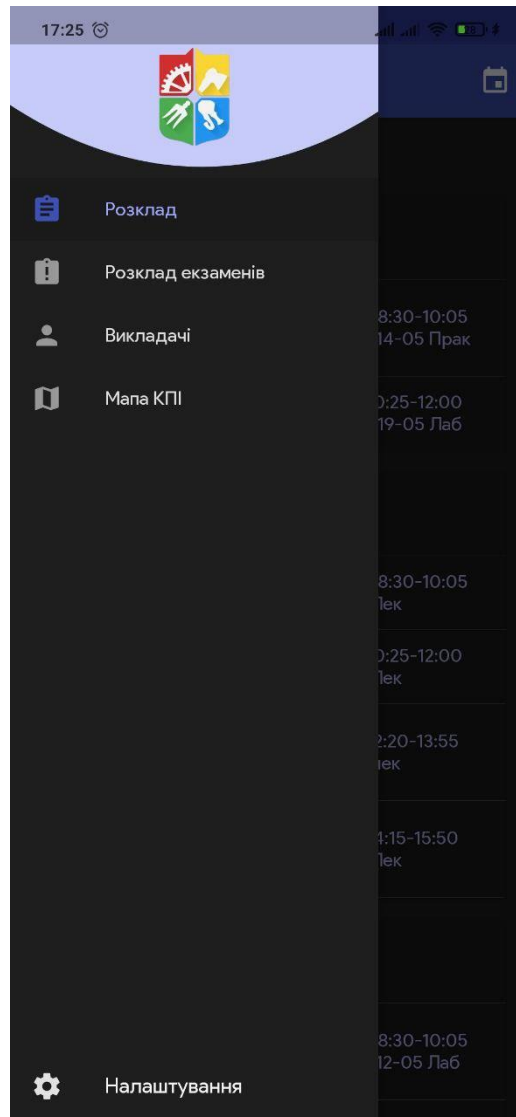


Рисунок 3.10 – Меню додатку

Користувач має можливість переглянути розклад занять, розклад екзаменів, список викладачів та мапу університету.

При натисканні на окрему пару вона відкривається у новому вікні, яке має вигляд, зображений на рисунку 3.11.

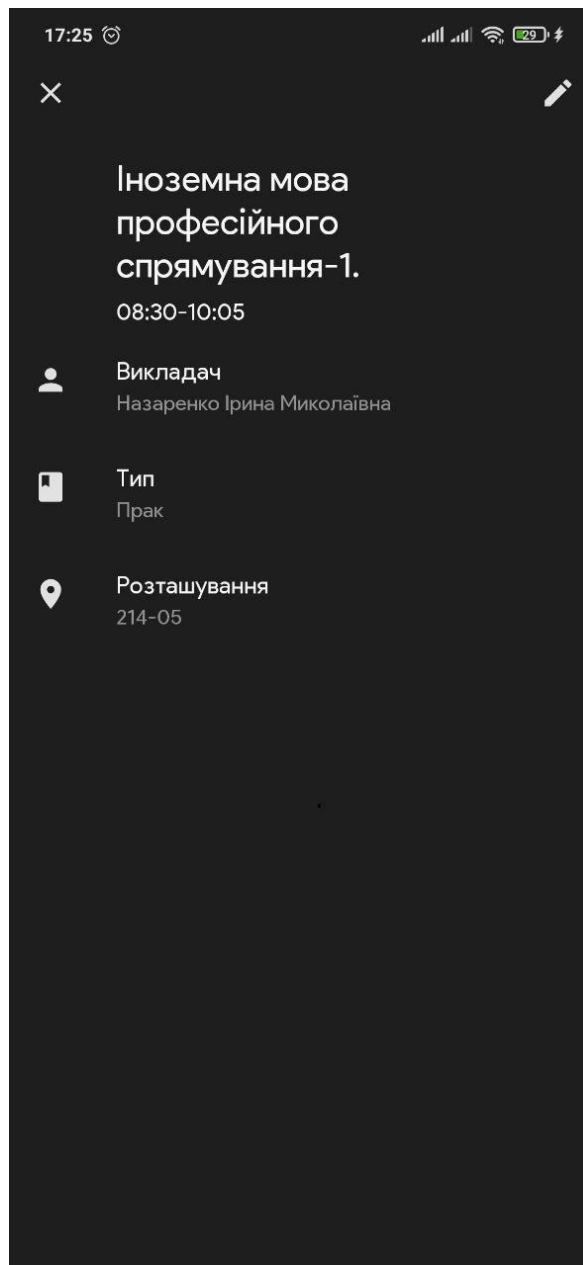


Рисунок 3.11 – Зображення пари

Користувач отримує інформацію про пару: час початку, прізвище ім'я по-батькові викладача та кабінет, в якому проходить пара.

Вікно з налаштуваннями має вигляд, зображений на рисунку 3.12.

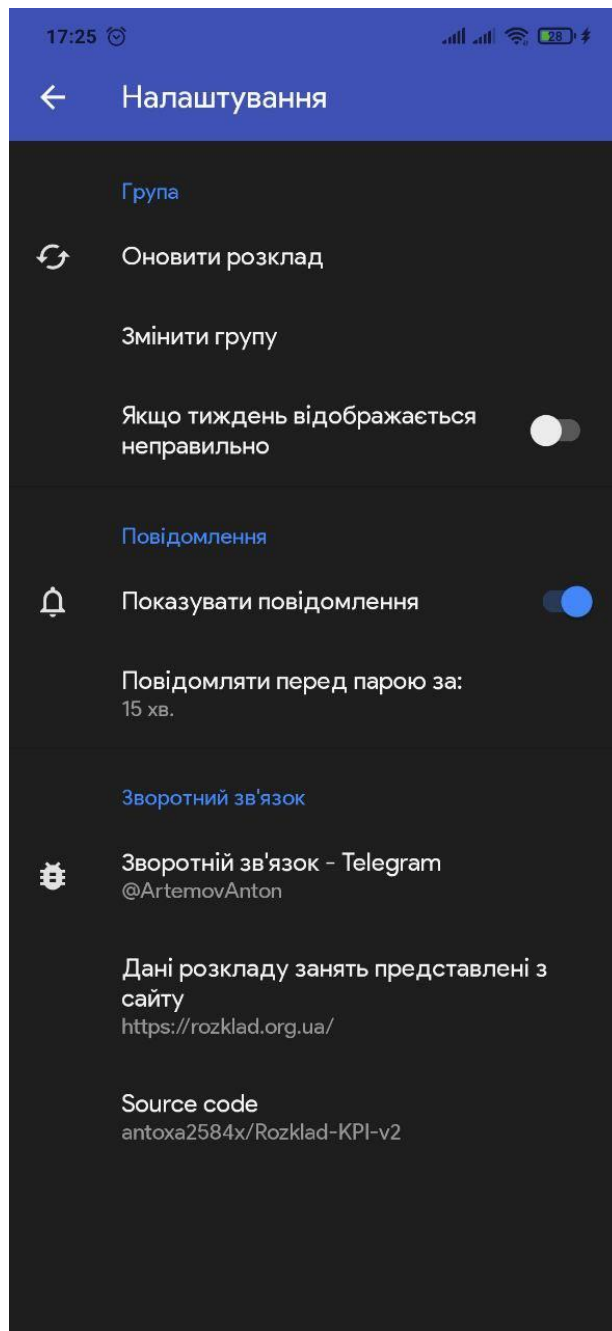


Рисунок 3.12 – Налаштування додатку

В додатку можливі такі налаштування: можливість оновити розклад, змінити групу, налаштувати повідомлення.

Розклад завантажується автоматично з бази даних університету.

Недоліки: неможливість робити окремі нотатки та додавати список справ. Немає можливості додавати завдання на окремий предмет.

## **ВИСНОВКИ ДО РОЗДІЛУ 3**

У цьому розділі було проаналізовано існуючі рішення на прикладі 3 додатків. Було проаналізовано структуру їхнього інтерфейсу.

Були виявлені недоліки існуючих додатків, які були враховані при створенні власного додатку, а саме:

- Додаток повинен мати зручний інтерфейс
- Додаток повинен мати більш ширший функціонал ніж існуючі – додання нотаток, списку справ, предметів, завдань, розкладу.
- Додаток повинен мати можливість передавати інформацію на інший пристрій та зчитувати її.
- Користувач повинен мати можливість завантажити список предметів та розклад автоматично.

## **4 ЗАСОБИ РОЗРОБКИ**

В цьому розділі розглянемо інструменти, за допомогою яких створюався додаток.

### **4.1 Операційна система Android**

Android – одна з найпоширеніших ОС у світі. На цій системі працюють не лише мобільні пристрої а і планшети, фітнес браслети, годинники, електронні книги. В 2017 році на ОС Android було активовано понад 2 млрд. пристроїв. У 2014 році 86% всіх проданих смартфонів у світі працювали на цій ОС.

На відміну від свого головного конкурента IOS, Android – відкрита система, що дає розробнику набагато більше можливостей для створення додатків.

Першу версію Android було оприлюднено в 2008 році. З цього часу вихідний код ОС знаходиться у вільному доступі.

Android має власний онлайн магазин додатків. Через нього можна загрузити на смартфон ігри, месенджери, додатки для управління фінансів, тощо.

Android дозволяє використовувати всі функції API, що використовуються програмами ядра. Архітектура побудована таким чином, що будь-яка програма може використовувати вже реалізований функціонал іншої програми, якщо вона надає доступ до можливості. Таким чином, архітектура втілює в собі принцип багаторазового використання компонентів ОС та додатків.

Основні труднощі при розробці для платформи Android виникають тому, що система встановлена на велику кількість різних пристроїв з різними видами екрану. Через це можуть виникати проблеми з сумісністю.

## **4.2 Мова програмування Java**

Java — це мова програмування загального призначення. Об'єктно-орієнтована мова програмування, яка є суворо типізованою мовою. Має механізм Garbage Collector, який полегшує роботу з пам'яттю.

Java має синтаксис схожий на синтаксис C++. Сама модель мови досить схожа на C-подібні мови. На Java легше писати об'єктно орієнтовані програми. Багато чого віртуальна машина мови робить замість програмістів.

Java підходить для різних задач. Цією мовою можна створювати мобільні додатки, додатки для комп'ютера, ігри, веб застосунки.

Так як Android написана саме на Java то ця мова ідеально підходить для розробки мобільних додатків для цієї ОС.

## **4.3 Бази даних SQL**

Android підтримує одну з систем управління SQLite. Для автоматизації роботи з SQLite також була використана бібліотека Room. Room надає можливість створювати SQL запити безпосередньо з коду Java та створювати версії бази даних.

## **4.4 Середовище розробки Android Studio**

Android Studio — інтегроване середовище розробки розроблене для створення додатків для ОС Android. Android Studio було створене на базі IntelliJ IDEA.

Android Studio адаптоване саме для розробки Android додатків. Середовище надає можливість працювати з різними версіями ОС, має вбудовані засоби тестування.

В середовищі можна працювати з інтерфейсом додатка не лише через .xml файли а й через оновлюючий макет екрану, який зображений на рисунку 4.1.

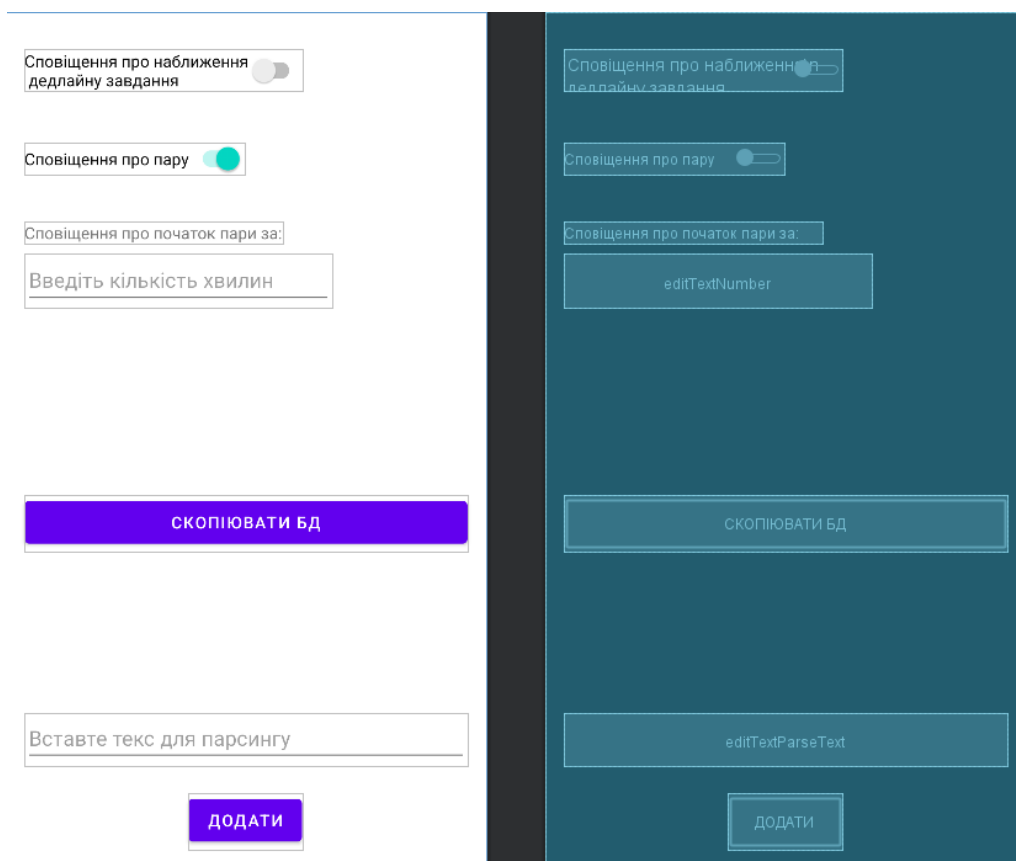


Рисунок 4.1 – Макет екрану в Android Studio

Також має вбудований емулятор смартфона з великим списком моделей для тестування застосунку. Емулятор зображений на рисунку 4.2.



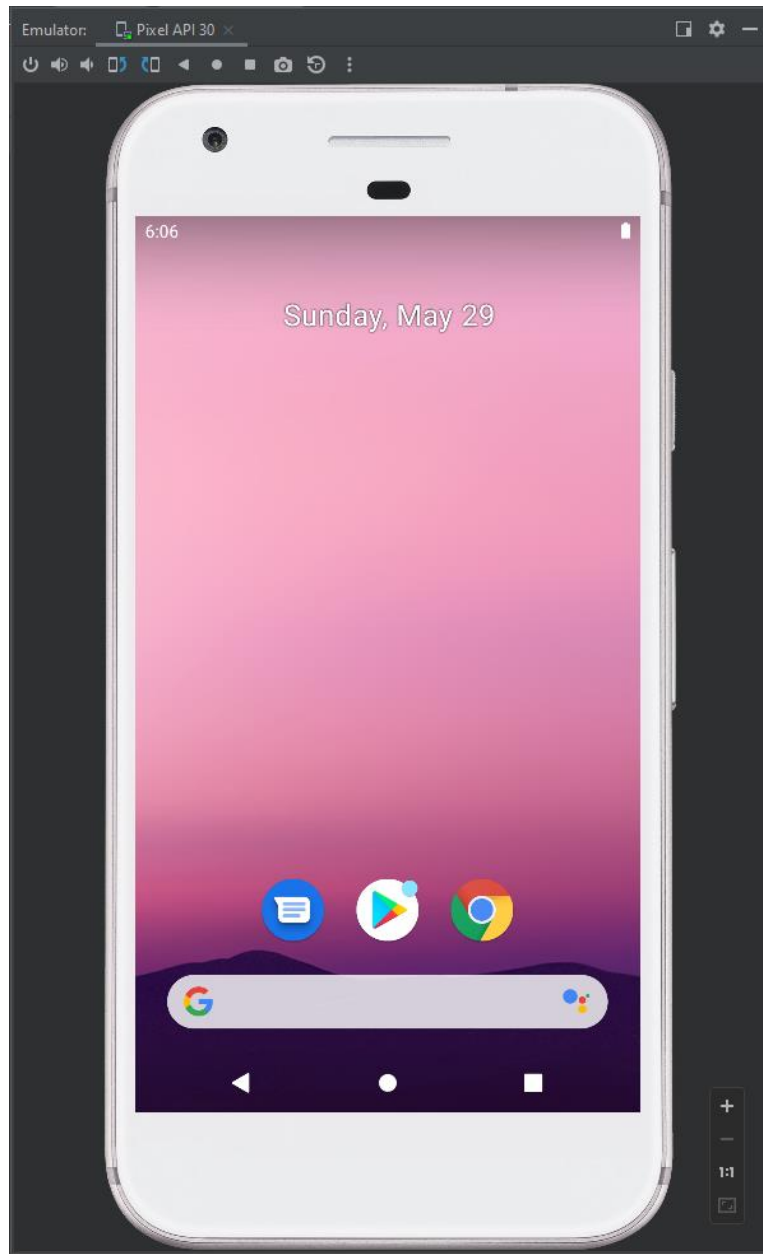


Рисунок 4.2 – Емулятор в Android Studio

Надає можливість підключати зовнішній пристрій для тестування. Android Studio само створює макет проекту та має зручний доступ до підключення бібліотек.

## 4.5 Мова XML

XML – мова розмітки. Складається з інформації, розташованої в тегах.

Переваги мови XML:

- Єдиний стандарт
- Код написаний цією мовою легко сприймається
- Підтримує дерева розгалуження

Багато людей плутають мови XML і HTML. Але в них різні призначення.

Основне завдання HTML – зображувати інформацію. Основне призначення XML – збереження та передача інформації.

Головна відмінність цих мов полягає в тому, що всі теги в мові HTML задекларовані стандартами. Теги ж XML користувач може створювати сам.

## 4.6 Вимоги до обладнання

Щоб запустити додаток, користувач повинен мати пристрій на ОС Android версії не нижче 5.0. Для завантаження розкладу та списку предметів необхідне інтернет з'єднання.

## **ВИСНОВКИ ДО РОЗДІЛУ 4**

У розділі було проаналізовано та обрано інструменти для розробки додатку.

А саме:

- ОС Android
- Мову програмування Java для реалізації програмної частини
- Мову розмітки XML для створення інтерфейсу
- Мову програмування SQL для створення БД
- Середовище розробки Android Studio

Було визначено мінімальні вимоги до обладнання користувача.

## 5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

В цьому розділі описано програмну реалізацію додатка.

### 5.1 Створення структури проекту

Структура проекту зображена на рисунку 5.1.

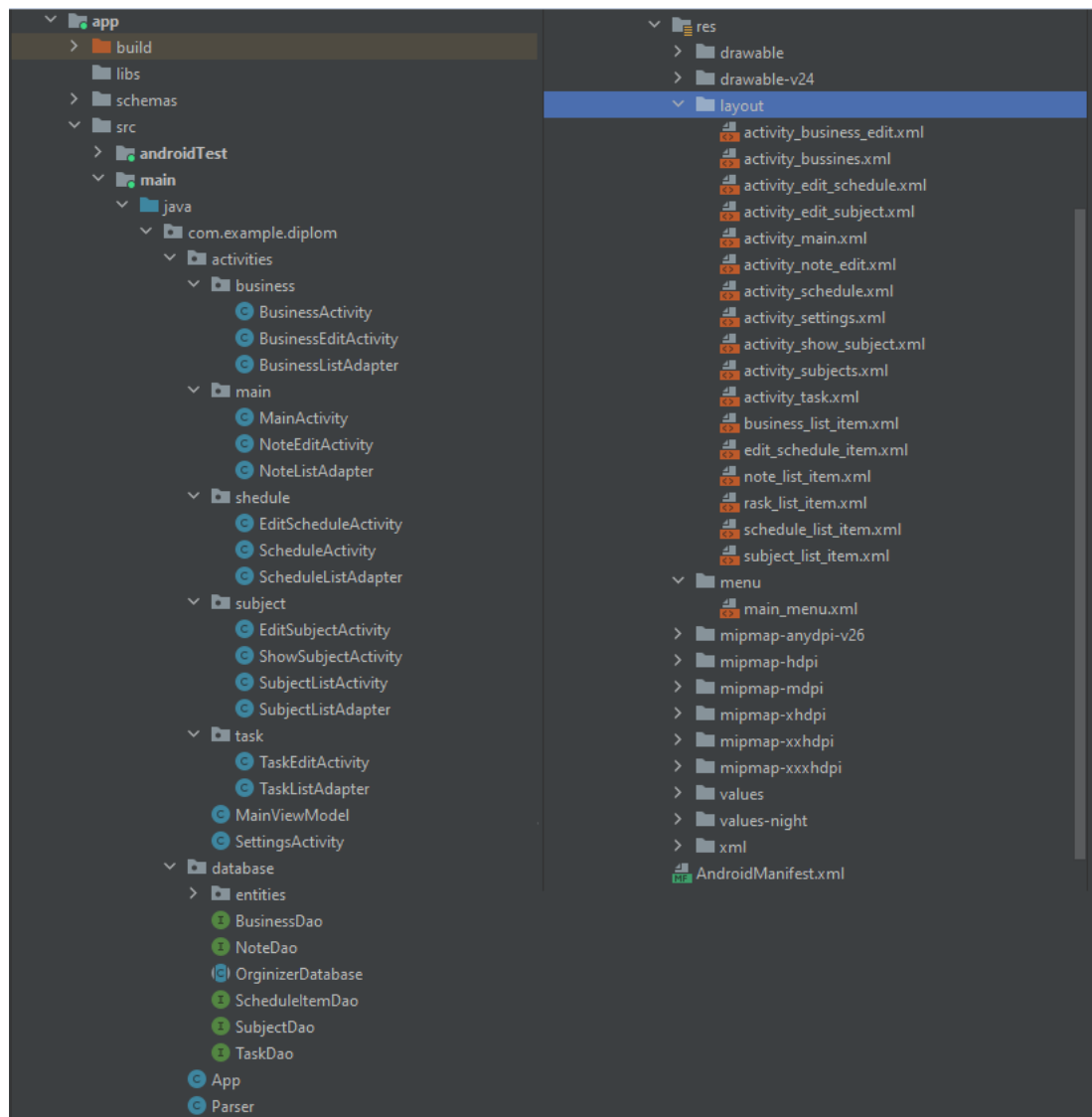


Рисунок 5.1 – Структура проекту

Всі класи та xml файли знаходяться в каталозі Java. Для зручності в проект було додано додаткові підкаталоги activities і database.

Activities зберігає всі класи Activity та адаптера для списків. В цьому каталозі зберігається основна програмна частина проекту.

Database зберігає модель бази даних: Entities, Daos, Database.

Клас App визивається програмою при її запуску. В ньому визивається метод створення БД, та зберігаються глобальні змінні. Клас існує до закінчення роботи додатку.

В файлі AndroidManifest.xml зберігається вся інформація про конфігурацію додатка: назву проекту, список вікон додатку.

## **5.2 Створення моделі БД**

Модель БД була створена за допомогою бібліотеки Room.

Room має такі компоненти:

- Клас типу Database, який зберігає в собі інформацію про БД
- Класи типу Entity, які містять таблиці БД
- Класи типу Dao, які містять методи для виконання SQL звернень.

Схема роботи БД зображена на рисунку 5.2.

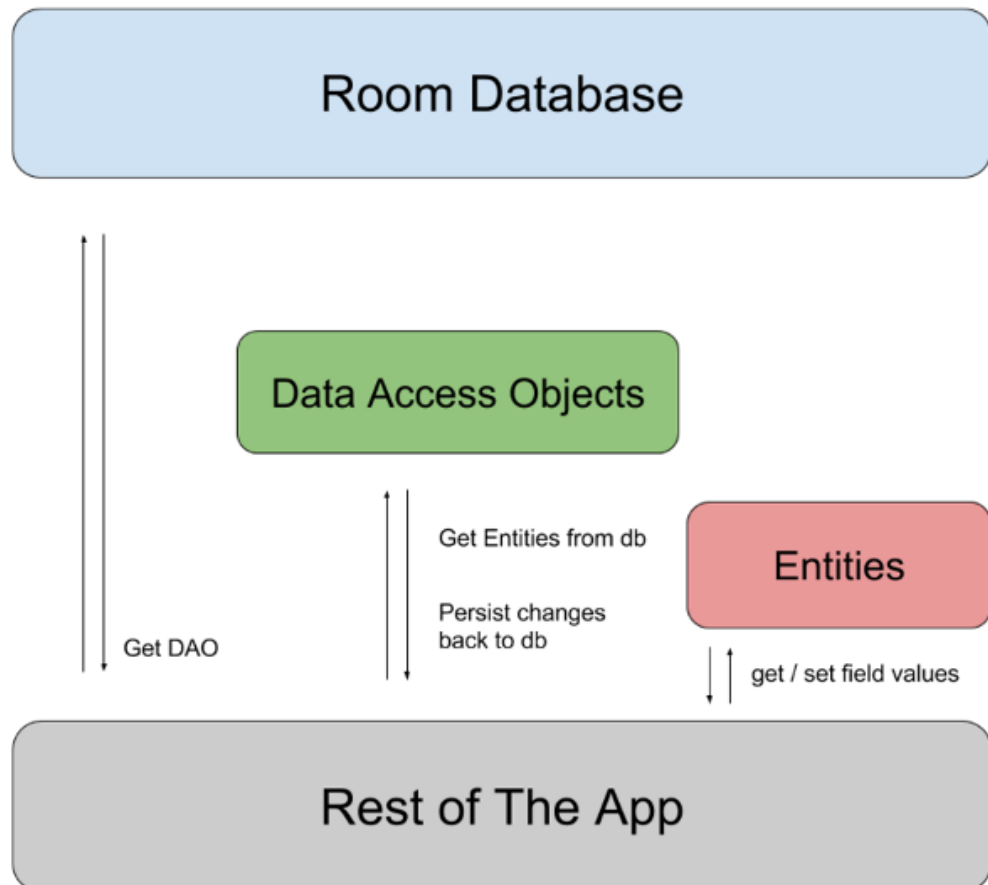


Рисунок 5.2 – Схема роботи БД з бібліотекою Room

БД застосунку складається з 5 сутностей: Task, Subject, Note, Business, ScheduleItem.

Task містить дані про завдання (назву завдання, дату виконання, назву предмету).

Subject містить дані про предмет (назву предмету, ПІБ викладача, тип кредиту, посилання на онлайн пару).

Note містить дані про нотатку (текст нотатки, чи добавлена нотатка в обране).

Business містить дані про справу (текст справи, чи виконана справа).

ScheduleItem містить дані про пару (день неділі, час початку пари, посилання на онлайн пару).

Для об'єктів типу Entity існують спеціальні анотації:

@PrimaryKey встановлює змінну як ключ таблиці.

Має змінну autoGenerate. Якщо вона дорівнює true то ключ бази даних буде оновлюватись самостійно

@ColumnInfo встановлює змінну як стовпчик таблиці.

Змінна name містить назву стовпчика.

Для кожної сутності був створений Data access object. Він дозволяє сутності взаємодіяти з додатком, виконувати різні SQL запити, добавляти, видаляти, редагувати данні з таблиць.

Для об'єктів типу Dao існують спеціальні анотації:

- @Insert додає об'єкт в таблицю БД.
- @Update оновлює об'єкт в таблиці БД.
- @Delete видаляє об'єкт з таблиці БД.
- @Query() надає можливість робити запити до таблиць на мові SQL. Для цього потрібно додати в метод SQL запит в змінній типу String.

Приклад запиту: @Query("SELECT \* FROM Task");

Для всієї бази даних був створений клас OrganizerDatabase який зберігає інформацію про всю БД. В цей клас потрібно ввести всі об'єкти типу Entity та ініціалізувати об'єкти типу @Dao;

Таблиці БД зображені на рисунку 5.3.

tableName	fields				primaryKey	
Note	fieldPath	columnName	affinity	notNull	columnNames id autoGenerate true	
	id	id	INTEGER	true		
	noteText	note_text	TEXT	false		
	favorites	note_favorites	INTEGER	true		
Business	fieldPath	columnName	affinity	notNull	columnNames id autoGenerate true	
	id	id	INTEGER	true		
	businessText	business_text	TEXT	false		
	businessDone	business_done	INTEGER	true		
Subject	fieldPath	columnName	affinity	notNull	columnNames id autoGenerate true	
	id	id	INTEGER	true		
	subjectName	subject_name	TEXT	false		
	teacherName	teacher_name	TEXT	false		
	zoomLink	zoom_link	TEXT	false		
	isExs	is_exs	INTEGER	true		
Task	fieldPath	columnName	affinity	notNull	columnNames id autoGenerate true	
	id	id	INTEGER	true		
	taskText	task_text	TEXT	false		
	subjectName	subject_name	TEXT	false		
	year	year	INTEGER	true		
	sId	s_id	INTEGER	true		
	month	month	INTEGER	true		
	day	day	INTEGER	true		
	isTaskDone	is_task_done	INTEGER	true		
ScheduleItem	fieldPath	columnName	affinity	notNull	columnNames id autoGenerate true	
	id	id	INTEGER	true		
	subjectName	subject_name	TEXT	false		
	weekDay	week_day	TEXT	false		
	zoomLink	zoom_link	TEXT	false		
	hour	hour	INTEGER	true		
	minute	minute	INTEGER	true		
	sId	s_id	INTEGER	true		

Рисунок 5.3 – Таблиці БД

Рисунок демонструє всі поля таблиць на їх властивості.



## 5.3 Створення інтерфейсу

Інтерфейс додатків Android створюється за допомогою xml файлів. В них додаються елементи інтерфейсу: кнопки, поля для вводу та редагування тексту, текстові поля, перемикачі, віджети вибору часу та дати.

На рисунку 5.4 зображені всі файли .xml.

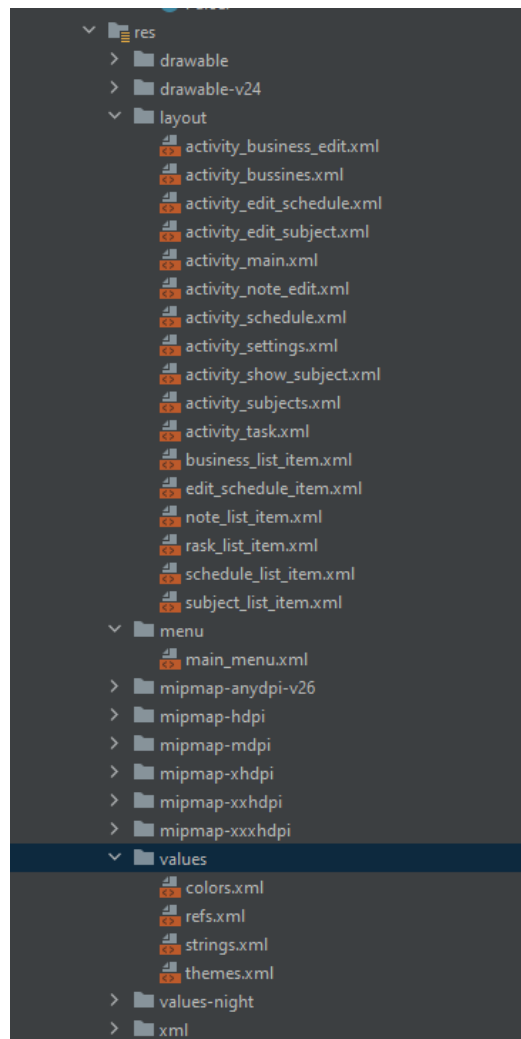


Рисунок 5.4 – Список .xml файлів

У каталозі layout містяться xml файли, в яких міститься розмітка для вікон додатку та елементів складних списків.

У каталозі menu міститься файл з розміткою головного меню.

У каталозі values містяться додаткові данні:

- Colors: збереження кольорів за замовчуванням
- Refs: посилання
- Strings: текст типу string
- Themes: теми за замовчуванням

За допомогою мови xml можна не лише добавляти елементи інтерфейсу, а розміщувати їх на екрані, редагувати параметри елементів, прикріплюють елементи один до одного, встановлювати відступи для елементів.

Також .xml має можливість додавати контейнери які можна налаштувати для більш зручного розташування елементів на екрані.

Розглянемо елементи інтерфейсу, які були використані в додатку:

Button – кнопка. Основні параметри: id, text. Може приймати параметр onClick() в який додається дія, яку потрібно виконати при натисканні на кнопку.

ImageButton – аналог Button. Різниця полягає в тому, що зовнішній вид кнопки можливо задати зображенням.

TextView – поле для виводу тексту. Можна встановити текст за замовчуванням, можна встановити в ході виконання програми.

EditText – поле для вводу або редагування тексту. Можна встановити текст за замовчуванням.

CheckBox – кнопка, яка має 2 стани: вибрано чи не вибрано.

Switch – аналог CheckBox, має другий вигляд.

TimePicker – віджет, який дає можливість вибрати час на інтерактивному годиннику. Щоб отримати вибраний час потрібно визвати методи getHour та getMinute.

DatePicker – віджет, який дає можливість вибрати дату на календарю.

Щоб отримати вибрану дату, потрібно визвати методи getYear(), getMonth та getDay().

## 5.4 Створення Activity

Activity – Java клас, який наслідує клас AppCompatActivity. Зазвичай містить методи для взаємодії з інтерфейсом: реагування на натискання кнопки, зміну тексту тощо.

До кожного Activity прив'язаний свій .xml файл. За допомогою функції findViewById() клас Activity має можливість дістати елемент інтерфейсу з .xml файлу за id, який задається у кожного елемента.

Після того як будь-який клас Activity отримує елемент інтерфейсу, в ньому викликаються методи для редагування або відгуку на поведінку елемента.

Додаток містить 5 головних класів Activity: MainActivity, BusinessActivity, ScheduleActivity, SubjectActivity, SettingActivity на яких описана поведінка елементів таблиць БД.

Перехід між головними вікнами виконується за допомогою меню.

Також наявні побічні Activity для редагування елементів БД та встановлення поведінки для списків.

Activity класи зберігаються у каталозі activities

На рисунку 5.5 зображено каталог з класами Activity.

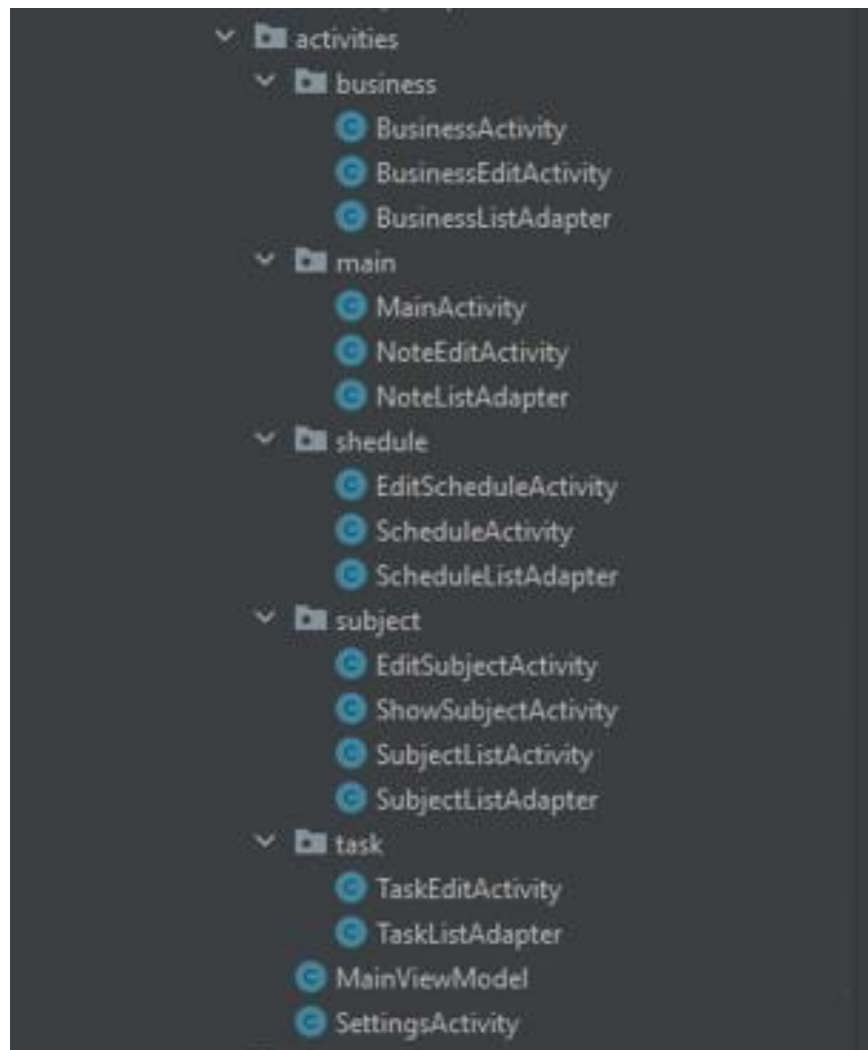


Рисунок 5.5 - Структура проекту (Activity)

Для кожного елементу було створено окремий каталог.

## 5.5 Створення алгоритму для імпорту і експорту інформації

Для імпорту і експорту інформації був створений клас Parser. Щоб експортувати данні, до класу передаються таблиці БД. За допомогою метода `makeOutputParserTextTask()` перетворює данні з таблиць у масив символів та завантажує його в буфер обміну пристрою.

Приклад експортованого тексту:

<Task>\ntaskText\ntaskSubjectName\ntaskYear\ntaskMonth\ntaskDay\ntaskSIId>

Для імпорту даних клас навпаки отримує масив символів та розділяє його на лексеми за допомогою метода `parseInputText()`. Після цього за допомогою метода `readKey()` клас зчитує лексему та створює відповідні запити до БД.

## **5.6 Створення алгоритму для автоматичного додавання списку предметів та розкладу**

Автоматичне додавання розкладу та списку предметів реалізовано шляхом отримання інформації через API в файлі з розширенням `Json`. Данні надає сервіс `api.rozklad.org.ua`.

Було використано бібліотеку `Retrofit` для створення HTTP запитів та бібліотека `GSON` для автоматичного парсингу отриманого файлу.

Щоб отримати данні з серверу, користувач повинен ввести назву своєї групи до відповідного поля.

Для додавання списку предметів виконується наступний запит до API:

GET `http://api.rozklad.org.ua/v2/groups/{group_name}/lessons`

З поданої інформації система відбирає назву предмета, ПІБ викладача та додає всі предмети до БД додатку.

Для додавання розкладу виконується наступний запит до API:

GET `http://api.rozklad.org.ua/v2/groups/{group_name}/timetable`

З поданої інформації система відбирає назву предмета, день тижня, час початку пари та тип заняття. Після парсингу відповідні об'єкти додаються до БД.

## **ВИСНОВКИ ДО РОЗДІЛУ 5**

У цьому розділі було описано програмну реалізацію додатку, а саме:

- Створення БД
- Створення інтерфейсу
- Зв'язування програмної частини з інтерфейсом
- Створення парсеру
- Створення зв'язку з API

Була описана структура БД та методи, за допомогою яких робляться звернення до БД.

Було описано методи, за допомогою яких інформація виводиться на екран додатку та методи, які дозволяють інтерфейсу визивати окремі функції.

## 6 ІНСТРУКЦІЯ РОБОТИ З ПРОГРАМОЮ

Для початку роботи потрібно завантажити додаток на свій мобільний телефон. Після цього користувач повинен встановити його. Інструкції що до встановлення додатку можуть відрізнятися відповідно до пристрою.

Після успішного інсталювання іконка додатку з'явиться на робочому столі смартфона.

Щоб відкрити додаток, користувач повинен натиснути на відповідний значок з підписом “Органайзер”

Додаток має 5 головних вікон:

- Вікно з нотатками
- Вікно зі списком справ
- Вікно зі списком предметів
- Вікно з розкладом занять
- Вікно з налаштуваннями

Вікном за замовчуванням виступає вікно з нотатками, яке має вигляд, зображений на рисунку 6.1.

Вікно за замовчуванням це перше, що побачить користувач, коли зайде до додатку.

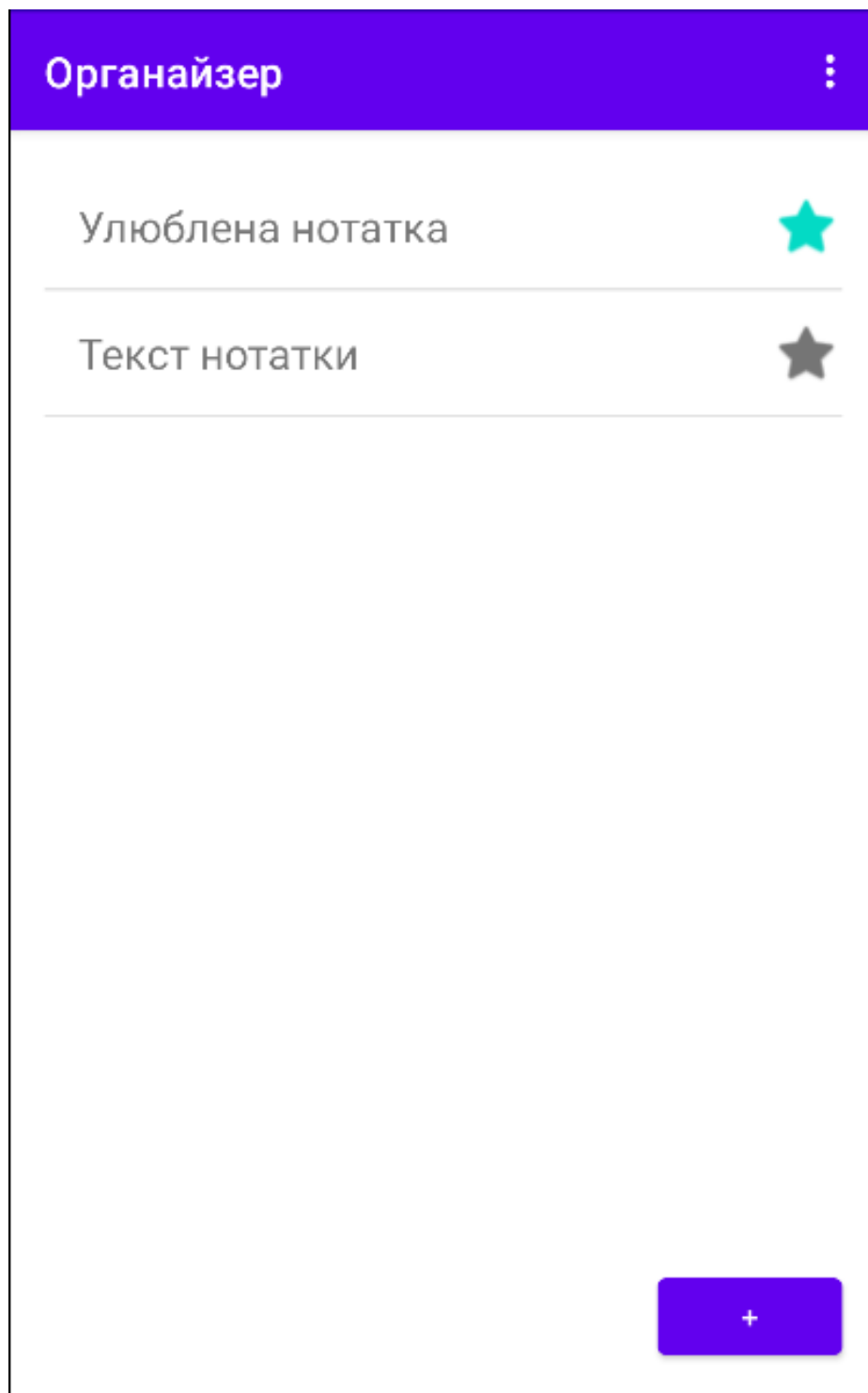


Рисунок 6.1 – Вікно з нотатками

В цьому вікні відображаються нотатки. Нотатки помічені зіркою завжди знаходяться зверху списку. Щоб помітити нотатку потрібно натиснути на зірку.



Після натискання список буде автоматично оновлено. Щоб створити нотатку, потрібно натиснути на кнопку “+”. Щоб редагувати нотатку потрібно натиснути на її текст. Обидві дії відкриють додаткове вікно для редагування нотатки, яке має вигляд, зображений на рисунку 6.2.



Органайзер

Введіть текст нотатки

ВИДАЛИТИ

ЗБЕРЕГТИ

Рисунок 6.2 – Редагування нотатки

В вікні редагування нотатки можна редагувати її текст. Щоб видалити або зберегти нотатку треба натиснути на відповідні кнопки. Після цього вікно автоматично повернеться до списку нотаток.

Для переходу між основними вікнами використовується меню, яке знаходиться в правому-верхньому куту екрану, та має вигляд, зображений на рисунку 6.3.

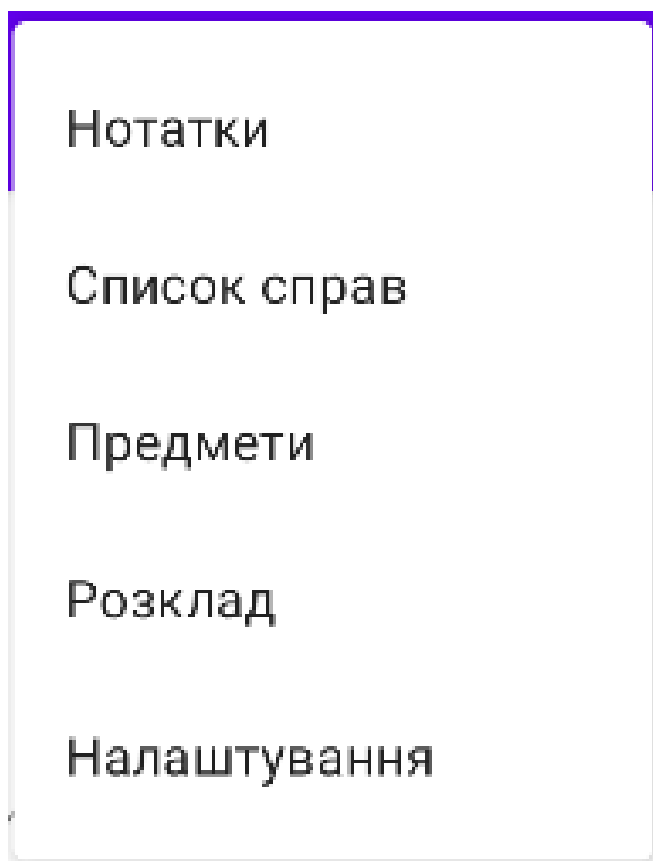


Рисунок 6.3 – Головне меню

Щоб перейти до будь-якого головного вікна треба натиснути на його назву. Вікно зі списком справ має вигляд, зображений на рисунку 6.4.

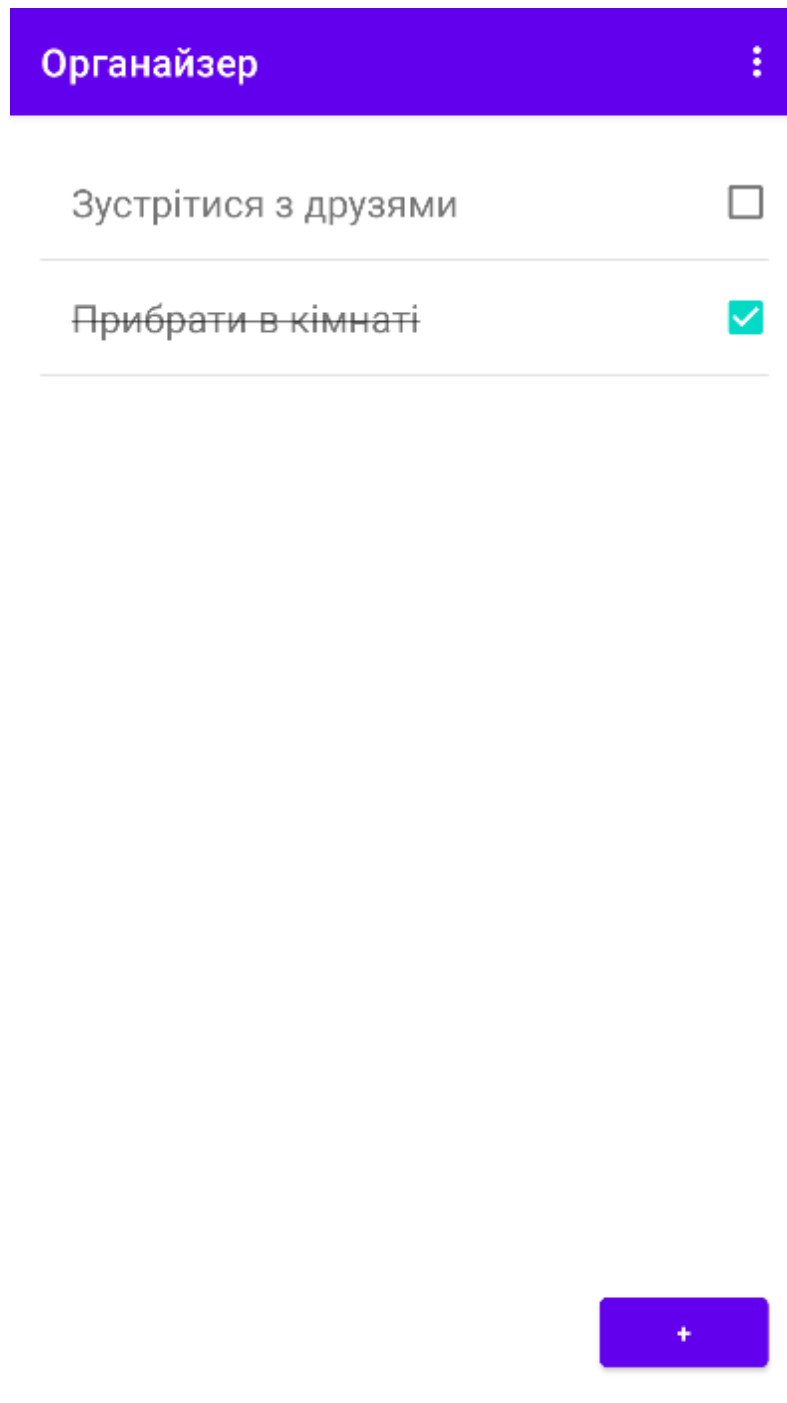


Рисунок 6.4 – Список справ

В цьому вікні відображаються список справ. Щоб помітити справу як виконану, необхідно натиснути на кнопку, яка знаходиться праворуч від тексту справи. Текст виконаних справ буде перечерпнутим, а сама справа буде знаходитись внизу списку справ. Щоб створити справу треба натиснути на кнопку

“+”. Щоб редагувати справу треба натиснути на її текст. Після цього відкриється вікно редагування справи, яке має вигляд вікна, зображеного на рисунку 6.5.



Органайзер

Введіть текст нотатки

ВИДАЛИТИ

ЗБЕРЕГТИ

Рисунок 6.5 – Редагування справи

Вікно зі списком предметів має вигляд, зображений на рисунку 6.6

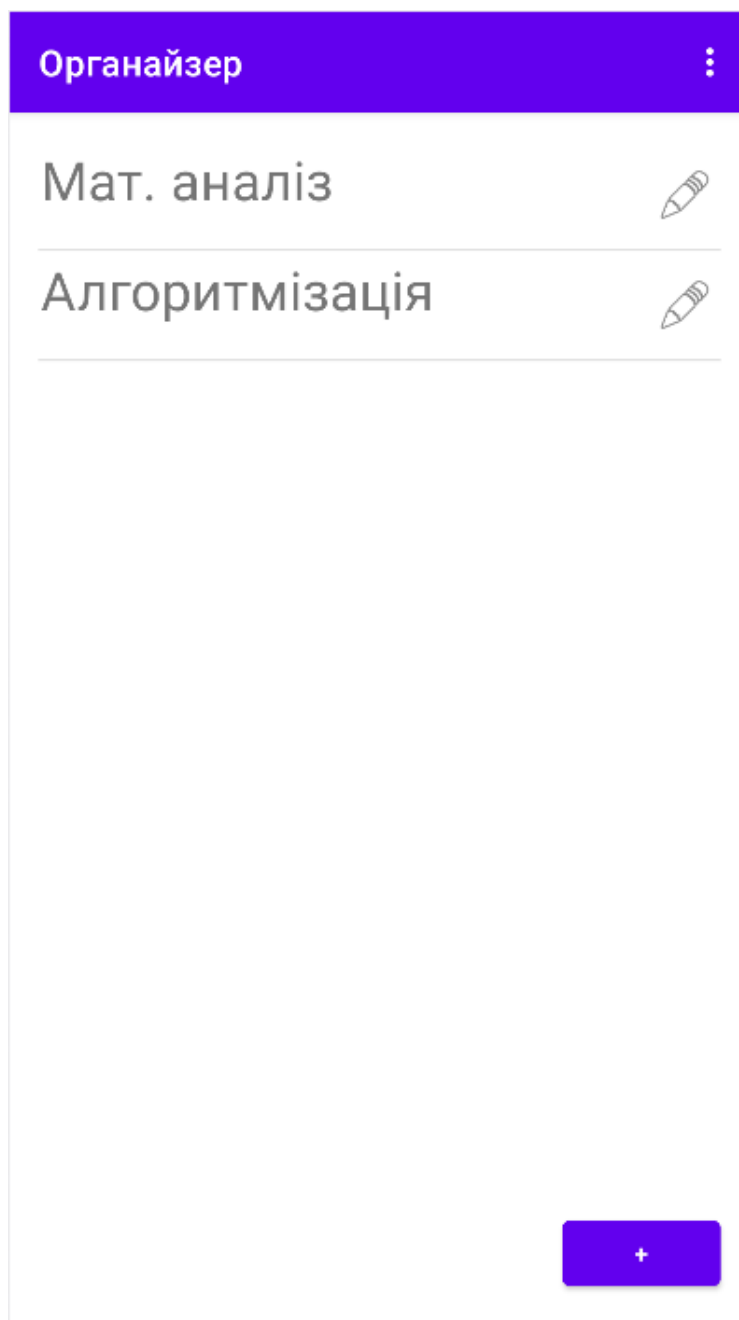


Рисунок 6.6 – Список предметів

В цьому вікні зображується список предметів. Щоб додати предмет треба натиснути на кнопку “+”. Щоб редагувати – на кнопку з зображенням олівця.

Після цього відкриється вікно для редагування предмета, яке має вигляд, зображений на рисунку 6.7.

Органайзер

Предмет:

Мат. аналіз

Викладач

Піфагор

Посилання на пару

<https://www.zoom.link>

Кредит:

☐ Залік

☒ Екзамен

ВИДАЛИТИ

ЗБЕРЕГТИ

Рисунок 6.7 – Редагування предмету

Користувач може ввести назву предмета, ПІБ викладача, тип кредиту(екзамен або залік) та посилання на дистанційну пару. Щоб зберегти або видалити предмет треба натиснути на відповідні кнопки.

Щоб переглянути предмет треба натиснути на його назву у списку предметів. Після цього відкриється вікно з інформацією про предмет, яке має вигляд, зображений на рисунку 6.8.

Органайзер

Алгоритмізація

Білл Гейтс

Екзамен

Завдання

Лабораторна робота 1

29.4.2022

Лабораторна робота 2

2.5.2022

ДОДАТИ ЗАВДАННЯ

Пари:

Алгоритмізація

пн 8:30

Алгоритмізація

пн 10:10

Алгоритмізація

чт 12:10

ДОДАТИ ПАРУ У РОЗКЛАД

Рисунок 6.8 – Огляд предмету

В цьому вікні зображена інформація про предмет: назва, ПІБ викладача, тип кредиту, список завдань та список пар.

Щоб додати завдання до предмета потрібно натиснути на відповідну кнопку. Щоб редагувати завдання потрібно натиснути на його текст. Після цього відкриється вікно редагування завдання, яке має вигляд, зображений на рисунку 6.9.

Органайзер

Введіть завдання

Дата здачі завдання:

2022

Sun, May 29

<

May 2022

>

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

ВИДАЛИТИ

>

ЗБЕРЕГТИ

Рисунок 6.9 – Редагування завдання

В цьому вікні користувач може ввести текст завдання та дату здачі завдання. Щоб вибрати дату потрібно натиснути на відповідну дату у віджеті календаря. Щоб зберегти або видалити завдання потрібно натиснути на відповідні кнопки.

Щоб додати пару до предмета, потрібно натиснути на відповідну кнопку у вікні з інформацією про предмет. Після цього відкриється вікно редагування завдання, яке має вигляд, зображений на рисунку 6.10.



Органайзер

☐ ПН ☐ ВТ ☐ СР ☐ ЧТ ☐ ПТ ☐ СБ

23

01

00

:

02

01

03

Посилання на пару

☐ Лек.  
☐ Прак.  
☐ Лаб.

ВИДАЛИТИ

ЗБЕРЕГТИ

Рисунок 6.10 – Редагування розкладу

В цьому вікні користувач може вибрати день тижня, в який проходить пара, тип пари та час початку пари на віджеті з вибором часу. Щоб зберегти або видалити пару потрібно натиснути на відповідні кнопки.

Після збереження пари, вона автоматично додається до вікна з розкладом, яке має вигляд, зображений на рисунку 6.11.

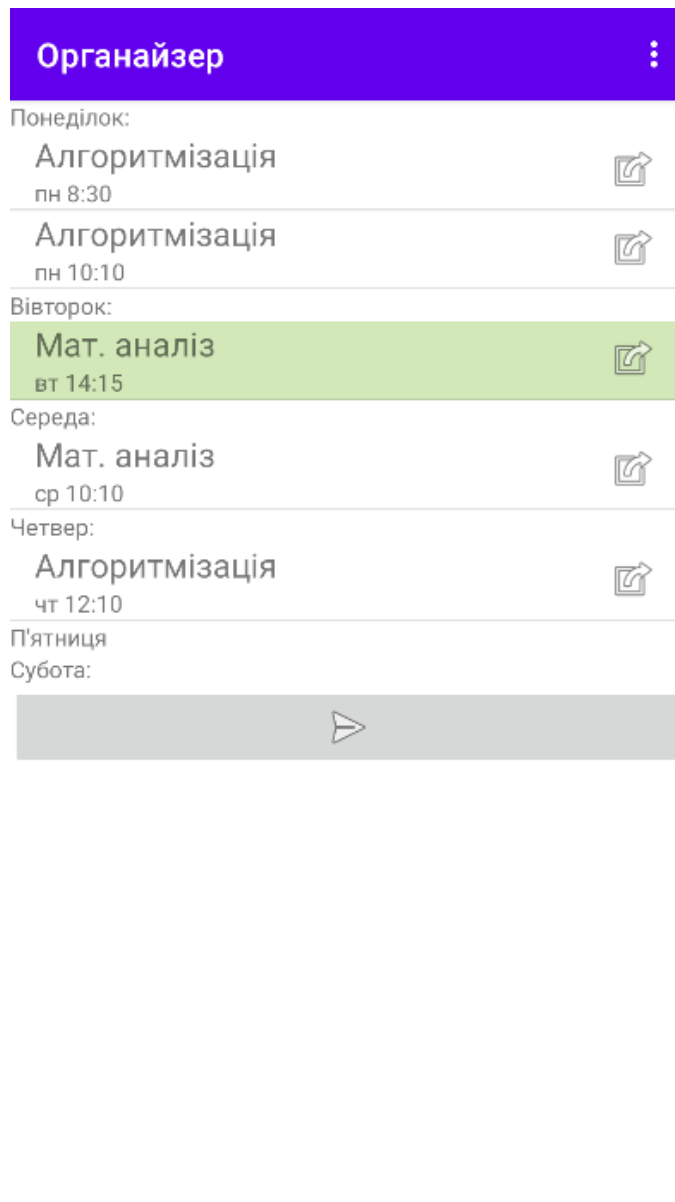


Рисунок 6.11 – Розклад

Поточний день тижня в розкладі відмічений зеленим кольором. Щоб скопіювати посилання на онлайн пару потрібно натиснути на кнопку, яка знаходиться праворуч від назви предмета.

Вікно налаштувань має вигляд, зображений на рисунку 6.12

Сповіщення про наближення  
дедлайну завдання ☐

Сповіщення про пару ☒

Сповіщення про початок пари за:

Введіть кількість хвилин

Введіть групу

ДОДАТИ РОЗКЛАД

ДОДАТИ ПРЕДМЕТИ

СКОПІЮВАТИ БД

Вставте текст для парсингу

ДОДАТИ

Рисунок 6.12 – Налаштування

Користувач може увімкнути або вимкнути сповіщення про наступну пару та про наближення дати здачі завдання. Також є можливість вибрати час у хвилинах, за який прийде сповіщення до початку пари. За замовчуванням він встановлений як 15 хвилин.

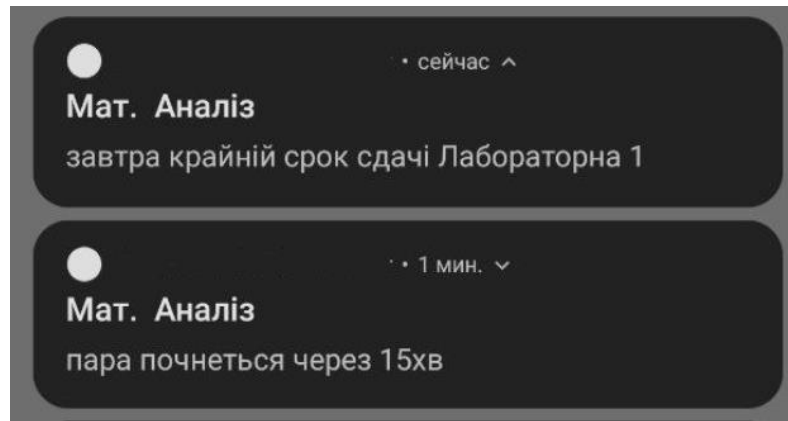


Рисунок 6.13 – Зразки повідомлень

Додаток має алгоритм, який дозволяє копіювати завдання, список пар або всі данні з додатку. Щоб скопіювати завдання або список пар треба натиснути на кнопку з зображенням паперового літака у відповідних вікнах.

Щоб скопіювати всі данні треба натиснути на кнопку “скопіювати БД” у вікні налаштувань. Додаток створить текст, який збережеться в буфер обміну смартфона. Після цього користувач зможе передати його будь яким зручним для нього способом. Якщо вставити цей текст в поле в вікні налаштувань та натиснути кнопку “додати” вся скопійована інформація буде додана до відповідних списків додатку та їх полів. Копіюється не лише текст пар або завдань а вся інформація, включаючи дату, час та інші данні. Таким чином користувачі можуть швидко ділитися інформацією про завдання, списком пар або зберігати всю інформацію в окремому носії. Наприклад один студент може заповнити список пар та поділитися їм зі своїми одногруппниками.

Щоб автоматично додати розклад або список предметів, користувач повинен ввести назву своєї групи в відповідне поле та натиснути на кнопку “додати розклад” або “додати предмети”. Для даної операції користувач повинен мати інтернет підключення.

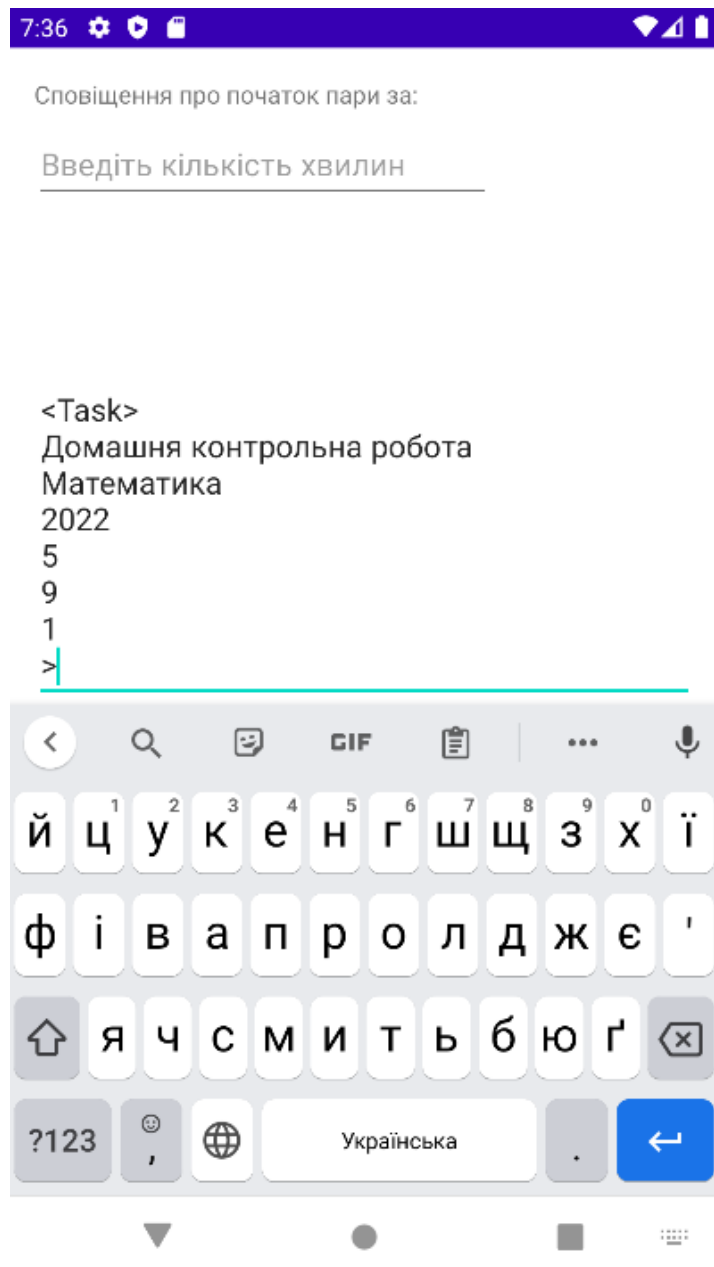


Рисунок 6.14 – Приклад додавання інформації

На рисунку 6.14 зображено приклад тексту з інформацією.

## **ВИСНОВКИ ДО РОЗДІЛУ 6**

У цьому розділі було показано як користувач має взаємодіяти з додатком.  
Показана інструкція до всіх властивостей додатку:

- створення нотаток
- редагування нотаток
- створення справ
- редагування справ
- створення предметів
- редагування предметів
- створення завдань
- редагування завдань
- створення розкладу
- редагування розкладу
- створення тексту для передачі інформації
- вставка тексту для додавання інформації

## ВИСНОВКИ

Під час виконання бакалаврської роботи було поставлено задачу створити мобільний додаток, який буде допомагати студенту в навчанні. Було проаналізовано вже створені організатори, та враховуючи їх недоліки було створено власний додаток.

Ознайомившись з літературою та з середовищем розробки були обрані інструменти для створення додатку.

Протягом написання роботи були здобуті навички побудови моделі мобільних додатків, побудови моделі баз даних, роботи з API, створення дизайну та безпосередньо самого інтерфейсу та програмної частини додатку.

Для ознайомлення з додатком було створено розділ з детальною інструкцією по його використанню.

В результаті маємо готовий додаток, в якому користувач може записувати свої нотатки, планувати справи та план навчання, отримувати нагадування про початок занять та термін здачі роботи та передавати цю інформацію.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wikipedia [Електронний ресурс] – Режим доступу:  
<https://uk.wikipedia.org/wiki>
2. ELAKP [Електронний ресурс] – Режим доступу: <https://ela.kpi.ua/>
3. Документація Android: веб-сайт.URL <https://developer.android.com>
4. Kathy Sierra, Bert Bates Head First Java, Sebastopol, USA : O'Reilly Media, 2022. 720 с.
5. Alan Beaulieu Learning SQL: Master SQL Fundamentals, Sebastopol, USA : O'Reilly Media, 2009. 338 с.
6. Brett McLaughlin, Justin Edelson Java and XML, Sebastopol, USA: O'Reilly Media 2006. 544 с.
7. Технічна документація Android: веб-сайт. URL:  
<https://developer.android.com/docs>
8. KPI.ua [Електронний ресурс] – Режим доступу: <https://kpi.ua/>
9. Stackoverflow [Електронний ресурс] – Режим доступу:  
<https://stackoverflow.com/>
10. Документація Room: веб-сайт.URL  
<https://developer.android.com/training/data-storage/room>
11. Документація Java: веб-сайт. URL <https://docs.oracle.com/en/java/>
12. Документація XML: веб-сайт. URL  
<https://support.crowdin.com/ru/enterprise/file-formats/android-xml/>



# ДОДАТОК А

Мобільний додаток для організації навчання студента

Текст програми

УКР.НТУУ“КПІ ім. Ігоря Сікорського”.ТР81\_\_\_\_22Б 12-1

Аркушів 7

Київ – 2022

```
package com.example.diplom.activities.main;

//Підключення бібліотек

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;


import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.NotificationCompat;

import androidx.core.app.NotificationManagerCompat;

import androidx.lifecycle.Observer;

import androidx.lifecycle.ViewModelProviders;

import androidx.recyclerview.widget.DividerItemDecoration;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;


import com.example.diplom.R;

import com.example.diplom.activities.MainViewModel;

import com.example.diplom.activities.SettingsActivity;
```

```

import com.example.diplom.activities.business.BusinessActivity;

import com.example.diplom.activities.business.BusinessEditActivity;

import com.example.diplom.activities.shedule.ScheduleActivity;

import com.example.diplom.activities.subject.EditSubjectActivity;

import com.example.diplom.activities.subject.SubjectListActivity;

import com.example.diplom.database.entities.Note;


import java.util.List;


public class MainActivity extends AppCompatActivity {


    private RecyclerView recyclerView;

    private Button newNote;


    // створення меню

    @Override

    public boolean onCreateOptionsMenu(Menu menu) {


        getMenuInflater().inflate(R.menu.main_menu, menu);

        return true;

    }

```

```

public boolean onOptionsItemSelected(MenuItem item) {

    int id = item.getItemId();

    switch(id){

        case R.id.NOTES:

            return true;

        case R.id.BUSINESS:

            BusinessActivity.startBusinessActivity(MainActivity.this);

            return true;

        case R.id.SUBJECTS:

            SubjectListActivity.startSubjectListActivity(MainActivity.this);

            return true;

        case R.id.SHEDULE:

            ScheduleActivity.startScheduleActivity(MainActivity.this);

            return true;

        case R.id.SET:

            SettingsActivity.startSettingsActivity(MainActivity.this);

            return true;

    }

    return super.onOptionsItemSelected(item);

}

// Початок існування вікна

@Override

```

```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);


    newNote = findViewById(R.id.buttonCreateNote);
    recyclerView = findViewById(R.id.recyclerViewNoteList);


    LinearLayoutManager layoutManager = new
LinearLayoutManager(this, RecyclerView.VERTICAL, false);

    recyclerView.setLayoutManager(layoutManager);

    recyclerView.addItemDecoration(new DividerItemDecoration(this,
DividerItemDecoration.VERTICAL));


    NoteListAdapter noteListAdapter = new NoteListAdapter();
    recyclerView.setAdapter(noteListAdapter);


    newNote.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            NoteEditActivity.startNoteEditActivity(MainActivity.this, null);

```

```
    }  
});
```

```
        MainViewModel mainViewModel =  
ViewModelProviders.of(this).get(MainViewModel.class);  
  
        mainViewModel.getNoteLiveData().observe(this, new  
Observer<List<Note>>() {  
  
            @Override  
  
            public void onChanged(List<Note> notes) {  
  
                noteListAdapter.updateNoteList(notes);  
  
            }  
  
        });  
    }  
  
//ф-ція запуску головного вікна  
  
    public static void startMainActivity(Activity activity){  
  
        Intent intent = new Intent(activity, MainActivity.class);  
  
        activity.startActivity(intent);  
  
    }  
  
}
```

```
////////// activity_main.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
  
    xmlns:tools="http://schemas.android.com/tools"  
  
    android:layout_width="match_parent"  
  
    android:layout_height="match_parent"  
  
    tools:context=".activities.main.MainActivity">
```

```
// розмітка кнопки
```

```
<Button  
  
    android:id="@+id/buttonCreateNote"  
  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
  
    android:layout_marginEnd="16dp"  
  
    android:layout_marginBottom="16dp"  
  
    android:text="+"  
  
    app:layout_constraintBottom_toBottomOf="parent"  
  
    app:layout_constraintEnd_toEndOf="parent" />
```

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recyclerViewNoteList"  
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="16dp"  
    android:layout_marginTop="16dp"  
    android:layout_marginEnd="16dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.0" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```