Міністерство освіти і науки України

Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»

Кафедра «Комп'ютерні інформаційні технології»

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему: «Мобільний додаток для перегляду розкладу занять університету»

за освітньою програмою: «Інженерія програмного забезпечення»

зі спеціальності: «121 Інженерія програмного забезпечення»

Виконав: студент групи «ПЗ20130»

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | /Владислав ЗАБОЛОТНИЙ/ |

(підпис студента) (Ім’я ПРІЗВИЩЕ)

|  |  |  |  |
| --- | --- | --- | --- |
| Керівник: |  |  | /доц. Олександр ЖЕВАГО/ |

(підпис) (посада, Ім’я ПРІЗВИЩЕ)

|  |  |  |  |
| --- | --- | --- | --- |
| Нормоконтролер: |  |  | /доц. Світлана ВОЛКОВА/ |

(підпис) (посада, Ім’я ПРІЗВИЩЕ)

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

(підпис)

Дніпро – 2023 рік

Ministry of Education and Science of Ukraine

Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»

Department «Computer information technology»

Explanatory Note

to Bachelor’s Thesis

on the topic: «Development of a mobile application for viewing the university class schedule»

according to educational curriculum «Software engineering»

in the Speciality: «121 Software engineering»

|  |  |  |
| --- | --- | --- |
| Done by the student of the group PZ20130: |  | /Vladyslav ZABOLOTNYI/ |
| Scientific Supervisor: |  | /Oleksandr ZHEVAGO/ |
| Normative controller: |  | /Svitlana VOLKOVA/ |

Dnipro – 2023

Міністерство освіти і науки України

Український державний університет науки і технологій

Факультет: Факультет «Комп'ютерні технології і системи»

Кафедра: «Комп'ютерні інформаційні технології»

Рівень вищої освіти: бакалавр

Освітня програма: «Інженерія програмного забезпечення»

Спеціальність: «121 Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри КІТ

/Вадим ГОРЯЧКІН/

(підпис)

Дата

**ЗАВДАННЯ**

на кваліфікаційну роботу бакалавра

студенту Заболотному Владиславу Олександровичу

1. Тема роботи: «Мобільний додаток для перегляду розкладу занять університету»

Керівник роботи: Жеваго Олександр Олександрович, доцент

затверджені наказом № 1209 ст від 07.12.2022

|  |  |  |
| --- | --- | --- |
| 2. Строк подання студентом роботи: | | \_\_.\_\_.202\_ р. |
| 3. Вихідні дані до роботи: |  | |
|  | | |
| 4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати): | | |
| Вступ | | |
| Збір та аналіз вимог до мобільного додатку | | |
| Проектування мобільного додатку | | |
| Проектування прототипу мобільного додатку | | |
| Розробка прототипу мобільного додатку | | |
| Розробка мобільного додатку | | |
| Тестування мобільного додатку | | |
| Загальні висновки | | |
| 5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень): | | |
| Презентація | | |
| Відео роботи програми | | |

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |
| --- | --- | --- |
| Стадії розробки | Етапи розробки | Терміни виконання |
| 1.Технічне завдання (ТЗ) | Постановка задачі | 13.02.23 – 18.02.23 |
| Огляд літератури та аналіз аналогів | 19.02.23 – 22.02.23 |
| Розробка структур вхідних і вихідних даних | 23.02.23 – 04.03.23 |
| Визначення вимог до програми. Вибір та обґрунтування мови програмування | 05.03.23 – 08.03.23 |
| Узгодження та затвердження ТЗ | 09.03.23 – 11.03.23 |
| 2. Робочий проект | Розробка та програмування логіки програми | 12.03.23 – 02.05.23 |
| Розробка і реалізація інтерфейсу користувача | 03.05.23 – 19.05.23 |
| Відлагодження програми | 20.05.23 – 09.06.23 |
| Розробка, узгодження та затвердження програмної документації | 10.06.23 – 14.06.23 |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  |  | Владислав ЗАБОЛОТНИЙ |

(підпис) (Ім’я ПРІЗВИЩЕ)

|  |  |  |  |
| --- | --- | --- | --- |
| Керівник роботи |  |  | доц. Олександр ЖЕВАГО |

(підпис) (Ім’я ПРІЗВИЩЕ)

**РЕФЕРАТ**

**ЗМІСТ**

[**ВСТУП** 7](#_Toc137127319)

[**РОЗДІЛ 1. ЗБІР ТА АНАЛІЗ ВИМОГ ДО МОБІЛЬНОГО ДОДАТКУ** 8](#_Toc137127320)

[1.1. Опис та аналіз аналогів 8](#_Toc137127321)

[1.2. Постановка задачі 21](#_Toc137127322)

[**РОЗДІЛ 2. ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ** 22](#_Toc137127323)

[2.1. Задачі проекту 22](#_Toc137127324)

[2.2. Функціональні вимоги 22](#_Toc137127325)

[2.3. Вхідні та вихідні дані 22](#_Toc137127326)

[2.4. Вибір мови програмування 22](#_Toc137127327)

[2.5. Опис компонентно-орієнтованого підходу 22](#_Toc137127328)

[2.6. Проектування екранів прототипу мобільного додатку 22](#_Toc137127329)

[2.7. Проектування інтерфейсу користувача прототипу мобільного додатку 22](#_Toc137127330)

[2.8. Проектування навігації між екранами прототипу мобільного додатку 22](#_Toc137127331)

[2.9. Проектування дизайну прототипу мобільного додатку 22](#_Toc137127332)

[2.10. Розробка екранів прототипу мобільного додатку 22](#_Toc137127333)

[2.11. Розробка інтерфейсу користувача прототипу мобільного додатку 22](#_Toc137127334)

[2.12. Структура компонентів системи 22](#_Toc137127335)

[2.13. Розробка дизайну прототипу мобільного додатку 22](#_Toc137127336)

[2.14. Розробка навігації між екранами прототипу мобільного додатку 22](#_Toc137127337)

[**РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ** 22](#_Toc137127338)

[3.1. Розробка динаміки системи 22](#_Toc137127339)

[3.2. Розробка екранів мобільного додатку 23](#_Toc137127340)

[3.3. Розробка інтерфейсу мобільного додатку 23](#_Toc137127341)

[3.4. Розробка навігації мобільного додатку 23](#_Toc137127342)

[3.5. Розробка дизайну мобільного додатку 23](#_Toc137127343)

[**РОЗДІЛ 4. ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ** 23](#_Toc137127344)

[4.1. Вибір методів тестування 23](#_Toc137127345)

[4.2. Тестування 23](#_Toc137127346)

[**Загальні висновки** 23](#_Toc137127347)

[**Список використаних джерел** 23](#_Toc137127348)

# ВСТУП

Університети відіграють ключову роль у навчанні молоді та формуванні майбутніх професіоналів. Однак, з огляду на зростання числа студентів та складність розкладу занять, студентам дедалі важче відстежувати свої заняття та організовувати свій навчальний процес. Це може призводити до пропусків, невдалих планувань і, врешті-решт, погіршення академічних результатів.

Причиною виникнення розробки є вирішення цього проблемного питання та відсутність адаптивних додатків для нашого університету. Метою дипломного проєкту є розробка мобільного додатку для перегляду розкладу занять університету. Додаток буде надавати студентам та викладачам зручний та легко доступний інструмент для перегляду актуальних розкладів занять та модулів.

Основною метою проєкту є спрощення процесу отримання та оновлення інформації про розклад занять університету, забезпечуючи студентам та викладачам зручний та швидкий доступ до актуальних даних.

Для розробки додатку будуть використані сучасні технології мобільної розробки, що дозволить забезпечити швидку та надійну роботу програмного забезпечення.

Завдяки даному мобільному додатку, студенти та викладачі університету зможуть легко та швидко отримувати актуальну інформацію про свої заняття, уникнути пропусків та незручностей, пов'язаних з некоректним або застарілим розкладом.

Результатом розробки буде функціональний мобільний додаток, який зробить процес отримання та оновлення розкладу занять університету зручним та ефективним для всіх користувачів.

# ЗБІР ТА АНАЛІЗ ВИМОГ ДО МОБІЛЬНОГО ДОДАТКУ

## Опис та аналіз аналогів

Для пошуку аналогів буде використовуватись маркет додатків Google Play Store. Пошук буде відбуватися у розділі освіта за назвами: «Розклад занять», «Рассписание», «Time table».

1. Додаток розроблений Candy Soft Inc «Мій розклад» дата випуску 23 березня 2019 р. Додаток виконує широкий спектр завдань та відображає розклади великої кількості Українських університетів.

Опис функцій:

* вибір навчального закладу зі списку доступних;
* вибір групи або викладача;
* вибір періоду часу відображення розкладу;
* відображення розкладу для груп та викладачів;
* перехід на сайт університету;
* створення заміток.

Опис інтерфейсу:

Мобільний додаток має 4 основні сторінки:

* Меню – на даній сторінці налаштовуються параметри відображення розкладу, а саме вибір університету, групи або викладача, період часу (рис 1.1 - рис 1.4);

|  |  |
| --- | --- |
|  |  |
| Рисунок 1.1 – Меню | Рисунок 1.2 – Вибір навчального закладу |
|  |  |
| Рисунок 1.3 – Вибір викладача | Рисунок 1.4 – Вибір групи |

* Блокнот – для створення заміток (рис 1.5)



Рисунок 1.5 – Блокнот

* Сайт – сторінка відображення сайту обраного університету (рис 1.6).

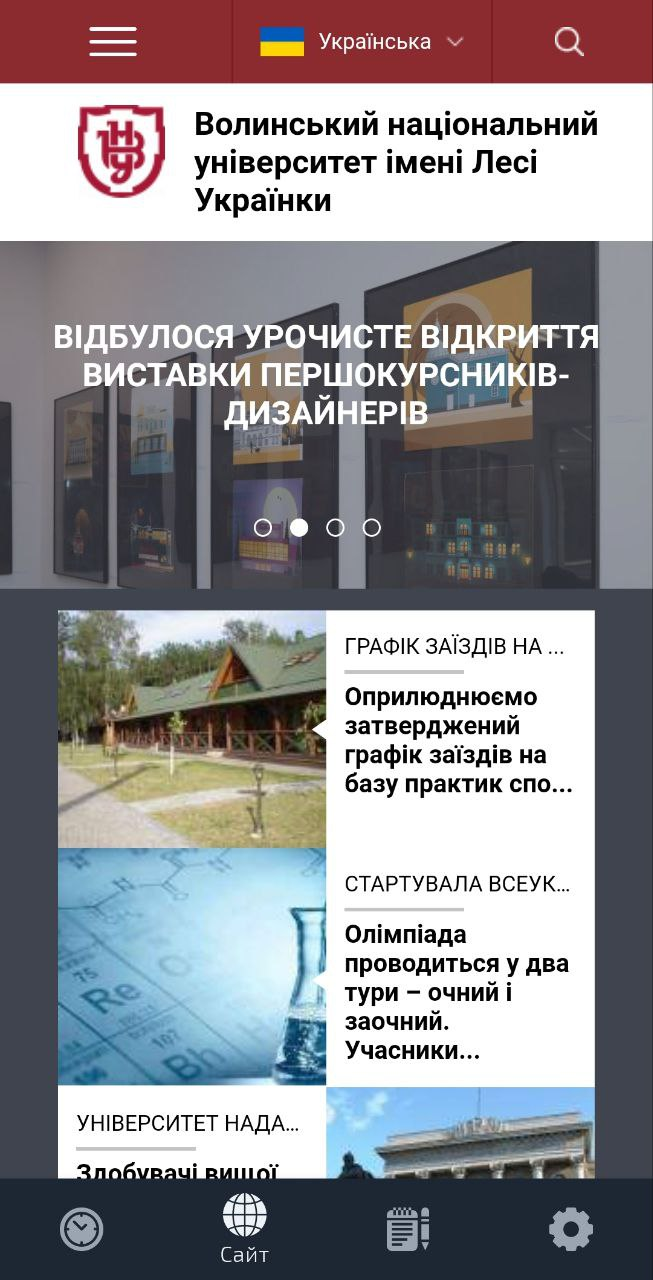


Рисунок 1.6 – Сайт

* Розклад – сторінка відображення розкладу для груп (рис 1.7) та викладачів (рис 1.8).

|  |  |
| --- | --- |
|  |  |
| Рисунок 1.7 – Розклад для групи | Рисунок 1.8 – Розклад для викладача |

Переваги:

* вибір університетів – можна переглядати розклади великої кількості українських університетів;
* зручний інтерфейс;
* перегляд сайту обраного університету не виходячи з додатку;
* можливість створювати замітки у додатку;

Недоліки:

* перегляд тільки поточного розкладу;
* відсутність у списку університетів Українського державного університету науки і технологій;
* неможливість перегляду розкладу сесії.

2. Мобільний додаток від розробника Golden Pie Devs «Розклад КПІ» дата випуску 1 лютого 2015р. Додаток відображає розклад занять Київського політехнічного інституту.

Опис функціоналу:

* перегляд розкладу для групи та викладачів;
* перегляд викладачів до групи з рейтингом;
* перегляд розширеної інформації про заняття;
* можливість згортання додатку до меншого виду.

Опис інтерфейсу:

Програма має наступні сторінки:

* сторінка перегляду розкладу занять для групи, де здійснюється вибір календарного дня (рис 1.9) та відбувається відображення розкладу за обраним днем на тиждень (рис 1.10).

|  |  |
| --- | --- |
|  |  |
| Рисунок 1.9 – Вибір календарного дня | Рисунок 1.10 – Розклад занять для групи |

* сторінка перегляду інформації про заняття – де відображається докладна інформація про заняття, а саме назва, час, тип та розташування з картою (рис 1.11).

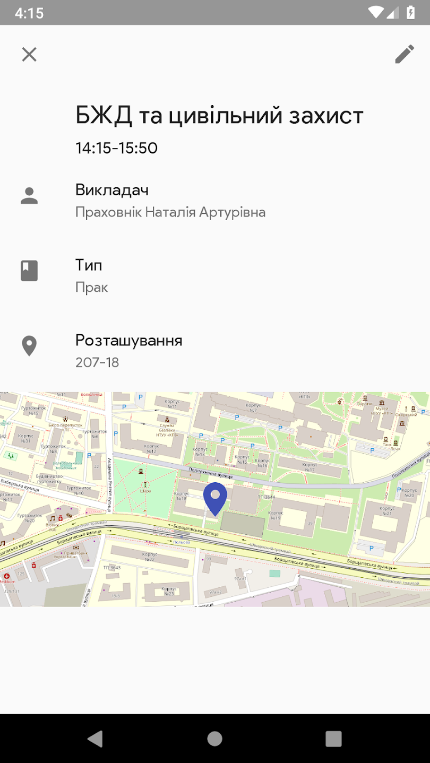


Рисунок 1.11 – Докладна інформація про заняття

* сторінка відображення викладачів з рейтингом для групи (рис 1.12).

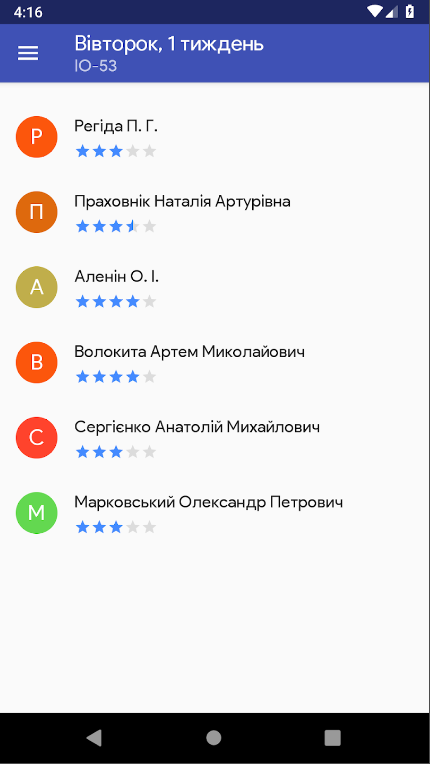


Рисунок 1.12 – Список викладачів для групи

* сторінка розкладу для викладача, на якій відображається розклад занять для обраного викладача (рис 1.13).

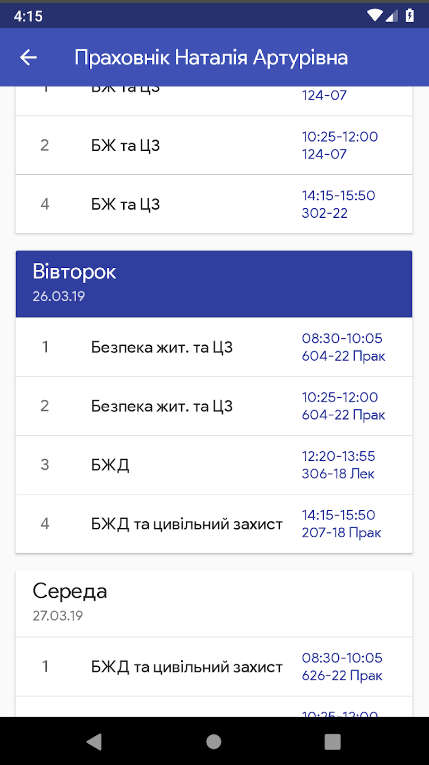


Рисунок 1.13 – Перегляд розкладу для викладача

* розклад занять у згорнутому вигляді (рис 1.14).

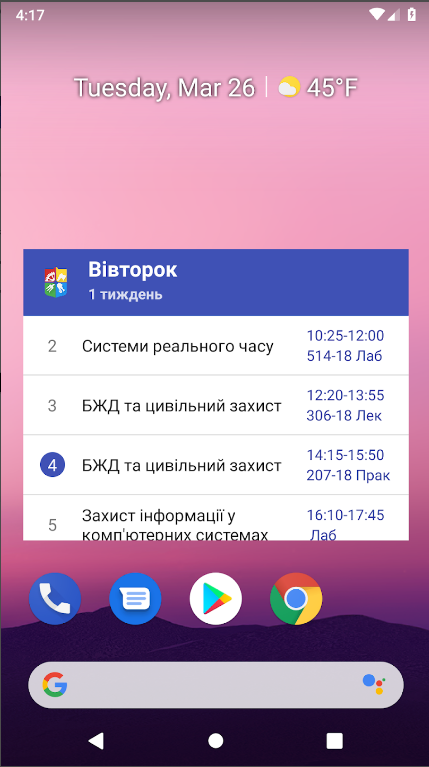


Рисунок – 1.14 Згорнутий вигляд розкладу

Переваги:

* зручний інтерфейс перегляду розкладу;
* рейтинг викладачів;
* карта з місцем знаходженням;

Недоліки:

* перегляд тільки поточного розкладу;
* неможливість перегляду розкладу сесії;
* додаток не підтримується на нових версіях операційних систем.

3. Мобільний додаток від розробника МГТУ ім. Н. Э. Баумана «МГТУ им. Н. Э. Баумана» дата випуску 16 вересня 2019р. Додаток реалізує відображення розкладу занять для студентів та співробітників МГТУ ім. Н. Э. Баумана.

Опис функцій:

* відстежування занять у реальному часі
* пошук розкладів за групою чи викладачем
* перегляд новин університету;

Опис інтерфейсу:

Мобільний додаток має 5 основних сторінок:

* новини – на даній сторінці відображається актуальна інформація університету (рис 1.15)

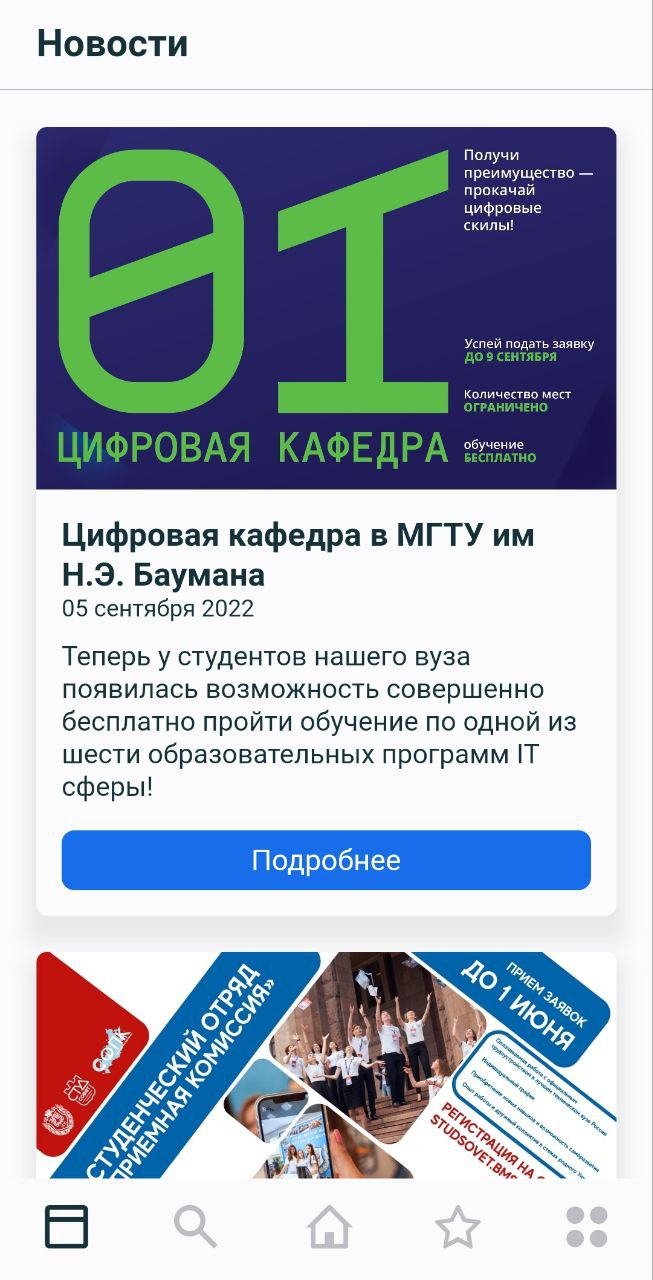


Рисунок 1.15 – Новини університету

* пошук розкладу – пошук відбувається за назвою (рис 1.16) або по факультетам та курсом (рис 1.17) після чого відображається розклад обраної групи зі списку доступних (рис 1.18).

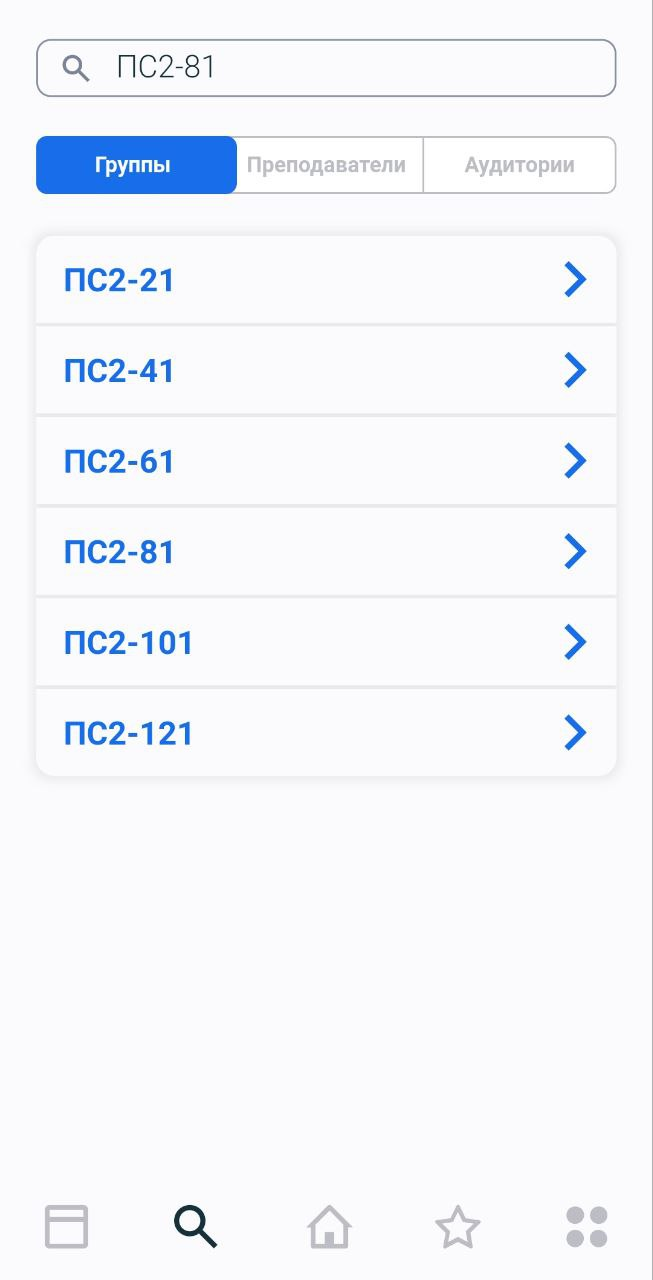


Рисунок 1.17 – Пошук розкладу за назвою групи

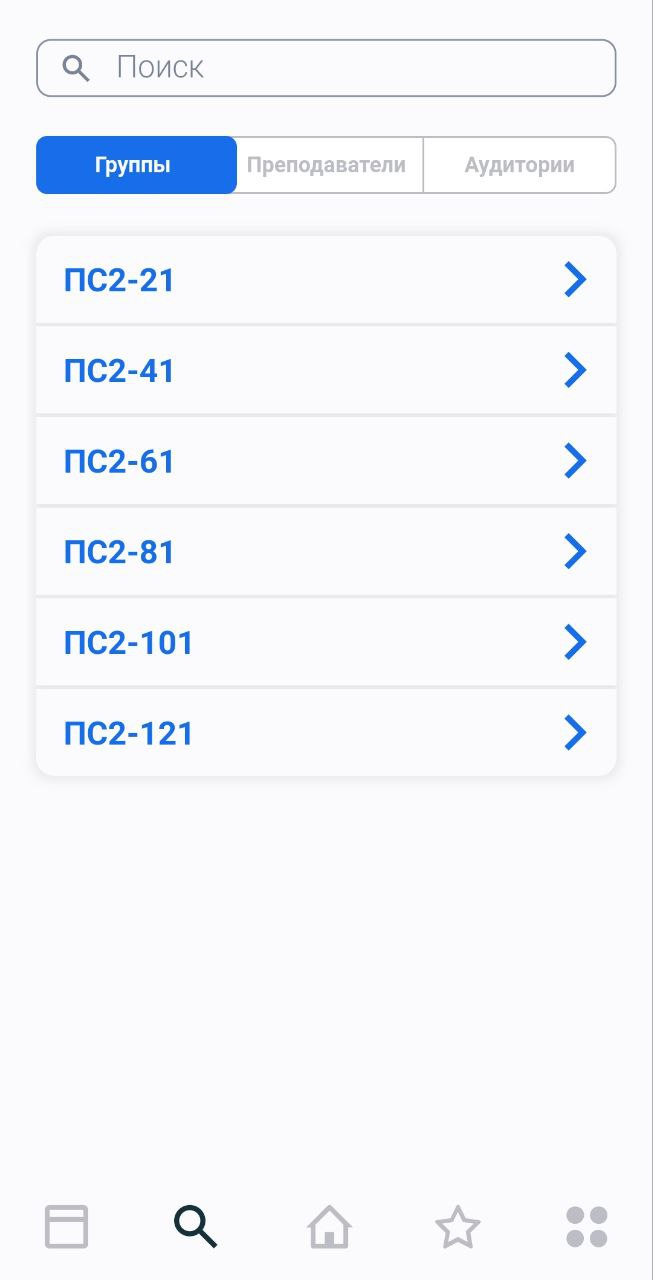
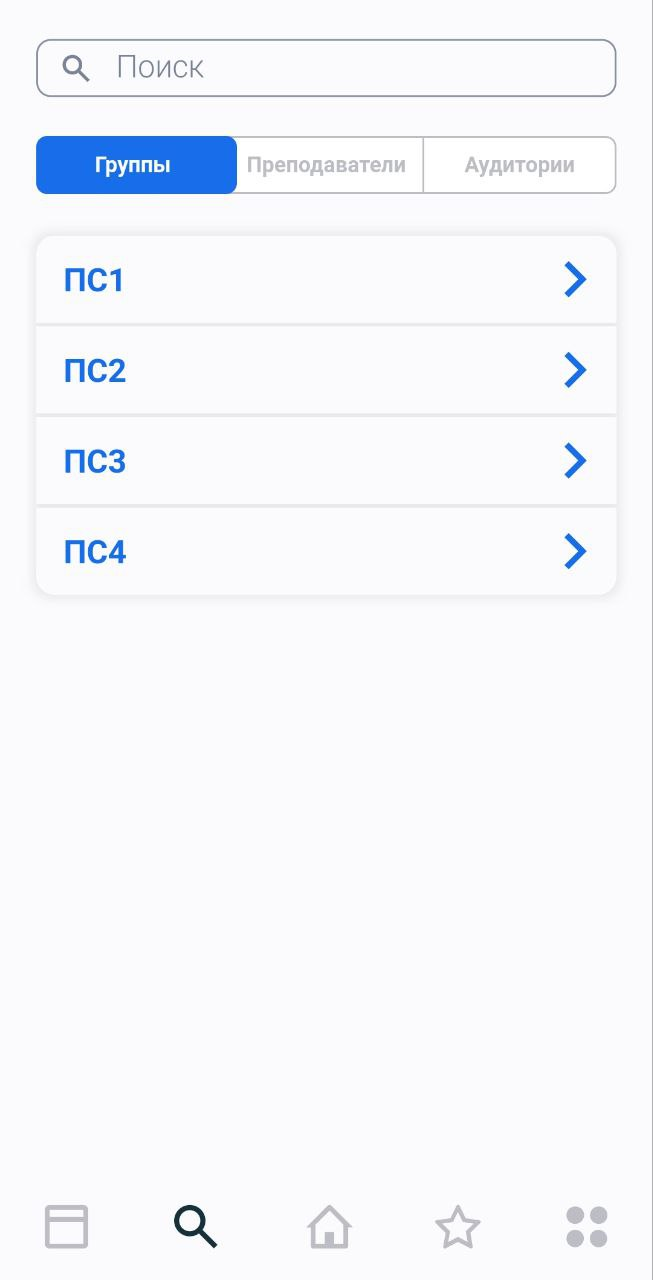
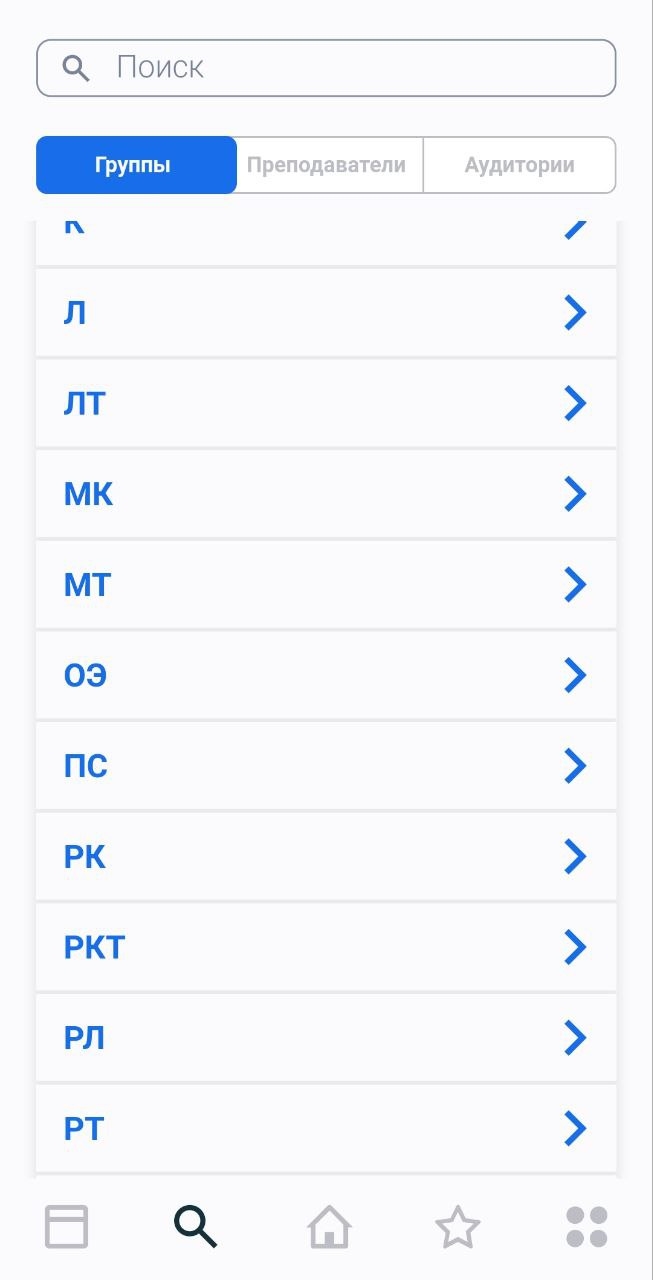


Рисунок 1.17 – Пошук розкладу по факультетом та курсом

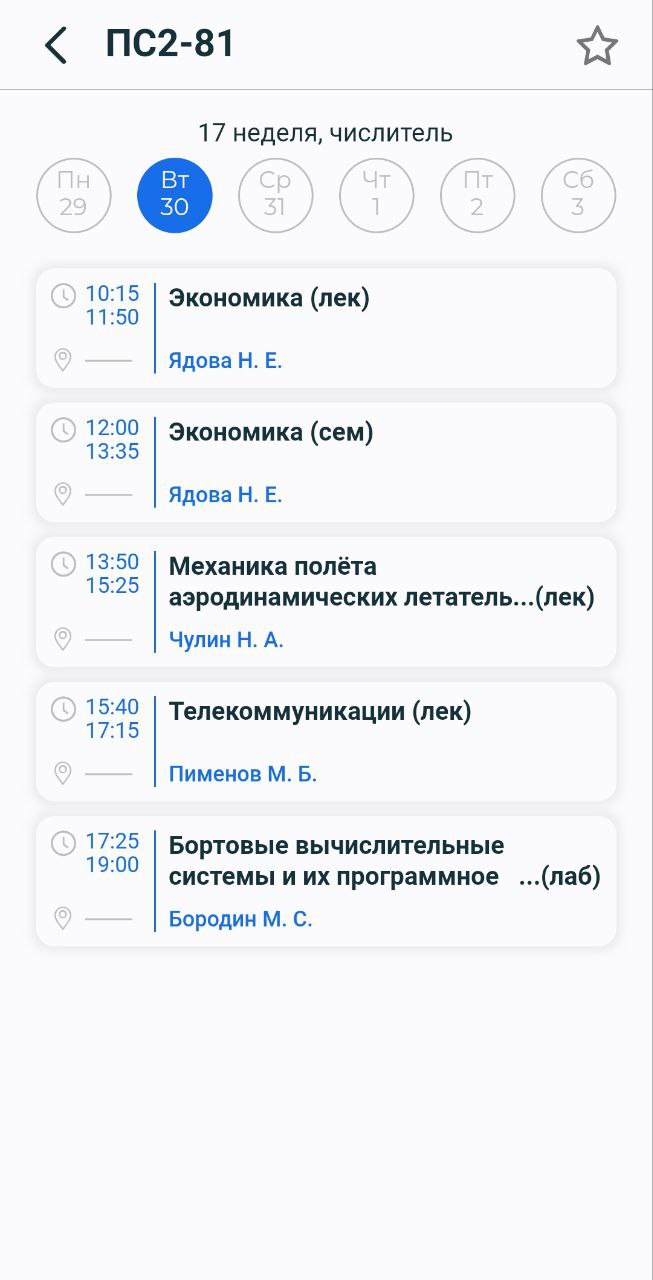


Рисунок 1.18 – Відображення знайденого розкладу

* головна сторінка – на даній сторінці відображається розклад заздалегідь обраної групи у налаштуваннях (рис 1.19).
* закладки груп, де відображається список доданих в закладки груп (рис. 1.20).

|  |  |
| --- | --- |
|  |  |
| Рисунок 1.19 – Відображення знайденого розкладу | Рисунок 1.20 – Вибрані групи |

* налаштування – на цій сторінці можна змінити основну групу, змінити тему додатку та дізнатися інформацію про додаток (рис. 1.21).

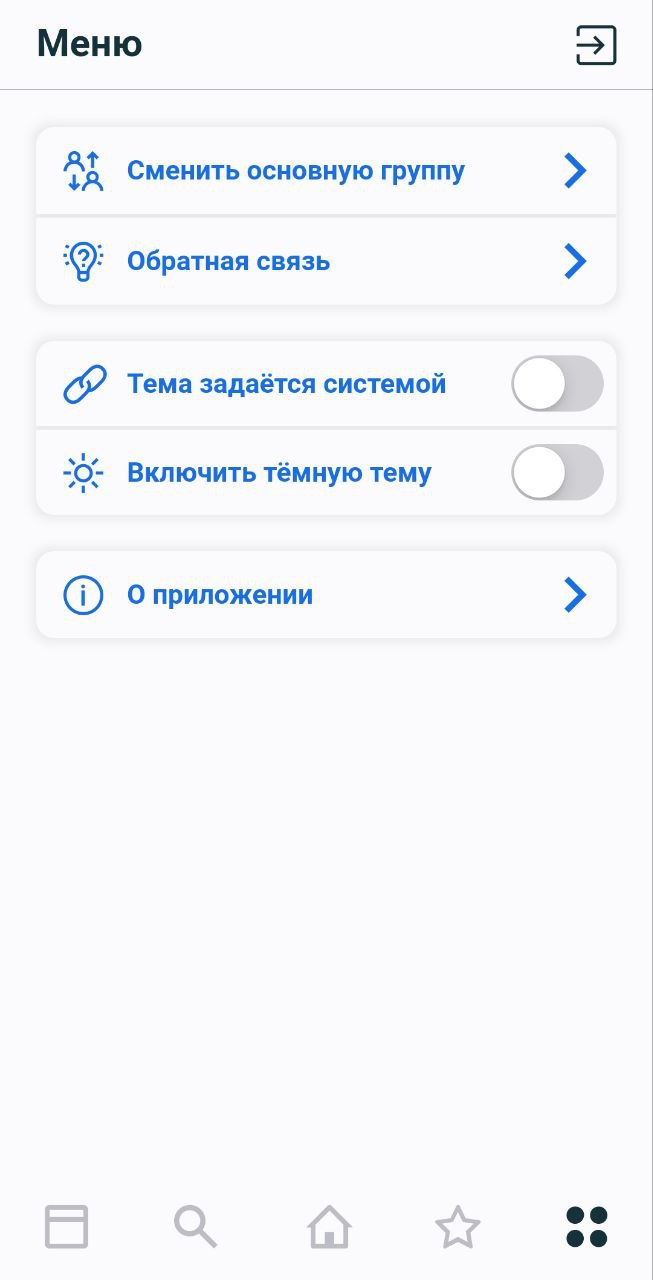


Рисунок 1.21 – Сторінка налаштувань

Переваги:

* зручний інтерфейс перегляду розкладу;
* зручний пошук групи;
* перегляд новин університету;

Недоліки:

* перегляд тільки поточного розкладу;
* неможливість перегляду розкладу сесії;
* відсутність розкладів для викладачів.

4. Мобільний додаток розробника Kravcov Denis «Розклад занять і Календар» дата випуску 5 серпня 2021р. Це універсальний додаток, який реалізує гнучкий варіант створення розкладу занять власноруч.

Можливості програми:

* відсутня реклама.
* декілька розкладів одночасно.
* розклад для 1, 2, 3, 4-х тижнів.
* будь-який гнучкий графік навчання.
* віджет Сьогодні з таймером занять.
* нагадування про заняття та завдання.
* фото, відео, аудіо та документи для завдань.
* простий експорт розкладу вашим друзям.
* програма доступна на Wear OS.

Опис інтерфейсу:

Програма має наступні екрани:

* головний екран розкладу, на якому відображається розклад за днем тижня та дата занять (рис 1.22)
* згорнутий екран розкладу, на якому відображається розклад, з параметрами таймеру та можливість навігації по дням (рис 1.23)

|  |  |
| --- | --- |
|  |  |
| Рисунок 1.22 – Головний екран розкладу | Рисунок 1.23 – Згорнутий екран розкладу |

* екран вибору палітри кольорів для занять та завдань (рис 1.24)
* екран домашніх завдань на кожен день (рис 1.25)

|  |  |
| --- | --- |
|  |  |
| Рисунок 1.24 – Екран вибору палітри кольорів | Рисунок 1.25 – Екран домашніх завдань |

* екран параметрів типу розкладу, де налаштовується сам тиждень та повторюваність розкладу по тижням (рис 1.26)

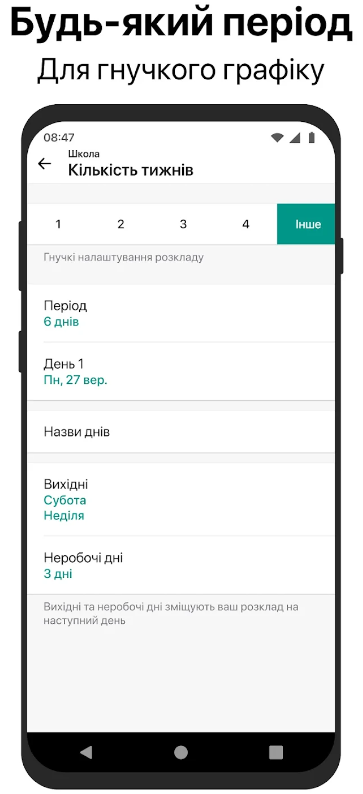
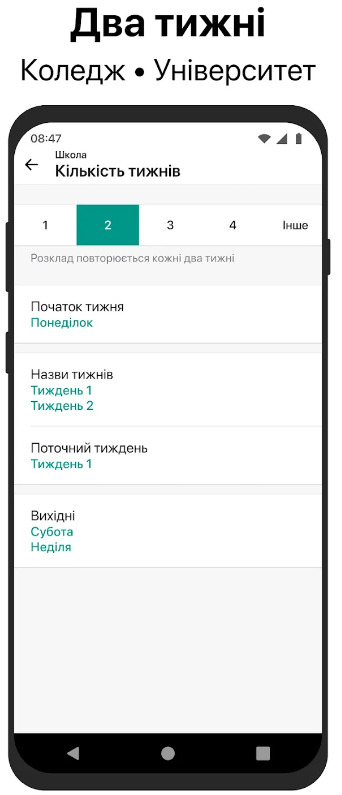
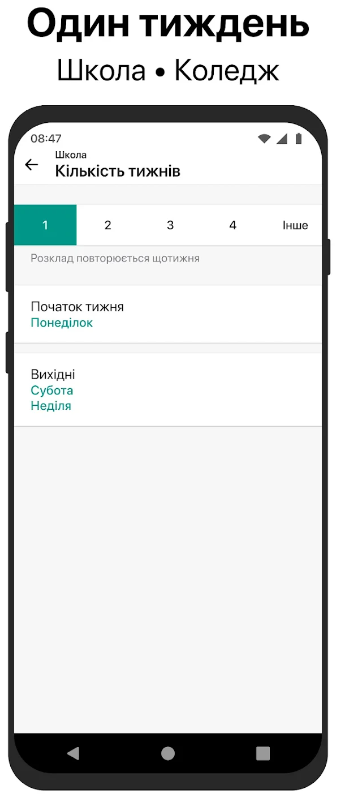


Рисунок 1.26 – Екран налаштування тижнів розкладу

* екран для надсилання розкладу, на якому можна надіслати створений розклад будь-яким зручним способом (рис 1.27)

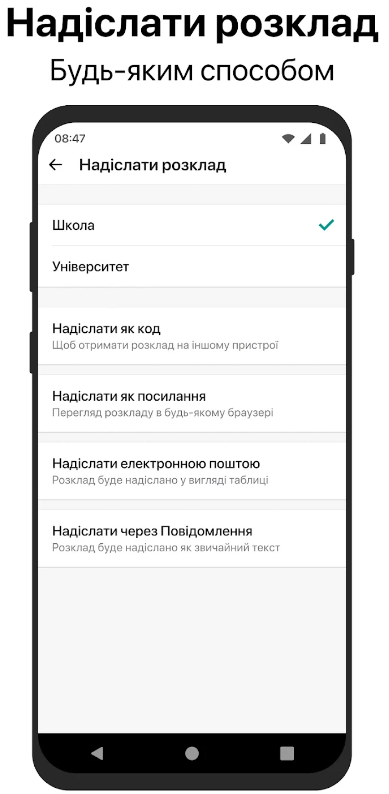


Рисунок 1.27 – Екран для надсилання розкладу

Переваги:

* зручний інтерфейс для створення та перегляду розкладу;
* можливість змінювати дизайн додатку;
* можливість отримання чи надсилання розкладу;
* гнучке налаштування розкладу.

Недоліки:

* розклади необхідно створювати вручну або отримувати розклад;
* відсутність автоматизацій отримання розкладів з освітніх установ.

1. Постановка задачі

Розробити мобільний додаток для перегляду розкладів занять та модулів Українського державного університету науки і технологій, в якому буде надаватися зручний та інтуїтивно зрозумілий інтерфейс.

Програмний продукт повинен надавати можливість:

* отримання інформації про розклади з офіційного сайту університету;
* відображення списку доступних розкладів;
* відображення обраного розкладу занять для групи;
* відображення обраного розкладу занять для викладача;
* відображення обраного розкладу модулів для групи;
* відображення обраного розкладу модулів для викладача.

# ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

1. Опис функціональних характеристик

Програмний додаток призначений для перегляду доступних розкладів занять та модулів для студентів та викладачів.

Програмний продукт повинен надавати можливість надавання списку розкладів з офіційного сайту університету:

* отримання інформації про розклади з офіційного сайту університету;
* відображення списку доступних розкладів;
* відображення обраного розкладу занять для групи;
* відображення обраного розкладу занять для викладача;
* відображення обраного розкладу модулів для групи;
* відображення обраного розкладу модулів для викладача.

1. Вхідні дані

Вхідними даними програми, що розробляється є:

* html вміст сторінки сайту ust.edu.ua;
* файл з розкладом.

1. Вихідні дані

Результатом роботи програми є наступні вихідні дані:

* список розкладів з сайту університету;
* заняття обраного розкладу;
* сформована на основі вхідної інформації база даних (табл 2.1 – 2.2).

Таблиця 2.1 – Розклади

|  |  |  |  |
| --- | --- | --- | --- |
| Назва | Рік | Url-адреса | Контрольна сума |
|  |  |  |  |

Примітки:

Назва – назва розкладу, рядок.

Рік – навчальний рік розкладу, рядок.

Url-адреса – Url-адреса за якою завантажується розклад, рядок.

Контрольна сума – контрольна сума завантаженого файлу, рядок.

Таблиця 2.2 – Заняття

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Розклад | Назва | Назва групи | Альтернативна назва групи | Викладач | Аудиторія |
|  |  |  |  |  |  |

Примітки:

Розклад – назва розкладу до якого входить занять, рядок.

Назва – назва заняття, рядок.

Назва групи – назва групи, рядок.

Альтернативна назва групи – інша назва групи, рядок.

Викладач – Прізвище та ініціали викладача, рядок.

Аудиторія – номер аудиторії, рядок.

Таблиця 2.3 – Продовження таблиці 2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| День тижня | Номер | Тип тижня | Тип заняття | Час |
|  |  |  |  |  |

Примітки:

День тижня – номер дня тижня, ціле число.

Номер – номер заняття, ціле число.

Тип тижня – тип тижня заняття (чисельник/знаменник/немає), рядок.

Тип заняття – тип заняття (консультація/модуль/немає), рядок.

1. Вибір мови програмування

Для розробки мобільного додатку була обрана мова програмування C# за допомогою фреймворку Xamarin.Forms, що дозволяє створювати програми для різних операційних систем.

Xamarin дозволяє створювати одну єдину логіку програми із застосуванням C# і .NET відразу для всіх трьох платформ - Android, iOS, UWP.

Переваги використання Xamarin.Forms:

* у процесі розробки створюється єдиний код для всіх платформ
* Xamarin надає прямий доступ до нативних API кожної платформи
* при створенні додатків ми можемо використовувати платформу.
* Xamarin Forms підтримує декілька платформ. Основні платформи: Android, iOS, UWP.

Графічне представлення роботи Xamarin (рис 2.1)

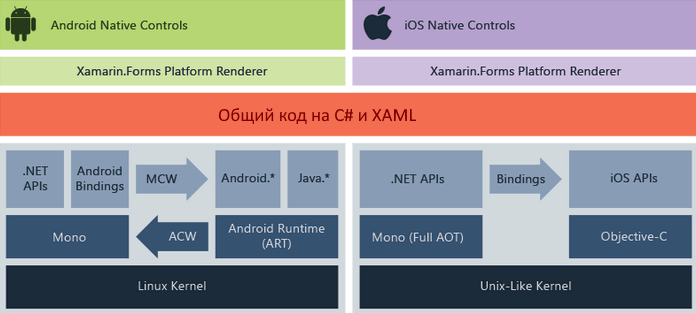


Рисунок 2.1 – Схема роботи Xamarin

Xamarin працює поверх фреймворку Mono, який надає opensource-реалізацію .NET Framework. Mono може працювати поверх різних платформ – Linix, MacOS тощо.

На рівні кожної окремої платформи Xamarin покладається на низку субплатформ. Зокрема:

* Xamarin.Android - бібліотеки для створення програм на ОС Android;
* Xamarin.iOS - бібліотеки для створення програм для iOS.

Ці субплатформи відіграють велику роль - через них програми можуть надавати запити до прикладних інтерфейсів на пристроях під керуванням ОС Android або iOS. Коротко це виглядає так.

За допомогою Xamarin.Android код C# з використанням Xamarin компілюється в Intermediate Language (IL), який після запуску програми компілюється в нативну збірку. Xamarin-програми запускаються в середовищі виконання Mono. Безпосередньо код не може звертатися до API Android. Для цього треба звернутися до функціональності просторів імен Android.\* і Java.\*, які надаються віртуальною машиною Android Runtime (ART). Спеціальний прошарок Managed Callable Wrappers (MCW) дозволяє транслювати виклики managed-коду в нативні виклики та звертатися до функціональності просторів імен Android.\* та Java.\*

І навпаки, коли Android Runtime (ART) звертається до програми з кодом Xamarin, всі виклики проходять через обгортку Android Callable Wrappers (ACW) .

Програми Xamarin.iOS на відміну від Xamarin.Android, який використовує JIT-компіляцію, застосовують AOT-компіляцію (Ahead-of-Time) коду C# в нативний ARM-код. Xamarin використовує проміжний шар Selectors (селектори) для трансляції викликів коду Objective-C код на C# і шар Registrars (реєстратори) для трансляції коду C# Objective-C. У результаті шари Selectors і Registrars загалом представляють переміжний шар, який на ілюстрації вище позначений як "bindings" і який дозволяє взаємодіяти коду Objective-C з кодом C#.

У результаті завдяки цим платформам ми можемо створювати окремо додатки для Android, окремо для iOS, але найбільш важливою особливістю Xamarin є можливість створювати кросплатформні додатки – тобто одна логіка для всіх платформ. Дана можливість представлена ​​технологією Xamarin.Forms і яка працює рівнем вище Xamarin.Android і Xamarin.iOS. Тобто за допомогою Xamarin.Forms ми один раз можемо визначити візуальний інтерфейс, один раз до нього прив'язати якусь логіку на C# і все це буде працювати на Android, iOS та Windows. Потім Xamarin.Forms за допомогою рендерерів (renderer) – спеціальних об'єктів для зв'язку контролів на XAML/C# з нативними контролами транслюють візуальні компоненти Xamarin.Forms у графічний інтерфейс, специфічний для кожної платформи [1].

1. Опис компонетно-орієнтованого підходу

У розробці програмного додатку використовується компонетно-орієнтованого підхід, що полегшує роботу з великими за об’ємамом програмним застосунками та надає можливість швидко та легко виправляти, видозмінювати або доповнювати функціональності.

Компонентне-орієнтоване програмування (КОП) – одна з парадигм програмування, що виникла як свого роду дисципліна, тобто набір певних обмежень, що накладаються на механізм об’єктно-орієнтованого програмування (ООП). Компонентно-орієнтоване програмування усуває проблему безконтрольного застосування ООП, що раніше призводило до виникнення проблем з надійністю великих програмних комплексів.

Суть КОП – є певний клас-контейнер, а також клас-компонент, який можна додати в клас-контейнер. Об’єкт складається з контейнера і компонентів в цьому контейнері.

Компоненти трохи нагадують інтерфейси. Але інтерфейси дозволяють тільки виділити у класів загальну сигнатуру функцій і властивостей, а компоненти дозволяють винести загальну реалізацію класів окремо.

В ООП підході об’єкт визначається, як екземпляр класу. У КОП підході об’єкт визначається компонентами, з яких він складається. Не важливо, що це за об’єкт. Важливо, що у нього є і що він вміє робити.

КОП спрощує повторне використання написаного коду – використання одного компонента в різних об’єктах. Також з різних комбінацій вже існуючих компонентів, можна зібрати новий тип об’єкту [2].

1. Проектування екрані мобільного додатку

Мобільний додаток маж складатись з двох екранів:

* екран переліку доступних розкладів;
* екран відображення обраного розкладу.

1. Проектування інтерфейсу користувача мобільного додатку

Для взаємодії користувача з програмою інтерфейс повинен містити наступні компоненти:

* StackLayout – компонент інтерфейсу, який необхідний для організації положення елементів на екрані;
* Frame – компонент інтерфейсу, який виконує роль групування компонентів;
* ListView – компонент інтерфейсу, який відображає список рокзладів;
* ScrollView – компонент інтерфейсу, який необхідний для гортання екрану по вертикалі;
* Picker – компонент інтерфейсу, який при натисканні на нього відображає список елементів, що дає можливість обрати один з варіантів;
* CheckBox – компонент інтерфейсу, який дає змогу обрати для яких днів тижня переглянути розклад;
* RadioButton – компонент інтерфейсу, який дає змогу обрати тип користувача, для кого відобразити розклад;
* Label – компонент інтерфейсу, який виводить текст на еркан;
* BoxView – компонент інтерфейсу, який необхідний для візуальної розмітки та створення пустого простору, у разі відсутності за чисельником або знаменником.

1. Проектування навігації між екранами мобільного додатку.

Навігацію між екранами у розроблювальному мобільному додатку здійснюється за допомогою класу NavigationPage. Взаємодія між екранами зображена за допомогою діаграми станів (рис 2.2).

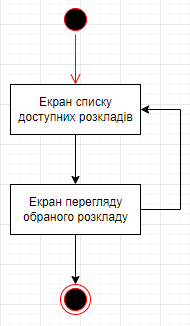


Рисунок 2.2 – Діаграма станів

1. Проектування дизайну мобільного додатку

При проектуванні дизайну мобільного додатку були розроблені ескізи для кожного компоненту програми:

* компонент типу Frame – заголовок екрану (рис 2.3)

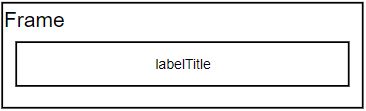


Рисунок 2.3 – Ескіз заголовку екрану розкладів

* компонент типу ListView (рис 2.4)

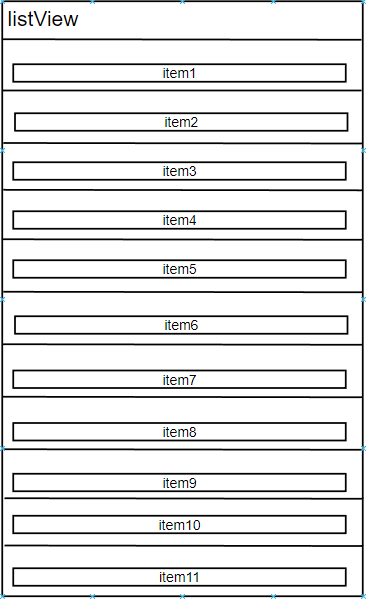


Рисунок 2.4 – Ескіз компоненту ListView

* компонент типу RadioButton (рис 2.5)



Рисунок 2.5 – Ескіз компоненту RadioButton

* компонент типу Picker (рис 2.6)



Рисунок 2.6 – Ескіз компоненту Picker

* компонент типу CheckBox (рис 2.7)

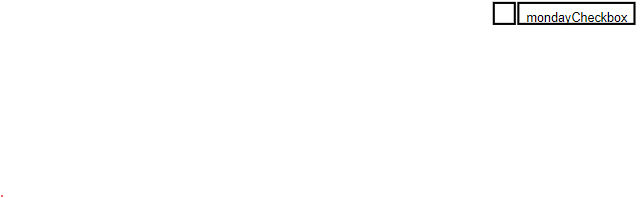


Рисунок 2.7 – Ескіз компоненту CheckBox

* компонент типу Label «Номер занять» (рис 2.8)



Рисунок 2.8 – Ескіз компоненту Label «Номер занять»

* компонент типу Label «Інформація про заняття» (рис 2.9)

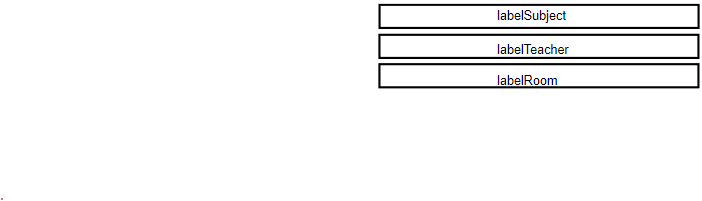


Рисунок 2.9 – Ескіз компоненту Label «Інформація про заняття»

* компонент типу StackLayout «Панель відображення розкладу» (рис 2.10)

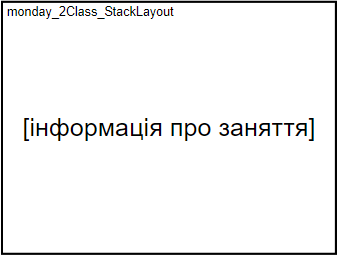


Рисунок 2.10 – Ескіз компоненту StackLayout «Панель відображення розкладу»

* компонент типу BoxView «Прямокутник розмітки» (рис 2.11)



Рисунок 2.11 – Ескіз компоненту BoxView «Прямокутник розмітки»

1. Розробка інтерфейсу користувача

Для кожного компоненту було обрано свій стиль та властивості:

* компонент типу Frame (заголовок екрану перегляду розкладів)

Програмний код опису:

<Frame BackgroundColor="#2196F3" Padding="24" CornerRadius="0">

<Label Text="Оберіть розклад" HorizontalTextAlignment="Center" TextColor="White" FontSize="32" />

</Frame>

Вигляд компоненту Frame зображено на рис. 2.6.



Рисунок 2.6 – Вигляд заголовку екрану перегляду розкладів

* компонент типу ListView (список розкладів)

Програмний код опису:

<ListView x:Name="listView" ItemSelected="onScheduleSelected" HorizontalOptions="Center"> </ListView>

Вигляд компоненту ListView зображено на рис. 2.7.



Рисунок 2.7 – Вигляд списку розкладів

* компоненти типу RadioButton (вибір критерію)

Програмний код опису:

<RadioButton x:Name="groupRadioButton" Content ="Студент" IsChecked="True" CheckedChanged="onScheduleTypeChange" />

<RadioButton x:Name="teacherRadioButton" Content ="Викладач" IsChecked="False" CheckedChanged="onScheduleTypeChange" />

Вигляд компонентів RadioButton зображено на рис. 2.8.



Рисунок 2.8 – Вигляд вибору критерію

* компоненти типу Picker (вибір групи/викладача)

Програмний код опису:

<Picker x:Name="groupPicker" SelectedIndexChanged="onSelectedGroupChange" Title="Оберіть групу" >

<Picker.Items>

</Picker.Items>

</Picker>

<Picker x:Name="teacherPicker" SelectedIndexChanged="onSelectedTeacherChange" Title="Оберіть викладача" IsVisible="False">

<Picker.Items>

</Picker.Items>

</Picker>

Вигляд компонентів Picker зображено на рис. 2.9 – 2.10.

|  |  |
| --- | --- |
|  |  |
| Рисунок 2.9 – Вигляд вибору групи | Рисунок 2.10 – Вигляд вибору викладача |

* компоненти типу CheckBox та Label (прапорці днів тижня)

Програмний код опису:

<CheckBox x:Name="mondayCheckbox" CheckedChanged="onMondayCheckChange" />

<Label Text="Понеділок" FontSize="26" FontAttributes="Bold" />

<CheckBox x:Name="tuesdayCheckbox" CheckedChanged="onTuesdayCheckChange" />

<Label Text="Вівторок" FontSize="26" FontAttributes="Bold" />

<CheckBox x:Name="wednesdayCheckbox" CheckedChanged="onWednesdayCheckChange"/>

<Label Text="Середа" FontSize="26" FontAttributes="Bold" />

<CheckBox x:Name="thursdayCheckbox" CheckedChanged="onThursdayCheckChange" />

<Label Text="Четвер" FontSize="26" FontAttributes="Bold" />

<CheckBox x:Name="fridayCheckbox" CheckedChanged="onFridayCheckChange" />

<Label Text="П'ятниця" FontSize="26" FontAttributes="Bold" />

Вигляд компонентів CheckBox та Label зображений на рис. 2.11.



Рисунок 2.11 – Вигляд прапорців днів тижня

* компоненти типу Frame для відображення розкладу

Програмний код опису для тижня П’ятниця (для інших днів тижня застосовується аналогічний опис з відповідними іменами):

<Frame x:Name="fridayFrame" IsVisible="false" BackgroundColor="Gray">

<StackLayout>

<StackLayout Orientation="Horizontal">

<Label Text="I" FontSize="22" FontAttributes="Bold" WidthRequest="25" />

<BoxView Color="#FCFCFC" WidthRequest="2" HorizontalOptions="Fill" />

<StackLayout x:Name="friday\_1Class\_StackLayout">

</StackLayout>

</StackLayout>

<BoxView Color="#FCFCFC" HeightRequest="2" HorizontalOptions="Fill" />

<StackLayout Orientation="Horizontal">

<Label Text="II" FontSize="22" FontAttributes="Bold" WidthRequest="25"/>

<BoxView Color="#FCFCFC" WidthRequest="2" HorizontalOptions="Fill" />

<StackLayout x:Name="friday\_2Class\_StackLayout">

</StackLayout>

</StackLayout>

<BoxView Color="#FCFCFC" HeightRequest="2" HorizontalOptions="Fill" />

<StackLayout Orientation="Horizontal">

<Label Text="III" FontSize="22" FontAttributes="Bold" WidthRequest="25"/>

<BoxView Color="#FCFCFC" WidthRequest="2" HorizontalOptions="Fill" />

<StackLayout x:Name="friday\_3Class\_StackLayout">

</StackLayout>

</StackLayout>

<BoxView Color="#FCFCFC" HeightRequest="2" HorizontalOptions="Fill" />

<StackLayout Orientation="Horizontal">

<Label Text="IV" FontSize="22" FontAttributes="Bold" WidthRequest="25"/>

<BoxView Color="#FCFCFC" WidthRequest="2" HorizontalOptions="Fill" />

<StackLayout x:Name="friday\_4Class\_StackLayout">

</StackLayout>

</StackLayout>

<BoxView Color="#FCFCFC" HeightRequest="2" HorizontalOptions="Fill" />

<StackLayout Orientation="Horizontal">

<Label Text="V" FontSize="22" FontAttributes="Bold" WidthRequest="25" />

<BoxView Color="#FCFCFC" WidthRequest="2" HorizontalOptions="Fill" />

<StackLayout x:Name="friday\_5Class\_StackLayout">

</StackLayout>

</StackLayout>

<BoxView Color="#FCFCFC" HeightRequest="2" HorizontalOptions="Fill" />

<StackLayout Orientation="Horizontal">

<Label Text="VI" FontSize="22" FontAttributes="Bold" WidthRequest="25"/>

<BoxView Color="#FCFCFC" WidthRequest="2" HorizontalOptions="Fill" />

<StackLayout x:Name="friday\_6Class\_StackLayout">

</StackLayout>

</StackLayout>

</StackLayout>

</Frame>

Вигляд компонентів Frame для днів тижня зображений на рис. 2.12.



Рисунок 2.12 – Вигляд компонентів Frame для днів тижня

* компоненти типу Label для відображення інформації про заняття

Програмний код опису

labelSubject.FontSize = 18; labelSubject.FontAttributes = FontAttributes.Bold;

labelGroup.FontSize = 18; labelGroup.FontAttributes = FontAttributes.Italic;

labelTeacher.FontSize = 18; labelTeacher.FontAttributes = FontAttributes.Italic;

labelRoom.FontSize = 16;

labelDate.FontSize = 20; labelSubject.FontAttributes = FontAttributes.Bold;

labelSubType.FontSize = 20; labelSubject.FontAttributes = FontAttributes.Bold;

labelSeparator.Text = "----------------------------------------------";

labelSeparator.FontAttributes = FontAttributes.Bold;

Вигляд компонентів Label для інформації про заняття на рис. 2.13 – 2.16.

|  |  |
| --- | --- |
|  |  |
| Рисунок 2.13 – Вигляд компонентів Label для розкладу занять груп | Рисунок 2.14 – Вигляд компонентів Label для розкладу занять викладачів |
|  |  |
| Рисунок 2.15 – Вигляд компонентів Label для модулів груп | Рисунок 2.16 – Вигляд компонентів Label для модулів викладачів |

# РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

# Розробка динаміки системи

Для відображення динаміки системи використані діаграми діяльності (рис. 3.1) та діаграма послідовності (рис. 3.2).

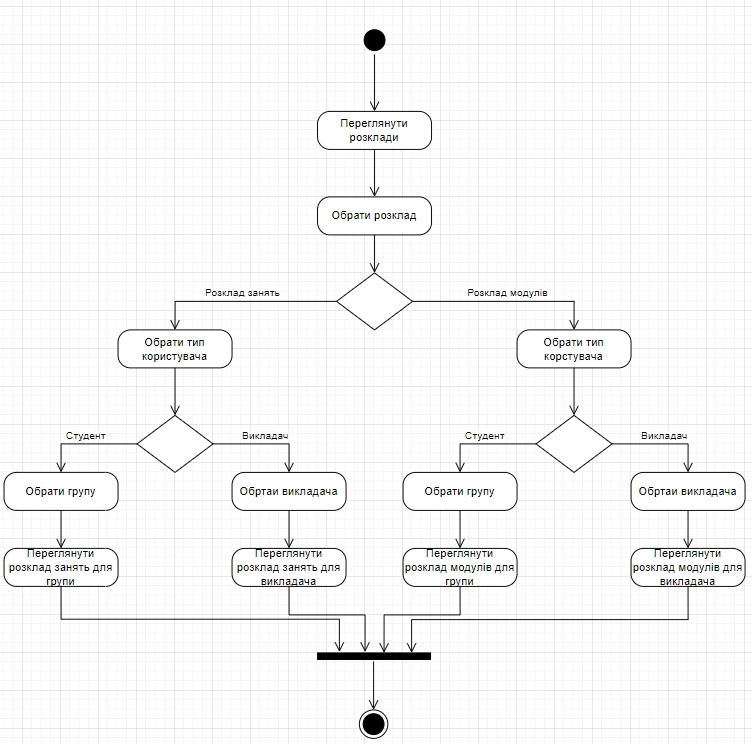


Рисунок 3.1 – Діаграма діяльності



Рисунок 3.2 – Діаграма послідовності

# Розробка екранів мобільного додатку

Мобільний додаток складається з 2 екранів:

* екран перегляду доступних розкладів;
* екран перегляду обраного розкладу.

Перелік використаних компонентів на екранах приведені в табл. 3.1 – 3.2.

Таблиця 3.1– Компоненти екрану перегляду доступних розкладів

|  |  |  |
| --- | --- | --- |
| Компонент | Назва | Опис |
| Frame | frameTitle | Заголовок вікна |
| Label | labelTitle | Текст «Оберіть розклад» |
| ListView | listView | Список розкладів |

Таблиця 3.2 – Компоненти екрану перегляду доступних розкладів

|  |  |  |
| --- | --- | --- |
| Компонент | Назва | Опис |
| ScrollView |  | Прокрутка сторінки |
| RadioButton | groupRadioButton | Пермикач на групу |
| RadioButton | teacherRadioButton | Перемикач на викладача |
| Picker | groupPicker | Список вибору доступних груп |
| Picker | teacherPicker | Список вибору доступних викладачів |
| CheckBox | mondayCheckbox | Прапорець для відкривання розкладу на понеділок |
| Label |  | Текст «Понеділок» для mondayCheckbox |
| Frame | mondayFrame | Панель для розкладу на понеділок |
| Label |  | Текс «I» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | monday\_1Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «II» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | monday\_2Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «III» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | monday\_3Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «IV» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | monday\_4Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «V» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | monday\_5Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «VI» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | monday\_6Class\_StackLayout | Контейнер для відображення інформації про заняття |
| CheckBox | mondayCheckbox | Прапорець для відкривання розкладу на вівторок |
| Label |  | Текст «Вівторок» для tuesdayCheckbox |
| Frame | tuesdayFrame | Панель для розкладу на понеділок |
| Label |  | Текс «I» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | tuesday\_1Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «II» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | tuesday\_2Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «III» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | tuesday\_3Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «IV» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | tuesday\_4Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «V» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | tuesday\_5Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «VI» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | tuesday\_6Class\_StackLayout | Контейнер для відображення інформації про заняття |
| CheckBox | wednesdayCheckbox | Прапорець для відкривання розкладу на вівторок |
| Label |  | Текст «Середа» для wednesdayCheckbox |
| Frame | wednesdayFrame | Панель для розкладу на понеділок |
| Label |  | Текс «I» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | wednesday\_1Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «II» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | wednesday\_2Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «III» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | wednesday\_3Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «IV» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | wednesday\_4Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «V» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | wednesday\_5Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «VI» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | wednesday\_6Class\_StackLayout | Контейнер для відображення інформації про заняття |
| CheckBox | thursdayCheckbox | Прапорець для відкривання розкладу на вівторок |
| Label |  | Текст «Середа» для thursdayCheckbox |
| Frame | thursdayFrame | Панель для розкладу на понеділок |
| Label |  | Текс «I» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | thursday\_1Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «II» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | thursday\_2Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «III» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | thursday\_3Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «IV» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | thursday\_4Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «V» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | thursday\_5Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «VI» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | thursday\_6Class\_StackLayout | Контейнер для відображення інформації про заняття |
| CheckBox | fridayCheckbox | Прапорець для відкривання розкладу на вівторок |
| Label |  | Текст «Середа» для fridayCheckbox |
| Frame | thursdayFrame | Панель для розкладу на понеділок |
| Label |  | Текс «I» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | friday\_1Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «II» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | friday\_2Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «III» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | friday\_3Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «IV» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | friday\_4Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «V» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | friday\_5Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label |  | Текс «VI» для позначення номеру занняття |
| BoxView |  | Пряма лінія |
| StackLayout | friday\_6Class\_StackLayout | Контейнер для відображення інформації про заняття |
| Label | labelSubject | Назва заняття |
| Label | labelTeacher | Прізвище та ініціали викладача (для груп) |
| Label | labelGroup | Назва групи (для викладачів) |
| Label | labelRoom | Номер аудиторії |
| Label | labelSeparator | Роздільник у разі якщо заняття має тип Чисельник чи Знаменник |
| BoxView | box | Створення пустого простору якщо немає заняття по чисельнику чи знаменнику |
| Label | labelDate | Дата заняття (для розкладу модулів) |
| Label | labelSubType | Тип заняття (для розкладу модулів) |

# Розробка інтерфейсу мобільного додатку

При розробці екранів мобільного додатку були розроблені ескізи інтерфейсу екранів (рис3.3 – 3.4).

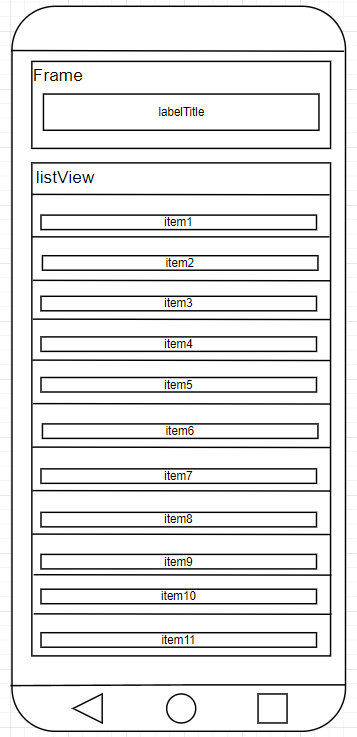


Рисунок 3.3 – Ескіз екрану перегляду доступних розкладів

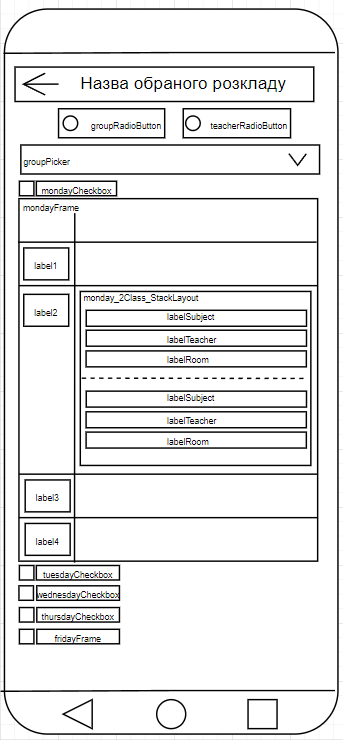


Рисунок 3.4 – Ескіз екрану перегляду обраного розкладу

# Розробка дизайну мобільного додатку

Дизайн екранів програмного додатку зображено на рис. 3.5 – 3.6



Рисунок 3.5 – Дизайн екрану перегляду обраного розкладу



Рисунок 3.6 – Дизайн екрану перегляду обраного розкладу

# ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ

# Вибір методів тестування

Для покращення якості програмного забезпечення використовуються два основні підходи для реалізації процесу тестування: функціональний (чорний ящик) та структурний (білий ящик).

При функціональному тестуванні у тестувальників немає доступу до вихідний коду програми, і програма розглядається як «чорний ящик». Тож його суть полягає в перевірці відповідності програми своїй специфікації.

Найбільш оптимальними видами функціонального тестування є:

* випадкове (стохастичне) тестування. Створюються незалежні тести із випадково генерованих вхідних даних. Значним недоліком є велика загальна кількість тестів, які треба генерувати з огляду на надійність програми.
* тестування за класами еквівалентності. Множина вхідних і вихідних даних розбивається на класи еквівалентності. Поділ проходить таким чином, що кожен тест, що входить до певного класу, є еквівалентним будь-якому іншому тесту цього класу, тобто програма однаково реагує на всі тести одного класу.
* метод аналізу граничних умов. Перевіряються випадки, що виникають безпосередньо на межах вхідних та вихідних даних. Так як більшість помилок з'являється не в центрі множини допустимих значень, а на верхній та нижній границях, відповідно основна увага зосереджена саме на них.

Структурне тестування полягає в перевірці та аналізі вихідного коду програми.

Основними різновидами структурного тестування є:

* Тестування маршрутів. Перевірка програмного коду виконується через проходження певного зазначеного шляху. На практиці, під час тестування, часто деякі маршрути залишаються неперевіреними. Складність в тестуванні полягає в тому, що всі маршрути мають бути перевірені ще під час створення програми, що вимагає великих часових затрат.
* Тестування обробки даних. Процес виконання програми можна розуміти як роботу з певними даними, що передаються із вхідного потоку у вихідний. Із вхідних даних формуються проміжні результати, що аналізуються до моменту їх виведення. Помилки виявляються як перекручування проміжних результатів або, взагалі, їхня відсутність.
* Тестування циклів. Цикли в програмі можуть значно ускладнити весь процес тестування та відлагодження. Повне тестування повинно включати в себе перевірку всіх можливих маршрутів в кожній ітерації циклу та всіх можливих сполучень циклів із ациклічною частиною маршруту[3].

# Тестування

# **Загальні висновки**

# **Список використаних джерел**

1. <https://metanit.com/sharp/xamarin/1.1.php>
2. <https://elartu.tntu.edu.ua/bitstream/lib/34279/2/VIII_NTK_2020_Yatsyshyn_V_V-Benefits_of_component_124-125.pdf>
3. https://conf.ztu.edu.ua/wp-content/uploads/2016/06/248.pdf