Київський національний торговельно-економічний університет Кафедра інженерії програмного забезпечення та кібербезпеки

# ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

## «Моделювання нативного мобільного додатку для системи Android "My KNUTE"»

Студента 2 курсу, 23 групи,		
спеціальності 121		
«Інженерія програмного		
забезпечення»,		
спеціалізації «Інженерія		Грімов Артем Ігорович
програмного забезпечення»		
	підпис студента	
Науковий керівник		
доктор технічних наук,		
професор кафедри інженерії		
програмного забезпечення та		Криворучко Олена
кібербезпеки		Володимирівна
	підпис керівника	
Гарант освітньої програми		
доктор економічних наук,		
професор кафедри інженерії		
програмного забезпечення та		Токар Володимир
кібербезпеки	підпис керівника	Володимирович

КИЇВ – 2021

#### Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра <u>інженерії програмного забезпечення та кібербезпеки</u>

Освітній ступінь магістр

Спеціальність <u>121 «Інженерія програмного забезпечення»</u>

Спеціалізація / освітня програма «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри <u>інженерії програмного</u> забезпечення та кібербезпеки Криворучко О.В. «29» грудня 2020 р.

## Завдання на випускний кваліфікаційний проект студентові Грімову Артему Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту <u>«Моделювання нативного</u> мобільного додатку для системи Android "My KNUTE"»

Затверджена наказом ректора від <u>«28»</u> грудня 2020 р. № 3923

2. Строк здачі студентом закінченого проекту (роботи) «03» листопада 2021 р.

3. Цільова установка та вихідні дані до проекту (роботи)

*Мета проекту* розробка нативного мобільного додатку на платформі Android для задоволення потреб учасників освітнього процесу у зручному доступі до організаційної інформації, що супроводжує навчання.

Об'єкт дослідження використання мобільних додатків в освітньому процесі.

Предмет дослідження розробка Android-додатку "Му KNUTE".

#### 4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант	Підпис, дата				
	(прізвище, ініціали)	Завдання видав	Завдання прийняв			

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

<u>РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУВАННЯ ВИМОГ ДО</u> МОБІЛЬНОГО ДОДАТКУ "МУ КNUTE"

<u>1.1. Місце мобільних додатків в освітньому процесі та основні вимоги до</u> додатка "Му KNUTE"

<u>1.2. Аналіз ринку технологій розробки мобільних додатків: нативні та крос-</u>платформні рішення

1.3. Технічне завдання на розробку мобільного додатку "Му KNUTE"

1.4. Висновки до розділу 1

<u>РОЗДІЛ 2. ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СИСТЕМИ</u> ANDROID ТА ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ РОЗРОБКИ

2.1. Життєвий цикл нативного мобільного додатку для OC Android

2.2. Середовище розробки Android Studio. Структура типового проекту

2.3. Вибір мови програмування нативного Android-додатку

2.4. Проектування інтерфейсу мобільного додатку "Му KNUTE"

2.5. Висновки до розділу 2

<u>РОЗДІЛ 3. РОЗРОБКА НАТИВНОГО ANDROID-ДОДАТКУ "МУ KNUTE"</u>

3.1. Опис модулів мобільного додатку "Му KNUTE"

3.2. Реалізація інтерфейсу користувача

3.3. Шляхи вдосконалення нативного Android-додатку "Му KNUTE"

3.4. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

Ma	Назва етапів випускного кваліфікаційного	Строк викон	ання етапів
JN <u>0</u>	проекту	прое	кту
nop.		за планом	фактично
1	2	3	4
1.	Вибір теми випускного кваліфікаційного проект	21.09.2020	21.09.2020
2.	Розробка та затвердження завдання на проект	29.12.2020	29.12.2020
	магістра		
3.	Вступ та перелік літературних джерел	27.02.2021	25.02.2021
4.	Розробка технічного завдання	20.03.2021	20.03.2021
5.	Розділ 1. Аналіз предметної області та	16.04.2021	10.04.2021
	формування вимог до мобільного додатку "Му		
	KNUTE"		
6.	Розділ 2. Проектування мобільного додатку для	24.05.2021	20.05.2021
	системи Android та вибір інструментарію для		
	розробки		
7.	Розділ 3. Розробка нативного Android-додатку	20.09.2021	15.09.2021
	"My KNUTE"		
8.	Написання наукової статті	22.05.2021	22.05.2021
9.	Керівництво користувача	21.10.2021	21.10.2021
10.	Висновки та пропозиції	01.11.2021	25.10.2021
11.	Здача випускного кваліфікаційного проекту на	03.11.2021	03.11.2021
	кафедру (перша перевірка)		
12.	Підготовка автореферату та презентації доповіді	03.11.2021	03.11.2021
13.	Попередній захист випускного кваліфікаційного	22.11.2021 -	24.11.2021
	проекту	25.11.2021	
14.	Здача зброшурованої випускного	25.11.2021	25.11.2021
	кваліфікаційного проекту		
15.	Зовнішнє рецензування випускного	26.11.2021	26.11.2021
	кваліфікаційного проекту		
16.	Підготовка до публічного захисту випускного		
	кваліфікаційного проекту		

### 6. Календарний план виконання проекту

7. Дата видачі завдання <u>«29» грудня 2020 р.</u>

8. Науковий керівник випускного кваліфікаційного проекту

	Криворучк	<u>o O.B.</u>
	(прізвище, ін	іціали, підпис)
9. Гарант освітньої програми _	Тока	p B.B.
	(прізвище, і	ніціали, підпис)
10. Завдання прийняв до виког	нання студент	Грімов А.І.
		(прізвище, ініціали, підпис)

4	4	D'	•		• 1 •	•••	
		Кілгук	кепівникя	випускного	кваліфікаі	пиного	ΠΠΛΕΚΤΥ
-		БЩТУК	керилика	Dhing children	NDainpina	ummor o	npockij

Науковий керівник випускного кваліфікаційного проекту
(nionuc, oama)
Відмітка про попередній захист Криворучко О.В.
12 Phonopok uno punyokung knowichikowichung unooku
12. Бисновок про випускний квалифікаційний проект
Випускний кваліфікаційний проект стулента Грімова А І
(прізвище, ініціали)
може бути допушений до захисту екзаменаційній комісії.
Гарант освітньої програми Токар В.В.
(прізвище, ініціали, підпис)
Завідувач кафедри Криворучко О.В.
(підпис, прізвище, ініціали)
«»20p.

#### АНОТАЦІЯ

Кваліфікаційна робота: 45 с., 23 рис., 27 додатків, 10 джерел.

Мета роботи: розробка нативного мобільного додатку на платформі Android для задоволення потреб учасників освітнього процесу у зручному доступі до організаційної інформації, що супроводжує навчання.

Об'єкт дослідження: використання мобільних додатків в освітньому процесі.

Предмет досліджень: розробка Android-додатку "Му KNUTE".

Методи дослідження: порівняння, аналіз, метод об'єктно-орієнтованого програмування, візуалізація.

У процесі роботи виявлено інформаційні потреби студентів та співробітників КНТЕУ для використання у проекті, складено технічне завдання на розробку нативного Android-додатку "Му KNUTE", базуючись на останніх тенденціях спроектовано інтерфейс та логіку додатку, сформовано висновки та пропозиції щодо подальшого розвитку розробленого додатку.

Результатом роботи є створений за допомогою сучасного інструментарію (середовище розробки Android Studio, мова розмітки XML та мова програмування Kotlin) нативний Android-додаток "Му KNUTE" із зрозумілим інтерфейсом, що надає актуальну інформацію, релевантну для абітурієнтів, студентів та співробітників КНТЕУ.

Ключові слова: мобільна розробка, мобільний додаток, Android-додаток, освіта у смартфоні, додаток "Му KNUTE".

#### ABSTRACT

Explanatory note: 45 pages, 23 pictures, 27 additions, 10 sources.

The purpose of the work: development of a native mobile application on the Android platform to meet the needs of participants in the educational process in convenient access to organizational information that accompanies learning.

Object of research: use of mobile applications in the educational process.

Subject of research: development of Android application "My KNUTE".

Research methods: comparison, analysis, object-oriented programming, visualization.

In the process of work the information needs of students and staff of KNUTE for use in the project are revealed, the technical task for development of the native Android-application "My KNUTE" is made, based on the latest tendencies the interface and logic of the application are designed, conclusions and proposals for further improvement of the application are formed.

The result is a native Android application "My KNUTE" created with the modern tools (Android Studio development environment, XML markup language and Kotlin programming language) with a clear interface that provides up-to-date information relevant for entrants, students and employees of KNUTE.

**Keywords:** mobile development, mobile application, Android application, education in a smartphone, application "My KNUTE".

# **3MICT**

ВСТУП.		• • • • • • • • • • • • •				• • • • • • • • •	4
РОЗДІЛ	1. АНАЛІЗ	преді	METH	ЮЇ ОБЛАСТІ ТА ФО	РМУВА	АННЯ	ВИМОГ
до мое	ыльного	ДОДА	тку '	'MY KNUTE"	•••••	•••••	6
1.1. Місц	е мобільних	к додатк	ів в ос	вітньому процесі та осн	овні ви	моги д	о додатка
"My KNI	JTE"						6
1.2. Анал	піз ринку т	ехнолог	тій роз	вробки мобільних дода	тків: н	ативні	та крос-
платформ	ині рішення					•••••	9
1.3. Техн	ічне завдані	ня на ро	зробку	у мобільного додатку "М	Iy KNU	JTE"	14
1.4. Висн	овки до роз	ділу 1					17
РОЗДІЛ	2. ПРОЕК	ТУВАН	ня м	ЮБІЛЬНОГО ДОДАТ	ку д.	ІЯ СІ	істеми
ANDRO	ID ТА ВИБ	IP IHC	ТРУМ	ІЕНТАРІЮ ДЛЯ РОЗІ	обки	[	18
2.1. Житт	тєвий цикл н	нативно	го моб	ы́льного додатку для OC	Andro	id	
2.2. Cepe,	довище роз	робки А	ndroid	l Studio. Структура типо	вого п	роекту	<i></i> 22
2.3. Вибі	р мови прог	рамуван	ня на	гивного Android-додатк	y		25
2.4. Прое	ктування ін	терфейс	су моб	ільного додатку "Му KI	UTE".		
2.5. Висн	овки до роз	ділу 2					32
РОЗДІЛ	3. PO3	РОБКА	НА	ТИВНОГО ANDRO	ID-ДO	ДАТК	у "МҮ
KNUTE"	,						
3.1. Опис	: модулів мо	обільног	го дода	атку "My KNUTE"			
3.2. Реалі	зація інтеро	фейсу ко	ористу	вача			35
3.3. Шлях	хи вдоскона	лення н	ативно	ого Android-додатку "M	y KNU'	ГЕ"	41
3.4. Висн	овки до роз	лілу З					43
				KHTEV 12	1-23-0	5 <i>MP</i>	
				1011125 12	1 25 0.	<b>9.1711</b>	
Зм. Аркуш	№ докум.	Підпис	Дата	Моделювання нативного мобільного додатку для систем	Стадія и	Аркуш	Аркушів
Зав. каф.	Криворучко О.В.		20.10.21	Android "My KNUTE"	"Зміст	2	46
Керівник Гарант	Криворучко О.В. Токар В.В.		20.10.21 20.10.21		Факул	њтет ін	формаційних
Розробив	Грімов А.І.		20.10.21	Зміст		технол	погій,

2 курс, 23 група

ВИСНОВКИ ТА ПРОПОЗИЦІЇ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	.45
ДОДАТКИ	47

						Аркуш
					КНТЕУ 121-23-05.МР	2
Изм.	Аркуш	№ докум	Подпись	Дата		5

#### ВСТУП

Світова діджиталізація стрімко поширюється та впроваджується в усі сфери нашого життя, в тому числі в освітній процес. Крім того, у свою чергу й пандемія COVID-19 прискорила трансформацію освіти, особливо середньої та вищої, змусивши перейти на онлайн-форму навчання, активно залучати гаджети, швидко опановувати нові інструменти тощо.

Не дивлячись на різноманітність середовищ для проведення онлайнзанять, контрольних заходів та ін., поза увагою залишається потреба учасників освітнього процесу мати швидкий і зручний доступ до відомостей, що носять більш організаційний характер і є допоміжними до навчання, але, в той же час, є важливою складовою для досягнення високого ступеню задоволеності освітнім процесом. До таких, наприклад, відносяться: розклад дзвінків та занять, довідник студента, телефонний довідник підрозділів університету, склад факультетів та кафедр тощо. Саме це обумовило актуальність обраної теми випускного кваліфікаційного проекту.

Об'єкт дослідження: використання мобільних додатків в освітньому процесі.

Предмет дослідження: розробка Android-додатку "Му KNUTE".

Мета дослідження: розробка нативного мобільного додатку на платформі Android для задоволення потреб учасників освітнього процесу у зручному доступі до організаційної інформації, що супроводжує навчання.

					КНТЕУ 121-23-05.МР				
					Моделювання нативного	Стадія	Аркуш	Аркушів	
Зм.	Аркуш	№ докум.	Підпис	Дата	мобільного додатку для системи				
Зав. каф.		Криворучко О.В.		25.02.21	Android "My KNUTE"	В	4	46	
Керівні	ик	Криворучко О.В.		25.02.21					
Гарант		Токар В.В.		25.02.21	Remun	Факультет інформаційних технологій		формаційних логій.	
Розробив		Грімов А.I.		25.02.21	Demyn	2 курс, 2з група			

Для досягнення мети були поставлені наступні завдання:

- виявити інформаційні потреби студентів та співробітників КНТЕУ, що будуть покладені в основу проекту;
- скласти технічне завдання на розробку нативного Android-додатку "Му KNUTE";
- базуючись на останніх тенденціях спроектувати інтерфейс та логіку додатку;
- за допомогою сучасного інструментарію розробити Android-додаток
  "My KNUTE";
- сформувати висновки та пропозиції щодо подальшого розвитку розробленого додатку.

Методи і технології розробки: передбачається застосування методів порівняння, аналізу, об'єктно-орієнтованого програмування, візуалізації; технології – середовище розробки Android Studio, мова розмітки XML та мова програмування Kotlin.

Результат: повністю робочий Android-додаток із зрозумілим інтерфейсом, що надаватиме актуальну інформацію, релевантну для користувача.

						Аркуш
					КНТЕУ 121-23-05.МР	5
Изм.	Аркуш	№ докум	Подпись	Дата		5

#### РОЗДІЛ 1

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУВАННЯ ВИМОГ ДО МОБІЛЬНОГО ДОДАТКУ "МУ KNUTE"

# 1.1. Місце мобільних додатків в освітньому процесі та основні вимоги до додатка "Му KNUTE"

Мобільні технології – це широкий спектр цифрових і повністю портативних мобільних пристроїв (смартфонів, планшетних комп'ютерів, електронних книг тощо), що дозволяють здійснювати операції з отримання, обробки та поширення інформації.

Застосування Інтернет-технології започаткувало нову форму навчального процесу, яка отримала назву – електронне навчання (E-learning). Найбільшого поширення ця форма набула в галузі вищої освіти. E-learning покладено в основу дистанційного навчання, яке сьогодні динамічно розвивається, використовуючи різноманітні спеціальні програмно-інструментальні платформи.

Такий підхід став поштовхом до подальших змін у вищій освіті [1, с.62]: модернізації навчально-виховного процесу; введення нових підходів до подання теоретичного матеріалу, проведення практичних і лабораторних занять; розвитку індивідуальних методів роботи; зміни методів і прийомів роботи викладачів, студентів; персоналізації профілю вивчення (перебудови навчального процесу до потреб суб'єктів освіти); зміни методів оцінювання знань студентів та їх корекції.

					КНТЕУ 121-23-05.МР				
					Моделювання нативного	Стадія	Аркуш	Аркушів	
Зм.	Аркуш	№ докум.	Підпис	Дата	мобільного додатку для системи				
Зав. каф.		Криворучко О.В.		10.04.21	Android "My KNUTE"	P1	6	45	
Керівни	ік	Криворучко О.В.		10.04.21					
Гарант		Токар В.В.		10.04.21	Аналіз предметної області та формування вимог до мобільного	Факул	Факультет інформаційн технологій.		
Розробив		Грімов А.І.		10.04.21	додатку "Му KNUTE"	2 курс, 2з група			

Однією з активних форм E-learning є мобільне навчання (M-learning), яке все більше набуває популярності серед сучасної молоді. M-learning поширюється завдяки розвитку технології мобільного зв'язку, що базується на застосуванні мережі Інтернет, і в майбутньому може стати потужним засобом підвищення успішності навчання на всіх етапах становлення особистості – від загальноосвітньої до вищої освіти [2].

Використання мобільних додатків в освітньому процесі є безперечно перспективним напрямом розвитку освітньої галузі, тому варто звернути увагу на основні переваги та недоліки мобільного навчання.

Основними перевагами застосування технології мобільного навчання є:

1) інноваційність технології;

2) можливість використання переносних пристроїв в освітніх цілях;

3) можливість застосовувати технології в якості додаткового засобу навчання;

4) можливість завантажити необхідний теоретичний матеріал для вирішення математичних задач у зручний для студента час за допомогою мобільних додатків;

5) мобільне навчання допомагає адаптуватися до навчання в інформаційному просторі;

6) можливість до швидкого доступу до побудови графіків, таблиць, вирахування формул;

7) можливість одночасної взаємодії як з однією особою, так і з групою осіб.

До недоліків упровадження технології мобільного навчання можна віднести [2, 3]:

1) дефіцит якісного повнофункціонального освітнього контенту для мобільних пристроїв і засобів його розроблення;

2) недостатня «технічна» підготовка викладачів у створенні мобільних додатків;

						Аркуш
					КНТЕУ 121-23-05.МР	7
Изм.	Аркуш	№ докум	Подпись	Дата		1

3) обмежені розміри дисплеїв і ємність батарей мобільних пристроїв, що не завжди зручно для рішення складних рівнянь, математичних функцій, побудови їх графіків;

4) обмежений термін дії батареї, який необхідно враховувати при тривалому періоді часу виконання завдання;

5) несумісність деяких мобільних пристроїв з іншими мобільними додатками.

Останнім часом питання мобільного навчання стало предметом досліджень багатьох закордонних і вітчизняних науковців, однак, використання мобільних пристроїв у процесі навчання не обмежується лише комунікацією між учасниками освітнього процесу, обміном теоретичного чи практичного матеріалу, проведенням онлайн-занять, перевіркою виконаних завдань тощо. Слід розглянути можливість застосування мобільного пристрою у якості «помічника», що супроводитиме процес навчання, а також як засіб просування навчального закладу.

Так, наприклад, універсальний контент з актуальними відомостями про КНТЕУ, став би у нагоді як студентам, так і працівникам університету, а розміщення мобільного додатку у Play Market та App Store (офіційні магазини мобільних додатків від Google та Apple відповідно) може стати засобом розповсюдження інформації про КНТЕУ, тобто використовуватися у якості реклами.

Проаналізувавши офіційну сторінку КНТЕУ[4], що є джерелом актуальної та достовірної інформації про заклад, нами складено перелік універсальних корисних відомостей, швидкий доступ до яких було б доцільно реалізувати у мобільному додатку:

- Загальна інформація про КНТЕУ;
- Факультети, кафедри;
- Мапа території КНТЕУ;

						Аркуш
					КНТЕУ 121-23-05.МР	8
Изм.	Аркуш	№ докум	Подпись	Дата		0

- Телефонний довідник КНТЕУ;
- Розклад занять та екзаменаційної сесії;
- Довідник студента.

Отже, з метою забезпечення учасників освітнього процесу в КНТЕУ релевантною інформацією про заклад та організацію навчання в ньому, а також для підкріплення іміджу університету, у процесі розробки мобільного додатку буде враховано вищенаведений функціонал.

# **1.2.** Аналіз ринку технологій розробки мобільних додатків: нативні та крос-платформні рішення

Мобільний додаток – програма, встановлена на тій чи іншій платформі, володіє певним функціоналом, що дозволяє виконувати певні дії. Розрізняють нативні, веб і гібридні додатки.

Нативний додаток – це додаток побудований повністю на засадах технологій, що відносяться до тієї чи іншої операційної системи. Це може бути Android, iOS, Windows, Blackberry тошо. Для Android, нативні додатки, як правило, побудовані із використанням Java або Kotlin, в той час як для iOS, додаток може бути побудовано з використанням Objective-C або Swift.

Оскільки нативний додаток розроблений лише під конкретний пристрій та ОС, він може користуватися апаратним та програмним забезпеченням для конкретного пристрою. Нативні додатки можуть забезпечувати оптимізовану продуктивність та користуватися найновішими технологіями, такими як GPS, у порівнянні з веб-додатками або мобільними додатками, які є мультисистемними.

Також нативні додатки можуть отримати доступ до системи оповіщення пристрою, а також, в залежності від призначення нативного додатка, він може повністю або частково обходитися без наявності інтернет-з'єднання.

						Аркуш		
					КНТЕУ 121-23-05.МР	0		
Изм.	Аркуш	№ докум	Подпись	Дата		9		

Переваги нативних додатків включають[5]:

- швидкість роботи і продуктивність;
- високий ступінь безпеки;
- розширений інтерфейс;
- максимально можлива функціональність;
- здатність працювати без Інтернету;
- зручність для кінцевого користувача.

Недоліки нативних програм включають:

- декілька базових кодів, оскільки кожен пристрій має свою версію програми;
- вартість додаткової роботи з побудовою і керуванням коду для кожної платформи;
- час, витрачений на декілька версій для окремих платформ при кожному оновленні функції.

Приклади популярних нативних додатків: Shazam, Instagram.

*Веб-додатки* використовують стандартні веб-технології, такі як HTML5, JavaScript і CSS. Цей підхід дає можливість створити крос-платформні мобільні додатки, які працюють на декількох пристроях.

Насправді мобільні веб-додатки не є додатками, як такими. Запускаючи мобільні веб-додатки, користувач виконує всі ті дії, які він виконує при переході на будь-який веб-сайт, а також отримує можливість «встановити» їх на свій робочий стіл, створивши закладку сторінки веб-сайту.

Переваги:

- скорочення витрат і часу розробки;
- підтримка всіх платформ;
- такі додатки легше підтримувати в порівнянні з нативними.

						Аркуш
					КНТЕУ 121-23-05.МР	10
Изм.	Аркуш	№ докум	Подпись	Дата		10

Недоліки[5]:

- обов'язкове підключення до Інтернету;
- дуже обмежений у можливостях інтерфейс програми;
- неможливість відправити push-повідомлення;
- продуктивність і швидкість роботи;
- незадовільний рівень безпеки.

Приклади популярних веб-додатків: last.fm, Google Maps.

*Гібридні додатки* поєднують в собі деякі функції нативних і веб-додатків, а саме: мультиплатформність і можливість використання програмного забезпечення телефону або смартфону. Разом з тим вони мають у своєму розпорядженні опцію автономного оновлення інформації, а для їх роботи необхідне інтернет-підключення. Без наявності останнього веб-функції просто не працюють.

Гібридні додатки на сьогоднішній день є найбільш популярними, оскільки, розробка відбувається швидше і дешевше, ніж у випадку з нативними додатками, хоча оболонка і написана на «рідній» мові програмування, внутрішня частина програми може бути написана в тому чи іншому обсязі на HTML5. У такому випадку користувач, швидше за все, не помітить різницю між нативним додатком і гібридним.

Отже, варто розробляти гібридний додаток, якщо:

- є необхідність заощадити в бюджетному плані;
- потрібно створити відносно нескладне додаток з простою анімацією;

є завдання оперативної розробки програми як мінімум на 2 платформи.
 Переваги:

- не вимагають від користувачів підтвердження дозволів;
- крос-платформність;
- скорочення витрат і часу на розробку;

- опція автономного оновлення.

						Аркуш
					КНТЕУ 121-23-05.МР	11
Изм.	Аркуш	№ докум	Подпись	Дата		11

Недоліки[5]:

- некоректна робота за відсутності інтернет-з'єднання;
- середня швидкість роботи на тлі нативних;
- мінімалізм щодо візуальних елементів.

Приклади популярних гібридних додатків: HeartCamera, TripCase.

Для прийняття рішення щодо цільової мобільної операційної системи, під яку розроблятиметься мобільний додаток "Му KNUTE" у рамках даного проекту, було проаналізовано статистику по використанню мобільних ОС в Україні протягом останніх трьох років.

Наведені на рисунках 1.1 – 1.3 показники свідчать, що лідерами у рейтингу стабільно виступають Android (1 місце) та iOS (2 місце), при чому значне переважання частки користувачів Android, порівняно з iOS, за 2019-2021 р.р. збільшилося з 58,69% до 65,45%.





Зважаючи на вищевикладене, доцільним є розробити мобільний додаток "My KNUTE" на платформі OC Android із використанням технологій нативної розробки.

#### 1.3. Технічне завдання на розробку мобільного додатку "Му KNUTE"

#### 1.Загальні положення

1.1. Найменування програмного забезпечення

Найменування програми: мобільний додаток "Му KNUTE".

1.2. Планові терміни початку та закінчення робіт

Відповідно до поставленого завдання та врахувавши виконання усіх необхідних етапів розробки якісного програмного продукту (див. п.6), запланований термін виконання – з 17 квітня до 20 червня 2021 року.

1.3. Головний бенефіціар та потенційні користувачі програмного забезпечення

Головний бенефіціар відсутній. Програмний продукт не має комерційного характеру.

Потенційними користувачами є: студенти КНТЕУ, співробітники КНТЕУ, майбутні абітурієнти КНТЕУ.

#### 2. Мета та призначення створення програмного забезпечення

2.1. Призначення програмного забезпечення

Мобільний додаток "Му KNUTE" призначений для отримання актуальної інформації про КНТЕУ, організацію освітнього процесу та іншої корисної довідкової інформації про університет.

2.2. Мета створення програмного забезпечення

Метою створення додатку є надання можливості усім зацікавленим особам отримати швидкий доступ до найбільш затребуваних відомостей про КНТЕУ, не витрачаючи час на пошук відповідних даних на офіційному сайті, при цьому

						Аркуш
					КНТЕУ 121-23-05.МР	14
Изм.	Аркуш	№ докум	Подпись	Дата		14

використовуючи менший обсяг інтернет-трафіку, а за відсутності інтернетз'єднання – переглядати збережену версію контенту з попереднього сеансу.

# 3. Вимоги до системи, на якій здійснюватиметься запуск програмного забезпечення

Для коректної роботи мобільного додатку "Му KNUTE" рекомендовано використовувати смартфон на платформі Android версії 4.4 (KitKat), з наявним підключенням до мережі Інтернет (хоча б для першого завантаження контенту).

#### 4. Вимоги до програмного забезпечення

Головне меню мобільного додатку має містити наступні опції:

- Загальна інформація про КНТЕУ;
- Факультети, кафедри;
- Мапа території КНТЕУ;
- Телефонний довідник КНТЕУ;
- Розклад занять та екзаменаційної сесії;
- Довідник студента.

Після вибору необхідного розділу додатку, здійснюється перехід на екран відповідного розділу.

Екран «Загальна інформація про КНТЕУ» має текстовий вміст, ідентичний вмісту сторінки «Організаційна структура» офіційного сайту КНТЕУ.

Екран «Факультети, кафедри» відображає актуальний структурований перелік факультетів і кафедр КНТЕУ, що входять до їх складу. У разі натискання на один із пунктів структури (факультет/кафедра), здійснюється перехід на екран з відповідною інформацією. У разі, якщо користувач обрав факультет, наступний екран міститиме склад деканату та напрями підготовки, а якщо кафедру – викладацький склад.

Екран «Мапа території КНТЕУ» містить мапу, представлену на офіційному сайті у розділі «Контакти», із можливістю масштабування.

						Аркуш
					КНТЕУ 121-23-05.МР	15
Изм.	Аркуш	№ докум	Подпись	Дата		15

Екран «Телефонний довідник КНТЕУ» складається з переліку структурних підрозділів університету, при виборі яких відображатимуться відповідні контакти, завантажені з офіційного сайту КНТЕУ.

Екран «Розклад занять та екзаменаційної сесії» передбачає перехід до Розкладу дзвінків, Розкладу занять або Розкладу екзаменаційної сесії, після чого відображається відповідний розклад.

Екран «Довідник студента» відображає актуальний Довідник студента КНТЕУ, завантажений з офіційного сайту.

#### 5. Вимоги до програмної документації

Попередній склад програмної документації включає у себе:

- керівництво користувача;
- перелік посилань, за якими додаток завантажує ресурси для відображення.

#### 6. Етапи розробки програмного забезпечення

У ході виконання даного проекту буде застосована послідовна модель розробки програмного забезпечення, що складається з наступних етапів:

- збір та аналіз вимог до ПЗ;
- дизайн та архітектура;
- програмування;
- тестування;
- впровадження.

						Аркуш
					КНТЕУ 121-23-05.МР	16
Изм.	Аркуш	№ докум	Подпись	Дата		10

#### 1.4. Висновки до розділу 1

Перспективність мобільного навчання стимулює розвиток ринку програмних рішень, що покликані удосконалювати освітній процес, полегшуючи його та роблячи більш доступним.

За результатами аналізу сучасного ринку мобільних додатків та операційних систем, доцільною є розробка нативних додатків під платформу ОС Android. Таким чином, даний підхід буде застосовано у процесі розробки мобільного додатку "Му KNUTE", до якого складено технічне завдання.

						Аркуш
					КНТЕУ 121-23-05.МР	17
Изм.	Аркуш	№ докум	Подпись	Дата		1/

#### **РОЗДІЛ 2**

# ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СИСТЕМИ ANDROID ТА ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ РОЗРОБКИ

#### 2.1. Життєвий цикл нативного мобільного додатку для OC Android

Перш ніж приступати до розробки будь-якого програмного продукту, необхідно отримати чітке уявлення про особливості розробки та функціонування конкретного виду програмного забезпечення, у даному випадку – мобільних додатків на базі OC Android.

Однією з ключових особливостей Android-додатків є конкретно визначений життєвий цикл додатку, зображений на рисунку 2.1.



Основні методи життєвого циклу додатку:

- protected void onCreate()
- protected void onStart()
- protected void onRestart()
- protected void onResume()
- protected void onPause()
- protected void onStop()
- protected void onDestroy()

Метод *onCreate()* викликається під час створення або перезапуску активності (визначений у проекті екран відображення, з власним макетом візуального представлення елементів, а також набором супроводжуючих подій, описаних мовою програмування).

У цьому методі слід завантажувати інтерфейс користувача, посилатися на властивості класу, пов'язувати дані з елементами управління, створювати сервіси та потоки. Метод onCreate() приймає об'єкт Bundle, що містить стан інтерфейсу користувача, збережений за останнього виклику обробника onSaveInstanceState. Для відновлення графічного інтерфейсу в його попередньому стані необхідно задіяти цю змінну: всередині onCreate() або перевизначивши метод onRestoreInstanceState().

Операції ініціалізації, що вимагаюсь багато часу, слід виконувати у фоновому процесі, а не за допомогою метода onCreate(). В інакшому випадку можна отримати діалогове вікно ANR (Application Not Responding, додаток не відповідає).

За onCreate() завжди слідує виклик onStart(), але перед onStart() не обов'язково має бути onCreate(), оскільки onStart() може викликатися і для відновлення роботи призупиненого додатку (зупиняється він методом onStop()). Викликається безпосередньо перед тим, як активність стає видимою

						Аркуш
					КНТЕУ 121-23-05.МР	10
Изм.	Аркуш	№ докум	Подпись	Дата		19

користувачеві. Супроводжується викликом методу onResume(), якщо активність отримує передній план, або викликом методу onStop(), якщо стає прихованою.

Метод **onResume**() викликається після onStart(), навіть коли вікно працює в пріоритетному режимі і користувач може його спостерігати. У цей момент користувач взаємодіє із створеним вікном. Додаток отримує монопольні ресурси. Запускає відтворення анімації, аудіо та відео. Також може викликатися після onPause().

Немає необхідності перезавантажувати стан інтерфейсу користувача, оскільки ці функції покладені на обробники onCreate() та onRestoreInstanceState.

Коли користувач вирішує перейти до роботи із новим вікном, система здійснює виклик для вікна, що переривається, методу *onPause()*. По суті відбувається згортання активності. Зберігаються незафіксовані дані. Зупиняється відтворення відео, аудіо та анімації. Від onPause() можна перейти до виклику onResume() або onStop().

У той же час, не варто використовувати onPause() для збереження користувацьких змін (таких, як персональні дані, введені в форму) на постійне зберігання. Виключення допускається, якщо є впевненість, що користувачі очікують зміни, що будуть автоматично збережені (наприклад, при створенні електронного листа). Тим не менш, слід уникати інтенсивної роботи в onPause(), оскільки це може сповільнити перехід до наступної активності.

Коли активність призупинена, всі компоненти зберігаються у пам'яті та при відновленні немає необхідності повторно їх ініціалізувати.

Метод **onStop**() викликається, коли вікно стає невидимим для користувача. Це може відбутися у разі знищення активності, або при запуску іншої активності, що перекриває вікно поточної. Метод завжди супроводжує будь-який виклик методу onRestart(), якщо активність повертається, для взаємодії із користувачем, або методу onDestroy(), якщо активність знищується.

						Аркуш
					КНТЕУ 121-23-05.МР	20
Изм.	Аркуш	№ докум	Подпись	Дата		20

Коли активність зупиняється, об'єкти активності зберігаються у пам'яті та відновлюються, коли активність відновлює свою роботу. Не потрібно повторно ініціалізовувати компоненти, створені раніше. Крім того, система відслідковує поточний стан для кожного представлення, тому, якщо користувач введе текст у текстове поле, то його вміст буде збережено і не буде необхідності самостійно зберігати та відновлювати його.

У даному методі можна виконувати складні операції збереження даних: для призупинення складної анімації, потоків, відслідковування показів датчиків, запитів до GPS, таймерів, сервісів чи інших процесів, що потрібні виключно для оновлення інтерфейсу користувача. Немає сенсу споживати ресурси (такти центрального процесора або мережевий трафік) для оновлення інтерфейсу, коли його не видно на екрані. Доцільним є застосування методів onStart() або onRestart() для відновлення чи повторного запуску цих процесів, коли активність знову стане видимою.

За браку пам'яті система може знищити приховану активність, уникаючи метод onStop() з викликом методу onDestroy().

Якщо вікно повертається у пріоритетний режим після виклику onStop(), то в даному випадку викликається метод *onRestart()*. Тобто виклик здійснюється після того, як активність була зупинена і знову запущена користувачем. Завжди супроводжується викликом методу onStart().

Даний метод слід застосовувати для спеціальних дій, що повинні виконуватися тільки при повторному запуску активності в рамках «повноцінного» стану.

Метод *onDestroy()* викликається по завершенню роботи активності, під час виклику методу **finish()** або у випадку, коли система знищує цей екземпляр активності для вивільнення ресурсів. Ці два сценарії знищення можна визначити викликом методу isFinishing(). Це останній запит, який отримує активність від системи. Якщо визначене вікно знаходиться у верхній позиції в стеку, але

						Аркуш
					КНТЕУ 121-23-05.МР	21
Изм.	Аркуш	№ докум	Подпись	Дата		21

невидиме користувачу і система вирішує завершити дане вікно, викликається метод onDestroy(). У даному випадку метод видаляє усі статичні дані активності, віддає усі використані ресурси.

Оскільки всі необхідні операції по вивільненню ресурсів вже виконані у методі onStop(), то в даному методі можна для впевненості перевірити ще раз всі невивільнені ресурси.

На практиці частіше за все доводиться працювати над методами onCreate(), onResume() i onPause(). Метод onCreate() буде викликатися під час створення користувацького інтерфейсу для роботи з вікном. Даний метод дозволить зв'язати дані з компонентами і під'єднати обробники подій до компонентів користувацького інтеріейсу. За допомогою onPause() можна зберегти важливу інформацію в базі даних додатку. Це останній безпечний метод, що буде викликаний перед тим, як система завершить роботу додатка. Метод onDestroy() не обов'язково буде викликаний, тому не варто покладатися на цей метод при реалізації критичної логіки.

#### 2.2. Середовище розробки Android Studio. Структура типового проекту

Android Studio — це офіційне середовище розробки додатків для Android. По суті, це відома Java IDE IntelliJ IDEA з плагінами, що відрізняється від IDEA дрібницями, але ці дрібниці значно полегшують роботу розробникам додатків під Android.

Для Windows існує три дистрибутива Android Studio: два для 64-bit і один для 32-bit. Один з дистрибутивів (64-bit, рекомендований) включає інсталятор. Інші – представлені у вигляді архіва zip (і пакета Mac).

Інсталятор дистрибутива містить усе необхідне — IDE, Android Emulator, Android SDK. Відсутні в комплекті модулі інсталятор завантажить самостійно. На сторінці завантажування дистрибутива наведені вимоги до ПК, для встановлення Android Studio[8].

						Аркуш
					КНТЕУ 121-23-05.МР	$\gamma\gamma$
Изм.	Аркуш	№ докум	Подпись	Дата		

Функціонал програми:

- За допомогою Android-емулятора продивляються етапи роботи безпосередньо з ПК, не встановлюючи додаток на смартфон.
- Аналізатор АРК призначений для контролю кількості файлів та їх розміру.
- Доступний режим порівняння двох пакетів, наприклад, застарілої та оновленої версії.
- Редактор Макетів інструмент для налаштування інтерфейсу.
- Профілювання в режимі реального часу відображає, скільки ресурсів споживає додаток.
- Новий формат публікації додатків від Android Studio Bundle, за допомогою якого оптимізація проходить без зміни коду.
- До складу IDE Android Studio включений інструментарій SDK, що застосовується для розробки мобільного програмного забезпечення. Пакет дозволено завантажити архівом окремо для кожної OC.

Android-проект представлений 2 кореневими теками: app i Gradle Scripts, як показано на рисунку 2.2.



Тека арр включає у себе 3 підтеки:

- 1. Тека manifests містить файли конфігурацій або файли маніфесту додатка
- 2. В теці java знаходиться вихідний код додатка
- 3. Тека res містить файли ресурсів для роботи Android-додатка (зображення, стилі, розмірності для різноманітних пристроїв і т.д.)

Файл AndroidManifest.xml є одним із найважливіших в Android-проекті. У ньому міститься інформація про пакети додатка, компоненти типу Activity, Service тощо.

Файл AndroidManifest.xml виконує наступні задачі:

- Надає дозволи додатку на використання чи доступ до інших компонентів системи.
- 2. Визначає як будуть запускатися, наприклад, Activity.

Тека java містить вихідний код додатка. Класи можуть бути розташовані в різних пакетах, але обов'язково всередині теки java.

В теці res розташовані всі необхідні додатку ресурси, включаючи зображення, різні xml файли, анімації, звукові файли та багато інших. Всередині теки res вищеперераховані ресурси розподілені по своїх теках, що можна побачити на рисунку 2.3:

- Тека drawable містить файли із зображеннями, що будуть використані у додатку.
- Тека layout містить xml-файли, що використовуються для побудови користувацького інтерфейсу Android-додатку.
- В теці тіртар зберігаються виключно піктограми додатка. Будь-які інші drawable-елементи мають бути розміщені у своїй теці.
- Тека values призначена для зберігання тих xml-файлів, у яких визначені прості значення типу рядків, масивів, цілих чисел, розмірностей, кольорів і стилів.

						Аркуш
					КНТЕУ 121-23-05.МР	24
Изм.	Аркуш	№ докум	Подпись	Дата		24



Рис. 2.3. Структура теки res Android-проекту

Скрипти Gradle використовуються для автоматизації збірки проекту. Android Studio виконує збірку додатка у фоновому режимі без будь-якого втручання з боку розробника. Процес збірки здійснюється із використанням системи Gradle — інструментарію для автоматичної збірки за допомогою набору конфігураційних файлів. Gradle-скрипти написані мовою groove.

Слід додати, що середовище розробки Eclipse використовує файл project.properties для налаштування метаданих проекту. В Android Studio цим займається тека .idea. Це означає, що метадані конкретного проекту зберігаються в Android Studio. Тека .idea на рисунку відсутня, однак, якщо перейти до вкладки Project Files, тека з'явиться.

#### 2.3. Вибір мови програмування нативного Android-додатку

З моменту появи *Java* стала основною мовою для розробки мобільних додатків на Android, завдяки тому, що вона забезпечує крос-платформну підтримку. Крім того, додатки на Java легко перенести на різні операційні системи. Програми Java працюють за принципом «Написано один раз, запускається всюди» (WORA – Write Once Run Anywhere), тобто вони будуть працювати однаково на будь-якому сумісному з Java пристрої без необхідності

						Аркуш
					КНТЕУ 121-23-05.МР	25
Изм.	Аркуш	№ докум	Подпись	Дата		23

зміни коду. І хоча Java – відносно стара мова програмування, вона зберігає популярність.

*Kotlin* – це новітня мова програмування, повністю сумісна з Java. Ці дві мови взаємозамінні. Минулого року Google назвав Kotlin «основною мовою для розробки додатків на Android».

За допомогою порівняння нижче з'ясуємо, завдяки чому мова Kotlin офіційно отримала першість і поступово витісняє Java в Android-проектах.

Ці дві мови програмування можна порівняти за багатьма ознаками[9], тож ми наведемо найбільш значущі, на наш погляд, переваги Kotlin порівняно із Java.

Стислість коду. Порівняння класу Java з еквівалентним класом Kotlin демонструє лаконічність коду Kotlin. Для тієї ж операції, що виконується в класі Java, клас Kotlin вимагає менше коду.

Наприклад, конкретний сегмент, де Kotlin може значно скоротити загальний обсяг стандартного коду, - це findViewByIds.

Класи даних. У повнорозмірних проектах зазвичай є кілька класів, призначених виключно для зберігання даних. Хоча ці класи практично не мають функціональності, розробнику необхідно написати багато стандартного коду Java.

Зазвичай розробник повинен визначити конструктор і кілька полів для зберігання даних, функції «геттери» та «сеттери» для кожного з полів, а також функції equals(), hashCode() та toString().

Kotlin має дуже простий спосіб створення таких класів. Розробнику достатньо включити лише ключове слово data у визначення класу, і все - компілятор сам подбає про все.

Сопрограми. Процеси, що інтенсивно завантажують процесор і мережеве введення-виведення, зазвичай використовують тривалі операції. Потік виклику блокується до завершення всієї операції. Оскільки Android є однопоточним за

						Аркуш
					КНТЕУ 121-23-05.МР	26
Изм.	Аркуш	№ докум	Подпись	Дата		20

замовчуванням, інтерфейс користувача повністю блокується, як тільки блокується основний потік.

Традиційне вирішення цієї проблеми Java - створити фоновий потік для тривалої або інтенсивної роботи. Однак керування кількома потоками призводить до збільшення складності, а також до помилок коду.

Kotlin дозволяє створювати додаткові потоки. Тим не менш, є найкращий спосіб керування інтенсивними операціями в Kotlin, відомий як сопрограми або корутини (coroutines). Код з використанням корутин не лише зрозумілий, а й лаконічний. Більше того, корутини дозволяють створювати елегантні додаткові стилі асинхронної неблокуючої розробки, такі як async/await.

Розумні приведення. Перш ніж об'єкт може бути наведений у Java, обов'язково потрібно перевірити тип. Це також правильно в сценаріях, де явно необхідно наводити об'єкт.

На відміну від Java, Kotlin має функцію розумного приведення, що автоматично обробляє такі надлишкові приведення. Вам не потрібно виконувати приведення всередині оператора, якщо він вже перевірений оператором іs в Kotlin.

Вбудована підтримка делегування. Делегування являє собою процес, в якому об'єкт, що приймає, делегує свої операції другому об'єкту делегата. Kotlin підтримує шаблон проектування «композиція поверх наслідування» за допомогою делегування першого класу, також відомий як неявне делегування.

Делегування Kotlin є альтернативою наслідування. Цей механізм дозволяє використовувати множинне наслідування. Крім того, делеговані властивості Kotlin запобігають дублюванню коду.

Саме вищеперераховані властивості мови програмування Kotlin й стали передумовою вибору її для розробки Android-додатку "Му KNUTE".

						Аркуш
					КНТЕУ 121-23-05.МР	27
Изм.	Аркуш	№ докум	Подпись	Дата		21

#### 2.4. Проектування інтерфейсу мобільного додатку "Му KNUTE"

Відповідно до технічного завдання додаток міститиме декілька екранів, що взаємодіятимуть одне з одним за схемою, наведеною на рисунку 2.4.



Рис. 2.4. Схема взаємодії екранів у додатку «Му KNUTE»

Для реалізації наведеної взаємодії необхідно наділити екрани додатка відповідними інтерактивними елементами, що певним чином реагуватимуть на дії користувача додатку.

На рисунках 2.5 – 2.12 показано, які елементи передбачаються для кожного типу екрану додатка, та їх призначення.











перехід до Екрану 5.1 або 5.2.1

перехід до Екрану 5.1 або 5.2.1

перехід до Екрану 5.1 або 5.2.1

Рис. 2.10. Макет екранів «Вибір типу розкладу» та «Вибір форми навчання»



Рис. 2.11. Макет екрану «Розклад дзвінків»

						Аркуш
					КНТЕУ 121-23-05.МР	20
Изм.	Аркуш	№ докум	Подпись	Дата		50
Вкладка 1	Вкладка 2					
-----------------------	-------------------------					
Текстови табличном	й вміст у 1у форматі					

Рис. 2.12. Макет екрану «Розклад»

У наступному розділі розглянемо детальніше особливості реалізації інтерактивності наведених елементів, а також кінцевий дизайн візуального представлення вказаних екранів.

						Аркуш
					КНТЕУ 121-23-05.МР	31
Изм.	Аркуш	№ докум	Подпись	Дата		51

# 2.5. Висновки до розділу 2

Розробка нативного Android-додатку вимагає від розробника не тільки знання необхідної мови програмування, а й, крім цього, глибокого розуміння життєвого циклу мобільних додатків на платформі Android, що надає змогу правильно розташувати логіку програми на відповідних етапах для ефективного використання ресурсів смартфону та стабільної роботи додатку.

Також, слід зазначити, що значно полегшує процес розробки нативних Android-додатків знання офіційного інструментарію від Google – Android Studio, оскільки він містить усе необхідне в одному програмному продукті.

						Аркуш
					КНТЕУ 121-23-05.МР	32
Изм.	Аркуш	№ докум	Подпись	Дата		32

# РОЗДІЛ З

# РОЗРОБКА НАТИВНОГО ANDROID-ДОДАТКУ "МУ KNUTE"

# 3.1. Опис модулів мобільного додатку "Му KNUTE"

У табл. 3.1 представлений набір класів, у яких описана логіка поведінки розробленого Android-додатку "Му KNUTE".

Таблиця 3.1

	,,,, <i>j j</i>
Назва класу	Призначення класу
MainActivity (Додаток А)	Визначає поведінку Головного екрану, в т.ч. дії при
	натисканні кнопок
HtmlTextActivity	Описує алгоритми представлення інформації у
(Додаток Б)	форматі html, в залежності від запиту на Головному
	екрані (Загальна інформація/ Розклад дзвінків/
	Довідник студента)
DepsActivity (Додаток В)	Визначає загальну поведінку екрану Факультети
DepartmentRecyclerAdapter	Містить алгоритми створення вибірки факультетів для
(Додаток Г)	відображення на екрані у вигляді списку
DepsActivity2 (Додаток Д)	Визначає загальну поведінку екрану Кафедри
DepartmentRecyclerAdapter2	Містить алгоритми створення вибірки кафедр
(Додаток Е)	конкретного факультету для відображення на екрані у
	вигляді списку
МарАсtivity (Додаток Ж)	Визначає поведінку екрану Мапа КНТЕУ
	•
	КНТЕУ 121-23-05.МР

Перелік програмних класів нативного Android-додатку "Му KNUTE"

					KHTEV 121	-23-03	5. <i>MP</i>	
					Моделювання нативного	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата	мобільного додатку для системи			
Зав. каф	).	Криворучко О.В.		15.09.21	Android "My KNUTE"	P3	33	46
Керівни	к	Криворучко О.В.		15.09.21				
Гарант		Токар В.В		15.09.21	Розробка нативного android-	Факул	ьтет ін технол	формаційних 10гій.
Розроби	IB	Грімов А.І.		15.09.21	додатку "Му КNUTE"		2 курс, 2	з група

Назва класу	Призначення класу
ContactsActivity (Додаток И)	Визначає загальну поведінку екрану Телефонний
	довідник
ContactsRecyclerAdapter	Містить алгоритми для відображення на екрані у
(Додаток К)	вигляді табличних рядків назв підрозділів та їх
	контактів
TimetableChooseActivity	Визначає загальну поведінку екрану Вибору типу
(Додаток Л)	розкладу (Розклад дзвінків/ занять/ екзаменів)
EduFormChooseActivity	Визначає загальну поведінку екрану Вибору форми
(Додаток М)	навчання для відображення розкладу занять
EduFormChooseActivity2	Визначає загальну поведінку екрану Вибору форми
(Додаток Н)	навчання для відображення розкладу екзаменаційної
	ceciï
TimetableActivity	Визначає поведінку екрану перегляду Розкладу
(Додаток П)	занять/ екзаменів
SectionsPagerAdapter.kt	Специфічний клас, що необхідний для коректної
(Додаток Р)	роботи з екранними вкладками
BachelorFragment.kt	Містить алгоритми створення вибірки посилань на
(Додаток С)	розклади ОС Бакалавр для відображення на екрані у
	вигляді таблиці
MasterFragment.kt	Містить алгоритми створення вибірки посилань на
(Додаток Т)	розклади ОС Магістр для відображення на екрані у
	вигляді таблиці

Варто нагадати, що мобільний додаток "Му KNUTE" є орієнтованим на роботу з користувачем, а, отже, крім відповідної логіки роботи додатка не менш важливе значення відіграє реалізація інтерфейсу, при взаємодії з яким виконуватимуться функції, передбачені завданням. Про це й піде мова у наступному підрозділі.

						Аркуш
					КНТЕУ 121-23-05.МР	34
Изм.	Аркуш	№ докум	Подпись	Дата		54

# 3.2. Реалізація інтерфейсу користувача

Одразу після запуску Android-додатку "Му KNUTE" користувач потрапляє на Головний екран із вибором різних розділів додатку, як показано на рис.3.1

З кодом файлу макету Головного екрану можна ознайомитися у дод. У.



Рис. 3.1. Реалізація макету Головного екрану

Візуальне представлення таких розділів як «Загальна інформація про КНТЕУ» і «Довідник студента» реалізуються за допомогою макета повноекранного веб-елемента (див. дод. Ф). Зазначені розділи у наповненому вигляді зображені на рисунках 3.2 та 3.3 відповідно.

Розділ «Мапа території» реалізований завдяки поєднанню елемента зображення та тексту (див. дод. Х), як показано на рисунку 3.4.

						Аркуш
					КНТЕУ 121-23-05.МР	35
Изм.	Аркуш	№ докум	Подпись	Дата		55

Розділ «Телефонний довідник» базується на використанні двох файлів макетів: екрану в цілому (див. дод. Ц.1) та повторюваного елементу (табличного рядка) (див. дод. Ц.2). Реалізація представлення наведена на рисунку 3.5.

Аналогічно, на застосуванні декількох макетів будується відображення розділів «Факультети, кафедри» та «Розклад занять та екзаменів».

Однак, маємо різні макети для усього екрану «Факультети» (див. дод. Ш.1), «Кафедри» (див. дод. Ш.2) за однакового макета їх повторюваного елемента (див. дод. Ш.3).

#### Про університет

1:09

КНТЕУ – один із найавторитетніших закладів вищої освіти України, атестований за найвищим, IV рівнем акредитації. Його історія бере початок від 1946 р. Указом Президента України у 2000 р. університету надано статус національного. У 2006 р. КНТЕУ приєднався до Великої Хартії університетів.

Управління діяльністю університету здійснюється на принципах автономії і самоврядування, демократизації прийняття рішень. Система управління якістю КНТЕУ сертифікована за міжнародними стандартами.

Університет — лідер у реформуванні вищої освіти України, першим у країні здійснив суттєве оновлення змісту освіти, впровадження новітніх навчальних технологій на базі програмних продуктів та глобальних інформаційних мереж. У закладі вищої освіти успішно розвивається інтегрована ступенева освіта, приділено увагу міжнародній мобільності студентів і викладачів.

Створено сучасний бібліотечний комплекс з електронним обслуговуванням читачів, доступом до повнотекстових електронних ресурсів міжнародних баз даних.

В університеті навчається більше 35 тис. студентів. Функціонують 6 факультетів: міжнародної торгівлі та права; економіки, менеджменту та психології, фінансів та обліку; інформаційних технологій; ресторанно-готельного та туристичного бізнесу;

Рис. 3.2. Реалізація макету екрану Загальна інформація про КНТЕУ

						Аркуш
					КНТЕУ 121-23-05.МР	36
Изм.	Аркуш	№ докум	Подпись	Дата		50



1:09	
Контакти підрозділ	ів
Телефонна довідкова служба	+380 44 531-31-73
Інформаційно-консультаці йний центр приймальної комісії	+380 44 531-48-88 pk@knute.edu.ua
Відділ організації та контролю діловодства	+380 44 531-47-41 knute@knute.edu.ua
Центр укладання договорів	+380 44 531-48-50, 531-48-51
Центр підготовки до ЗНО	+380 044-531-48-97 531-49-91 531-49-26
Підготовче відділення для іноземців та осіб без громадянства	+380 44 531-49-23
Відділ міжнародних зв'язків	+380 44 531-47-14, 531-47-44, 531-63-45 interdept@knute.edu.ua
Відділ організаційно-виховної роботи та інформаційного забезпечення	+380 44 531-49-48 531-31-06 info@knute.edu.ua
Бібліотека	+380 44 531-47-86 libr@knute.edu.ua
Центральна бухгалтерія	+380 44 531-47-55 buh@knute.edu.ua
Планово-фінансовий відділ (сектор контрактів, оплата за навчання)	+380 44 531-49-84
Відділ кадрів	+380 44 531-49-37 staff@knute.edu.ua
Юридичний відділ	+380 44 531-48-46

Рис. 3.5. Реалізація макету екрану Телефонний довідник

Реалізація розділу «Факультети, кафедри» зображена у вигляді пари екранів – факультету та відповідного йому переліку кафедр на рисунку 3.6.

Розділ «Розклад занять та екзаменів» задіяв найбільшу кількість макетів екранів. До них входять: екран Вибору типу розкладу (див. дод. Щ.1), екран Вибору форми навчання (див. дод. Щ.2), екран безпосередньо Розкладу з вкладками (див. дод. Щ.3) та макет вмісту кожної вкладки (див. дод. Щ.4).

Реалізація інтерфейсу перегляду розкладу зображена на рисунках 3.7-3.8 для різних типів розкладу відповідно.

Ознайомитися з посиланнями, що були застосовані у процесі розробки додатка, можна у дод. Ю відповідно.

						Аркуш
					КНТЕУ 121-23-05.МР	38
Изм.	Аркуш	№ докум	Подпись	Дата		50



Рис. 3.6. Реалізація макету екранів Факультети та Кафедри

Розилал			
Гозклад	Розклад	дзвінків	
	Орг у Київсько	анізація академічного (навч му національному торговел університеті	чального) дня льно-економічної
	Пара	Початок-закінчення	Перерва, хв.
	1	8:20 - 9:40	25
	2	10:05 - 11:25	40
РОЗКЛАД ДЗВІНКІВ	3	12:05 - 13:25	25
	4	13:50 - 15:10	15
	5	15:25 - 16:45	15
	6	17:00 - 18:20	10
	7	18:30 - 19:50	
розклад екзаменів			
РОЗКЛАД ЕКЗАМЕНІВ			
Розклад екзаменів Рис. 3.7. Реалізація інтеро	фейсу	Розклад дзв	інків
Розклад екзаменів Рис. 3.7. Реалізація інтеро	фейсу	Розклад дзв	інків
Розклад екзаменів Рис. 3.7. Реалізація інтеро	фейсу КНТF	Розклад дзв СУ 121-23-6	інків



Рис. 3.8. Реалізація інтерфейсу Розклад занять/екзаменів

Таким чином, було реалізовано користувацький інтерфейс додатку "Му KNUTE" із дотриманням останніх тенденцій розробки, рекомендованих Google.

# 3.3. Шляхи вдосконалення нативного Android-додатку "Му KNUTE"

У зв'язку з тим, що студенти КНТЕУ вже користуються хмарною системою Office 365 для спілкування у групі з викладачами/одногрупниками, відвідування онлайн-занять, виконання контрольних заходів тощо, нагромадження великої кількості різноманітних додатків для використання в освітньому процесі може перетворити зручність мобільної освіти на необхідного надлишковість інструментарію, спричинити плутанину i психологічний дискомфорт в учасника освітнього процесу.

1						
						Аркуш
					КНТЕУ 121-23-05.МР	40
Изм.	Аркуш	№ докум	Подпись	Дата		40

Переважна більшість студентів, що користується сервісами Office 365, має на своєму смартфоні встановлені, як мінімум, 2 додатки: Microsoft Outlook та Microsoft Teams.

З огляду на вищенаведене, перспективним напрямком розвитку додатку "My KNUTE" є інтеграція найбільш часто використовуваних Microsoft-cepвiciв, що дозволить звільнити смартфони учасників освітнього процесу від надлишкового споживання ресурсів і позбавить необхідності окремо встановлювати великі додатки при незначному їх використанні.

						Аркуш
					КНТЕУ 121-23-05.МР	41
Изм.	Аркуш	№ докум	Подпись	Дата		41

## 3.4. Висновки до розділу 3

Відповідно до вимог технічного завдання було виконано розробку нативного Android-додатку "Му KNUTE". У процесі розробки були опрацьовані різні типи макетів екранів, елементи інтерфейсу користувача, досліджені особливості реалізації кожного з них.

Розробка програмного коду дозволила на практиці впевнитися у перевагах мови програмування Kotlin порівняно з Java. Найкращі практики були застосовані у проекті, в т.ч. виконання окремих операцій в допоміжному потоці.

З огляду на тенденції діджиталізації в освіті та потреби безпосередньо студентів КНТЕУ, було сформовано рекомендований перелік розширення функціоналу додатку "Му KNUTE", що дозволить ще більше інтегрувати його в освітній процес.

	1					
						Аркуш
					КНТЕУ 121-23-05.МР	42
Изм.	Аркуш	№ докум	Подпись	Дата		42

# ВИСНОВКИ І ПРОПОЗИЦІЇ

У процесі роботи над випускним кваліфікаційним проектом було виконано поставлені завдання, згідно мети роботи, а саме:

- 1. Сформовано функціонал додатку "Му KNUTE" згідно з інформаційними потребами учасників освітнього процесу в КНТЕУ загальна інформація про університет, структура факультетів/кафедр, мапа території, розклад дзвінків, занять та екзаменів, телефонний довідник по підрозділах, довідник студента.
- 2. Відповідно до поставленого технічного завдання, розроблено нативний Android-додаток "Му KNUTE" за допомогою IDE Android Studio, мови розмітки макетів представлення XML та мови програмування Kotlin, оскільки даний набір інструментарію відповідає останнім тенденціям мобільної розробки та є рекомендованим Google.
- 3. Як один із перспективних напрямків розвитку додатку "Му KNUTE", запропоновано інтегрувати Microsoft-сервіси, застосовувані у навчанні, щоб мінімізувати кількість окремих додатків.

Таким чином, були набуті навички сучасної Android-розробки та успішно застосовані у процесі виконання випускного кваліфікаційного проекту.

					КНТЕУ 121-23-05.МР					
					Моделювання нативного	Стадія	Аркуш	Аркушів		
Зм.	Аркуш	№ докум.	Підпис	Дата	мобільного додатку для системи	ВΠ	43	46		
Зав. каф	þ.	Криворучко О.В.		25.10.21	Android "My KNUTE"					
Керівни	ік	Криворучко О.В.		25.10.21						
Гарант Розробив		Токар В.В.		25.10.21	Висноеки та пропозиції	Факультет інформаційних технологій				
		Грімов А.І.		25.10.21	Бисповки та пропозици	2 курс, 2з група				

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Лубіна Є. Мобільне навчання у дидактиці вищої школи / Єва Лубіна // Вісник Львівського ун-ту. Серія: Педагогіка. – 2009. – Вип. 25. – Ч. 2. – С.61-66

 Пэйн Н. 10 элементов мобильного обучения [Електронний ресурс] / Найджел Пейн // Дистанционное обучение: информационный портал. – Режим доступа:

<u>http://distancelearning.ru/db/el/C89AA03833448937C32577660010ACF1/doc.htm</u> <u>1</u> – Назва з екрану.

3. Бугайчук К. Л. Мобільне навчання: сутність і моделі впровадження в навчальний процес вищих навчальних закладів МВС України [Електронний ресурс] / К. Л. Бугайчук. // Інформ. технології і засоби навчання : [електрон. журн.]. - 2012. - № 1.

4. Офіційний сайт Київського національного торговельно-економічного університету [Електронний ресурс]. – Режим доступу: <u>https://knute.edu.ua</u>

5. Типи мобільних додатків [Електронний ресурс]. – Режим доступу: <u>https://training.qatestlab.com/blog/technical-articles/types-of-mobile-applications/</u>

6. Statcounter [Електронний ресурс]. – Режим доступу: <u>https://statcounter.com</u>

7. Android Developers [Електронний ресурс]. – Режим доступу: <u>https://developer.android.com</u>

8. Download Android Studio [Електронний ресурс]. – Режим доступу: <u>https://developer.android.com/studio</u>

					КНТЕУ 121-23-05.МР					
					Моделювання нативного	Стадія	Аркуш	Аркушів		
Зм.	Аркуш	№ докум.	Підпис	Дата	мобільного додатку для системи					
Зав. каф.		Криворучко О.В.		25.02.21	Android "My KNUTE"	КК	44	46		
Керівни	ік	Криворучко О.В.		25.02.21						
Гарант Розробив		Токар В.В.		25.02.21	Список викопистаних джерел	Факультет інформаційних технологій.				
		Грімов А.І.		25.02.21	enneon ounopuemunus oncepen	2 курс, 2з група				

9. KOTLIN VS JAVA: ЧТО ЛУЧШЕ ДЛЯ ANDROID-PA3PAБOTКИ? [Електронний pecypc]. – Режим доступу: <u>https://itvdn.com/ru/blog/article/kotlin-vs-java</u>

 Грімов А. Моделювання нативного мобільного додатку / А. Грімов // Software engineering комп'ютерних систем : зб. наук. ст. студ. / відп. ред.
 В.Я.Рассамакін. – Київ : Київ. нац. торг. – екон. ун-т, 2021. – С. 45-49.

						Аркуш
					КНТЕУ 121-23-05.МР	15
Изм.	Аркуш	№ докум	Подпись	Дата		43

# КЕРІВНИЦТВО КОРИСТУВАЧА

Головне меню мобільного додатку містить наступні опції:

- Загальна інформація про КНТЕУ;
- Факультети, кафедри;
- Мапа території КНТЕУ;
- Телефонний довідник КНТЕУ;
- Розклад занять та екзаменаційної сесії;
- Довідник студента.

Після вибору необхідного розділу додатку, здійснюється перехід на екран відповідного розділу.

Екран «Загальна інформація про КНТЕУ» має текстовий вміст, ідентичний вмісту сторінки «Організаційна структура» офіційного сайту КНТЕУ.

Екран «Факультети, кафедри» відображає актуальний структурований перелік факультетів і кафедр КНТЕУ, що входять до їх складу. У разі натискання на один із пунктів структури (факультет/кафедра), здійснюється перехід на відповідну сторінку сайту.

Екран «Мапа території КНТЕУ» містить мапу, представлену на офіційному сайті у розділі «Контакти» та уточнення щодо адрес навчальних корпусів та гуртожитків.

Екран «Телефонний довідник КНТЕУ» складається з переліку структурних підрозділів університету та їх робочі номери телефонів, а також електронні адреси.

Екран «Розклад занять та екзаменаційної сесії» передбачає перехід до Розкладу дзвінків, Розкладу занять або Розкладу екзаменаційної сесії, після чого відображається розклад відповідно до обраної форми навчання.

Екран «Довідник студента» відображає актуальний Довідник студента КНТЕУ.

					КНТЕУ 121	-23-05	5.MP		
					Моделювання нативного	Стадія	Аркуш	Аркушів	
Зм.	Аркуш	№ докум.	Підпис	Дата	мобільного додатку для системи	КК	46	46	
Зав. ка	þ.	Криворучко О.В.		21.10.21	Android "My KNUTE"				
Керівн	ик	Криворучко О.В.		21.10.21					
Гарант		Токар В.В.		21.10.21	Факультет і Список використацих джарал техн		ьтет ін технол	формаційних 10гій.	
Розроб	ИВ	Грімов А.І.		21.10.21	Список бикористиних бысерея	2 курс, 2з група			

# додатки

#### Додаток А

## MainActivity.kt

```
package com.example.myknute
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
class MainActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
     super.onCreate(savedInstanceState)
     setContentView(R.layout.activity_main)
     val intent = Intent(this, HtmlTextActivity::class.java)
     findViewById<Button>(R.id.menu_button_info).setOnClickListener {
       intent.putExtra("ACTION_TYPE", "info")
       startActivity(intent)
     }
     findViewById<Button>(R.id.menu_button_deps).setOnClickListener {
       startActivity(Intent(this, DepsActivity::class.java))
     }
     findViewById<Button>(R.id.menu_button_map).setOnClickListener {
       startActivity(Intent(this, MapActivity::class.java))
     }
     findViewById<Button>(R.id.menu_button_contacts).setOnClickListener {
       startActivity(Intent(this, ContactsActivity::class.java))
     }
     findViewById<Button>(R.id.menu_button_timetable).setOnClickListener {
       startActivity(Intent(this, TimetableChooseActivity::class.java))
     }
     findViewById<Button>(R.id.menu_button_manual).setOnClickListener {
       intent.putExtra("ACTION_TYPE", "manual")
       startActivity(intent)
     }
  }
}
```

## HtmlTextActivity.kt

package com.example.myknute

import android.os.Bundle

import android.webkit.WebView

import androidx.appcompat.app.AppCompatActivity

import kotlinx.coroutines.Dispatchers

import kotlinx.coroutines.runBlocking

import kotlinx.coroutines.withContext

import org.jsoup.Jsoup

import org.jsoup.nodes.Document

import org.jsoup.select.Elements

```
class HtmlTextActivity : AppCompatActivity() {
```

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate (savedInstanceState)

 $setContentView(R.layout.activity\_text)$ 

```
val typeAct: String = intent.getStringExtra("ACTION_TYPE").toString()
var url = ""
```

```
if (typeAct == "info") {
    url = "https://knute.edu.ua/blog/read/?pid=5&uk"
```

```
setTitle("Про університет")
```

#### }

```
if (typeAct == "manual") {
```

```
url = "https://knute.edu.ua/blog/read/?pid=1086&uk"
```

```
setTitle("Довідник студента")
```

#### }

```
if (typeAct == "calls") {
```

```
url = "https://knute.edu.ua/file/MjIxNw==/fff867341b9ecbae68301a3b3ea29598.pdf"
setTitle("Розклад дзвінків")
```

```
if (typeAct != "calls") {
    val result: Elements = connectJsoup(url).select("#timeline > div > small")
    result.select("h1,hr").remove()
    findViewById<WebView>(R.id.htmlText).loadData(
    result.html(),
```

```
"text/html; charset=UTF-8",
```

```
null
    )
    }
    else {
    findViewById<WebView>(R.id.htmlText).loadUrl("https://docs.google.com/gview?embedded=true&url=" +
url)
    }
    private fun connectJsoup (url: String) : Document = runBlocking {
    withContext(Dispatchers.IO) {Jsoup.connect(url).get()}
    }
}
```

#### DepsActivity.kt

package com.example.myknute

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import androidx.recyclerview.widget.LinearLayoutManager

import androidx.recyclerview.widget.RecyclerView

import kotlinx.coroutines.Dispatchers

import kotlinx.coroutines.runBlocking

import kotlinx.coroutines.withContext

import org.jsoup.Jsoup

import org.jsoup.nodes.Document

import org.jsoup.select.Elements

class DepsActivity : AppCompatActivity() {

var facList = mutableListOf<String>()
var depMap = mutableMapOf<String, MutableList<String>>()
var depLinksMap = mutableMapOf<String, String>()

override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.activity\_deps)

setTitle("Перелік факультетів")

val result: Elements = connectJsoup("https://knute.edu.ua").select("ul.nav > li.dropdown:nth-child(2) > ul.dropdown-menu > li > a")

println(result.text()) lateinit var fac: String lateinit var dep: String

lateinit var link: String

```
for (line in result) {
```

if (line.ownText().isEmpty()) {
 fac = line.select("span").text()
 link = line.attr("href")
 depLinksMap.put(fac, link)
 facList.add(fac)

```
depMap.put(fac, mutableListOf<String>())
}
else {
    dep = line.text()
    link = line.attr("href")
    depLinksMap.put(dep, link)
    depMap[fac]?.add(dep)
    }
}
val recyclerView: RecyclerView = findViewById(R.id.recyclerDepView)
```

recyclerView.layoutManager = LinearLayoutManager(this) recyclerView.adapter = DepartmentRecyclerAdapter(facList, depMap, depLinksMap)

# } private fun connectJsoup (url: String) : Document = runBlocking { withContext(Dispatchers.IO) { Jsoup.connect(url).get()} }

## DepartmentRecyclerAdapter.kt

package com.example.myknute

import android.app.PendingIntent.getActivity
import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import androidx.core.content.ContextCompat.startActivity
import androidx.recyclerview.widget.RecyclerView
import java.security.AccessController.getContext

class DepartmentRecyclerAdapter(private val faculties: List<String>, private val allDeps: Map<String, List<String>>, private val allLinks: Map<String, String>) : RecyclerView.Adapter<DepartmentRecyclerAdapter.MyViewHolder>() {

```
class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView){
```

```
var nameButton: Button? = null
```

#### init {

nameButton = itemView.findViewById(R.id.button\_department)

```
}
```

}

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {
    val itemView =
    LayoutInflater.from(parent.context)
```

.inflate(R.layout.department\_item, parent, false)

return DepartmentRecyclerAdapter.MyViewHolder(itemView)

```
}
```

override fun onBindViewHolder(holder: MyViewHolder, position: Int) {

```
holder.nameButton?.text = faculties[position]
```

```
holder.nameButton?.setOnClickListener {
  val facName = faculties[position]
  var facLink: String = ""
  lateinit var depList: List<String>
  val linkList = mutableListOf<String>()
  var arrDepList = ArrayList<String>()
```

var arrLinkList = ArrayList<String>()

```
allDeps.forEach { (k, v) -> if (k == facName) depList = v }
allLinks.forEach { (k, v) -> if (k == facName) facLink = v }
depList.forEach { allLinks.forEach { (k, v) -> if (k == it) linkList.add(v) } }
```

arrDepList.addAll(depList) arrLinkList.addAll(linkList)

val intent = Intent(holder.nameButton?.context, DepsActivity2::class.java)
intent.putExtra("FACULTY\_NAME", facName)
intent.putExtra("FACULTY\_LINK", facLink)
intent.putStringArrayListExtra("DEPARTMENT\_NAMES", arrDepList)
intent.putStringArrayListExtra("DEPARTMENT\_LINKS", arrLinkList)
holder.nameButton?.context?.startActivity(intent)

```
}
```

```
override fun getItemCount(): Int {
  return faculties.size
}
```

## DepsActivity2.kt

package com.example.myknute

import android.content.Intent

import android.content.Intent.ACTION\_VIEW

import android.net.Uri

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

 $import\ and roid.widget.TextView$ 

 $import\ and roid x.recyclerview.widget.LinearLayout Manager$ 

import androidx.recyclerview.widget.RecyclerView

class DepsActivity2 : AppCompatActivity() {
 override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.activity\_deps2)

val facName = intent.getStringExtra("FACULTY\_NAME").toString()
val facLink = intent.getStringExtra("FACULTY\_LINK").toString()
val depList = intent.getStringArrayListExtra("DEPARTMENT\_NAMES")
val linkList = intent.getStringArrayListExtra("DEPARTMENT\_LINKS")

setTitle(facName)

```
val recyclerView: RecyclerView = findViewById(R.id.recyclerDepView2)
recyclerView.layoutManager = LinearLayoutManager(this)
recyclerView.adapter = DepartmentRecyclerAdapter2(facName, facLink, ArrayList(depList), ArrayList(linkList))
```

```
val aboutButton: TextView = findViewById(R.id.aboutView)
aboutButton.setOnClickListener {
   val intent = Intent(ACTION_VIEW, Uri.parse("https://knute.edu.ua" + facLink))
   startActivity(intent)
}
```

## DepartmentRecyclerAdapter2.kt

package com.example.myknute

}

import android.content.Intent import android.net.Uri import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import android.widget.Button import androidx.core.content.ContextCompat.startActivity import androidx.recyclerview.widget.RecyclerView

class DepartmentRecyclerAdapter2(private val facName: String, private val facLink: String, private val depList: ArrayList<String>, private val linkList: ArrayList<String>) : RecyclerView.Adapter<DepartmentRecyclerAdapter2.MyViewHolder>() {

```
class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView){
```

```
var nameButton: Button? = null
  init {
     nameButton = itemView.findViewById(R.id.button_department)
  }
}
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {
  val itemView =
    LayoutInflater.from(parent.context)
       .inflate(R.layout.department_item, parent, false)
  return DepartmentRecyclerAdapter2.MyViewHolder(itemView)
}
override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
  holder.nameButton?.text = depList[position]
  holder.nameButton?.setOnClickListener {
     val intent = Intent(Intent.ACTION_VIEW, Uri.parse("https://knute.edu.ua" + linkList[position]))
    holder.nameButton?.context?.startActivity(intent)
  }
}
override fun getItemCount(): Int {
  return depList.size
}
```

#### Додаток Ж

#### MapActivity.kt

package com.example.myknute

import android.os.Bundle import android.webkit.WebView import android.widget.TextView import androidx.appcompat.app.AppCompatActivity import kotlinx.coroutines.Dispatchers import kotlinx.coroutines.runBlocking import kotlinx.coroutines.withContext import org.jsoup.Jsoup import org.jsoup.nodes.Document import org.jsoup.select.Elements

class MapActivity : AppCompatActivity() {
 override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_map)

findViewById<WebView>(R.id.mapView).loadUrl("https://knute.edu.ua/image/MzI3MA==/ef74781780f560103c2719 722e2f0c23.png")

 $val \ result: \ Elements = connectJsoup("https://knute.edu.ua/blog/read/?pid=1206&uk").select("\#timeline > div > small > p:nth-child(3) > span")$ 

```
var address: String = "Адреси навчальних корпусів:" + "\n\n"
for (line in result) {
    if (line.ownText().isNotEmpty()) address += (line.ownText() + "\n")
}
findViewById<TextView>(R.id.mapAddress).text = address
```

#### }

private fun connectJsoup (url: String) : Document = runBlocking {
 withContext(Dispatchers.IO) {Jsoup.connect(url).get()}
}

#### ContactsActivity.kt

package com.example.myknute

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import androidx.recyclerview.widget.LinearLayoutManager

import androidx.recyclerview.widget.RecyclerView

import kotlinx.coroutines.Dispatchers

import kotlinx.coroutines.runBlocking

import kotlinx.coroutines.withContext

import org.jsoup.Jsoup

import org.jsoup.nodes.Document

import org.jsoup.select.Elements

import org.jsoup.select.Selector.selectFirst

```
class ContactsActivity : AppCompatActivity() {
```

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_contacts)

```
val \ result: \ Elements = connectJsoup("https://knute.edu.ua/blog/read/?pid=16857\&uk").select("\#timeline > div > small > table > tbody > tr")
```

```
var names = mutableListOf<String>()
    var tels = mutableListOf<String>()
    var emails = mutableListOf<String>()
    for (line in result) {
       names.add(line.select(">td:nth-child(1)").text())
       tels.add(line.select(">td:nth-child(2)").text())
       emails.add(line.select(">td:nth-child(3)").text())
     }
    val recyclerView: RecyclerView = findViewById(R.id.recyclerView)
    recyclerView.layoutManager = LinearLayoutManager(this)
    recyclerView.adapter = ContactsRecyclerAdapter(names, tels, emails)
  }
  private fun connectJsoup (url: String) : Document = runBlocking {
    withContext(Dispatchers.IO) { Jsoup.connect(url).get()}
  }
}
```

## ContactsRecyclerAdapter.kt

package com.example.myknute

import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import android.widget.TextView import androidx.recyclerview.widget.RecyclerView

class ContactsRecyclerAdapter(private val names: List<String>, private val tels: List<String>, private val emails: List<String>) : RecyclerView.Adapter<ContactsRecyclerAdapter.MyViewHolder>() {

class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView){

```
var nameView: TextView? = null
var contactView: TextView? = null
```

```
init {
```

nameView = itemView.findViewById(R.id.nameView)
contactView = itemView.findViewById(R.id.contactView)
}

```
}
```

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {

```
val itemView =
```

```
LayoutInflater.from(parent.context)
.inflate(R.layout.contact_item, parent, false)
return MyViewHolder(itemView)
```

```
}
```

```
override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
    holder.nameView?.text = names[position]
    holder.contactView?.text = tels[position] + "\n" + emails[position]
}
override fun getItemCount(): Int {
```

```
return names.size
```

```
}
```

## TimetableChooseActivity.kt

```
package com.example.myknute
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
class TimetableChooseActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
     super.onCreate(savedInstanceState)
     setContentView(R.layout.activity_timetable_choose)
     val intent = Intent(this, HtmlTextActivity::class.java)
     findViewById<Button>(R.id.button_calls).setOnClickListener {
       intent.putExtra("ACTION_TYPE", "calls")
       startActivity(intent)
     }
     findViewById<Button>(R.id.button_lessons).setOnClickListener {
       startActivity(Intent(this, EduFormChooseActivity::class.java))
     }
    findViewById<Button>(R.id.button_exams).setOnClickListener {
       startActivity(Intent(this, EduFormChooseActivity2::class.java))
     }
  }
}
```

## EduFormChooseActivity.kt

```
package com.example.myknute
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
class EduFormChooseActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
     super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_edu_form_choose)
    val intent = Intent(this, TimetableActivity::class.java)
    findViewById<Button>(R.id.button_daytime).setOnClickListener {
       intent.putExtra("TIMETABLE_TYPE", "daytime_lessons")
       startActivity(intent)
     }
    findViewById<Button>(R.id.button_eveningtime).setOnClickListener {
       intent.putExtra("TIMETABLE_TYPE", "eveningtime_lessons")
       startActivity(intent)
     }
    findViewById<Button>(R.id.button_extramural).setOnClickListener {
       intent.putExtra("TIMETABLE_TYPE", "extramural")
       startActivity(intent)
     }
  }
}
```

## EduFormChooseActivity2.kt

```
package com.example.myknute
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
class EduFormChooseActivity2 : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
     super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_edu_form_choose2)
    val intent = Intent(this, TimetableActivity::class.java)
    findViewById<Button>(R.id.button_daytime2).setOnClickListener {
       intent.putExtra("TIMETABLE_TYPE", "daytime_exams")
       startActivity(intent)
     }
    findViewById<Button>(R.id.button_extramural2).setOnClickListener {
       intent.putExtra("TIMETABLE_TYPE", "extramural")
       startActivity(intent)
     }
  }
```

62

## TimetableActivity.kt

package com.example.myknute

import android.os.Bundle
import com.google.android.material.floatingactionbutton.FloatingActionButton
import com.google.android.material.snackbar.Snackbar
import com.google.android.material.tabs.TabLayout
import androidx.viewpager.widget.ViewPager
import androidx.appcompat.app.AppCompatActivity
import android.view.Menu
import android.view.MenuItem
import com.example.myknute.ui.main.SectionsPagerAdapter
import com.example.myknute.databinding.ActivityTimetableBinding

class TimetableActivity : AppCompatActivity() {

}

private lateinit var binding: ActivityTimetableBinding

override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)

binding = ActivityTimetableBinding.inflate(layoutInflater)
setContentView(binding.root)

val typeAct: String = intent.getStringExtra("TIMETABLE\_TYPE").toString()

val sectionsPagerAdapter = SectionsPagerAdapter(this, supportFragmentManager)
val viewPager: ViewPager = binding.viewPager
viewPager.adapter = sectionsPagerAdapter
val tabs: TabLayout = binding.tabs
tabs.setupWithViewPager(viewPager)

## SectionsPagerAdapter.kt

package com.example.myknute.ui.main

import android.content.Context import androidx.fragment.app.Fragment import androidx.fragment.app.FragmentManager import androidx.fragment.app.FragmentPagerAdapter import com.example.myknute.R

```
private val TAB_TITLES = arrayOf(
```

R.string.tab\_text\_1,

R.string.tab\_text\_2

)

```
/**
```

\* A [FragmentPagerAdapter] that returns a fragment corresponding to

\* one of the sections/tabs/pages.

\*/

class SectionsPagerAdapter(private val context: Context, fm: FragmentManager) :

```
FragmentPagerAdapter(fm) {
```

override fun getItem(position: Int): Fragment {

// getItem is called to instantiate the fragment for the given page.

// Return a PlaceholderFragment (defined as a static inner class below).

//return PlaceholderFragment.newInstance(position + 1)

```
if (position == 1) return MasterFragment()
else return BachelorFragment()
```

```
}
```

```
override fun getPageTitle(position: Int): CharSequence? {
  return context.resources.getString(TAB_TITLES[position])
}
```

```
override fun getCount(): Int {
    // Show 2 total pages.
    return 2
}
```

```
}
```

#### Додаток С

#### BachelorFragment.kt

package com.example.myknute.ui.main

import android.content.Intent.getIntent

import android.content.Intent.getIntentOld

import android.os.Bundle

 $import\ and roidx. fragment. app. Fragment$ 

 $import\ and roid. view. Layout Inflater$ 

import android.view.View

import android.view.ViewGroup

import android.webkit.WebView

import android.widget.Toast

import com.example.myknute.R

import com.example.myknute.TimetableActivity

 $import\ kot linx. coroutines. Dispatchers$ 

import kotlinx.coroutines.runBlocking

import kotlinx.coroutines.withContext

import org.jsoup.Jsoup

import org.jsoup.nodes.Document

import org.jsoup.select.Elements

// TODO: Rename parameter arguments, choose names that match // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER private const val ARG\_PARAM1 = "param1" private const val ARG\_PARAM2 = "param2"

#### /\*\*

\* A simple [Fragment] subclass.

\* Use the [BachelorFragment.newInstance] factory method to

\* create an instance of this fragment.

#### \*/

class BachelorFragment : Fragment() {

// TODO: Rename and change types of parameters

private var param1: String? = null

private var param2: String? = null

private var result: Elements = connectJsoup("https://knute.edu.ua/blog/read/?pid=1038&uk").select("#timeline > div > small")

```
override fun onCreate(savedInstanceState: Bundle?) {
  super.onCreate(savedInstanceState)
  arguments?.let {
     param1 = it.getString(ARG_PARAM1)
     param2 = it.getString(ARG_PARAM2)
  }
  val typeAct = requireActivity().intent.getStringExtra("TIMETABLE_TYPE").toString()
  if (typeAct == "daytime_lessons") {
     result = result.select("table:nth-child(11)")
  }
  if (typeAct == "daytime_exams") {
     result = result.select("table:nth-child(25)")
  }
  if (typeAct == "eveningtime_lessons") {
     result = result.select("table:nth-child(17)")
  }
  if (typeAct == "extramural") {
     result = result.select("table:nth-child(20)")
  }
override fun onCreateView(
  inflater: LayoutInflater, container: ViewGroup?,
  savedInstanceState: Bundle?
): View? {
  // Inflate the layout for this fragment
  var v = inflater.inflate(R.layout.fragment_timetable, container, false)
  result.removeAttr("width")
  result.select("a").forEach { a -> a.attr("href", "https://knute.edu.ua" + a.attr("href")) }
  v.findViewById<WebView>(R.id.TimetableView).loadData(result.outerHtml(), "text/html; charset=UTF-8", null)
  return v
```

```
}
```

}

```
companion object {
```

/\*\*

\* Use this factory method to create a new instance of

```
* this fragment using the provided parameters.
```

\*

```
* @param param1 Parameter 1.
```

```
* @param param2 Parameter 2.
   * @return A new instance of fragment BachelorFragment.
   */
  // TODO: Rename and change types and number of parameters
  @JvmStatic
  fun newInstance(param1: String, param2: String) =
    BachelorFragment().apply {
       arguments = Bundle().apply {
         putString(ARG_PARAM1, param1)
         putString(ARG_PARAM2, param2)
       }
    }
}
fun connectJsoup (url: String) : Document = runBlocking {
  withContext(Dispatchers.IO) { Jsoup.connect(url).get()}
}
```
## MasterFragment.kt

package com.example.myknute.ui.main

import android.os.Bundle import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import android.webkit.WebView import com.example.myknute.R import kotlinx.coroutines.Dispatchers import kotlinx.coroutines.runBlocking import kotlinx.coroutines.withContext import org.jsoup.Jsoup import org.jsoup.nodes.Document import org.jsoup.select.Elements

// TODO: Rename parameter arguments, choose names that match // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER private const val ARG\_PARAM1 = "param1" private const val ARG\_PARAM2 = "param2"

#### /\*\*

\* A simple [Fragment] subclass.

\* Use the [MasterFragment.newInstance] factory method to

\* create an instance of this fragment.

```
*/
```

class MasterFragment : Fragment() {

// TODO: Rename and change types of parameters

private var param1: String? = null

private var param2: String? = null

private var result: Elements = connectJsoup("https://knute.edu.ua/blog/read/?pid=1038&uk").select("#timeline > div > small")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

arguments?.let {

param1 = it.getString(ARG\_PARAM1)

param2 = it.getString(ARG\_PARAM2)

```
}
```

```
val typeAct = requireActivity().intent.getStringExtra("TIMETABLE_TYPE").toString()
  if (typeAct == "daytime_lessons") {
     result = result.select("table:nth-child(13)")
  }
  if (typeAct == "daytime_exams") {
     result = result.select("table:nth-child(27)")
  }
  if (typeAct == "eveningtime_lessons") {
     result.select("*").remove()
  }
  if (typeAct == "extramural") {
     result = result.select("table:nth-child(22)")
  }
override fun onCreateView(
  inflater: LayoutInflater, container: ViewGroup?,
  savedInstanceState: Bundle?
): View? {
  // Inflate the layout for this fragment
  var v = inflater.inflate(R.layout.fragment_timetable, container, false)
  result.select("a").forEach { a -> a.attr("href", "https://knute.edu.ua" + a.attr("href")) }
  v.findViewById<WebView>(R.id.TimetableView).loadData(result.outerHtml(), "text/html; charset=UTF-8", null)
  return v
```

}

}

```
companion object {
```

/\*\*

```
* Use this factory method to create a new instance of
```

```
* this fragment using the provided parameters.
```

```
* @param param1 Parameter 1.
```

```
* @param param2 Parameter 2.
```

```
* @return A new instance of fragment MasterFragment.
```

\*/

// TODO: Rename and change types and number of parameters

### @JvmStatic

```
fun newInstance(param1: String, param2: String) =
```

```
MasterFragment().apply {
    arguments = Bundle().apply {
        putString(ARG_PARAM1, param1)
        putString(ARG_PARAM2, param2)
        }
    }
    fun connectJsoup (url: String) : Document = runBlocking {
        withContext(Dispatchers.IO) { Jsoup.connect(url).get()}
    }
}
```

### 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=".MainActivity">

<ImageView

android:id="@+id/logo\_image"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:contentDescription="@string/university\_full\_name"

app:layout\_constraintBottom\_toTopOf="@id/menu\_button\_info"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHeight\_max="300dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintWidth\_max="300dp"

app:srcCompat="@drawable/\_6d2f5b583864d2ea173e9a528111b1c" />

<Button

android:id="@+id/menu\_button\_info" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginTop="16dp" android:layout\_marginEnd="16dp" android:text="Загальна інформація про КНТЕУ" app:layout\_constraintBottom\_toTopOf="@+id/menu\_button\_deps" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@id/logo\_image" />

<Button

android:id="@+id/menu\_button\_deps" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginEnd="16dp" android:text="Факультети, кафедри" app:layout\_constraintBottom\_toTopOf="@+id/menu\_button\_map" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/menu\_button\_info" />

#### <Button

android:id="@+id/menu\_button\_map" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginEnd="16dp" android:text="Maпa території" app:layout\_constraintBottom\_toTopOf="@+id/menu\_button\_contacts" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/menu\_button\_deps" />

#### <Button

android:id="@+id/menu\_button\_contacts" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginEnd="16dp" android:text="Tелефонний довідник" app:layout\_constraintBottom\_toTopOf="@+id/menu\_button\_timetable" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/menu\_button\_map" />

#### <Button

android:id="@+id/menu\_button\_timetable" android:layout\_width="Odp" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginEnd="16dp" android:text="Розклад занять та екзаменів" app:layout\_constraintBottom\_toTopOf="@+id/menu\_button\_manual" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/menu\_button\_contacts" />

### <Button

android:id="@+id/menu\_button\_manual" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginEnd="16dp" android:layout\_marginBottom="16dp" android:text="Довідник студента" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/menu\_button\_timetable" />

## Activity\_text.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:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".HtmlTextActivity">

<WebView

android:id="@+id/htmlText"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:isScrollContainer="false"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

### activity\_map.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:id="@+id/ConstraintLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_centerHorizontal="true"

tools:context=".MapActivity">

#### <WebView

android:id="@+id/mapView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:isScrollContainer="false" app:layout\_constraintBottom\_toTopOf="@+id/mapAddress" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent">

</WebView>

# <TextView

android:id="@+id/mapAddress" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="24dp" android:layout\_marginEnd="24dp" android:ems="10" android:gravity="start|top" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@id/mapView" /> </androidx.constraintlayout.widget.ConstraintLayout>

## Activity\_contacts.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=".ContactsActivity">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerView"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_marginBottom="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

tools:listitem="@layout/contact\_item"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

### contact\_item.xml

<?rxml version="1.0" encoding="utf-8"?> <LinearLayout 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:id="@+id/linearContactLayout" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_height="sdp">

#### <TextView

android:id="@+id/nameView" android:layout\_width="200dp" android:layout\_height="wrap\_content" android:layout\_marginEnd="5dp" android:layout\_marginBottom="5dp" android:textStyle="bold" />

<TextView

android:id="@+id/contactView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:textIsSelectable="true" />

</LinearLayout>

## activity\_deps.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=".DepsActivity">

<androidx.recyclerview.widget.RecyclerView android:id="@+id/recyclerDepView" android:layout\_width="match\_parent" android:layout\_height="match\_parent" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" tools:listitem="@layout/department\_item" /> </androidx.constraintlayout.widget.ConstraintLayout>

## activity\_deps2.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=".DepsActivity2">

<TextView

android:id="@+id/aboutView" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="8dp" android:layout\_marginTop="8dp" android:layout\_marginEnd="8dp" android:layout\_marginBottom="8dp" android:gravity="right" android:gravity="right" android:text="Дізнатися більше про факультет ->" app:layout\_constraintBottom\_toTopOf="@id/recyclerDepView2" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

<androidx.recyclerview.widget.RecyclerView android:id="@+id/recyclerDepView2" android:layout\_width="match\_parent" android:layout\_height="0dp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@id/aboutView" tools:listitem="@layout/department\_item" />

## department\_item.xml

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<Button

android:id="@+id/button\_department" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_marginStart="16dp" android:layout\_marginEnd="16dp" android:padding="8dp" android:text="Факультет/Кафедра" /> </LinearLayout>

### activity\_timetable\_choose.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=".TimetableChooseActivity">

<Button

android:id="@+id/button\_calls" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginTop="150dp" android:layout\_marginEnd="32dp" android:layout\_marginBottom="50dp" android:text="Po3клад дзвінків" app:layout\_constraintBottom\_toTopOf="@id/button\_lessons" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

#### <Button

android:id="@+id/button\_lessons" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginEnd="32dp" android:text="Po3клад занять" app:layout\_constraintBottom\_toTopOf="@+id/button\_exams" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/button\_calls" />

<Button

android:id="@+id/button\_exams" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginEnd="32dp" android:layout\_marginBottom="150dp" android:text="Розклад екзаменів" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@id/button\_lessons" />

### activity\_edu\_form\_choose.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=".TimetableChooseActivity">

<Button

android:id="@+id/button\_calls" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginTop="150dp" android:layout\_marginEnd="32dp" android:layout\_marginBottom="50dp" android:text="Po3клад дзвінків" app:layout\_constraintBottom\_toTopOf="@id/button\_lessons" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

#### <Button

android:id="@+id/button\_lessons" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginEnd="32dp" android:text="Po3клад занять" app:layout\_constraintBottom\_toTopOf="@+id/button\_exams" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/button\_calls" />

<Button

android:id="@+id/button\_exams" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginEnd="32dp" android:layout\_marginBottom="150dp" android:text="Розклад екзаменів" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@id/button\_lessons" />

## activity\_edu\_form\_choose2.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=".EduFormChooseActivity2">

<Button

android:id="@+id/button\_daytime2" android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="32dp"

android:layout marginTop="200dp"

android:layout\_marginEnd="32dp"

android:text="Денна"

app:layout\_constraintBottom\_toTopOf="@id/button\_extramural2"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

#### <Button

android:id="@+id/button\_extramural2" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginEnd="32dp" android:layout\_marginBottom="200dp" android:text="3aouta" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@id/button\_daytime2" />

## fragment\_timetable.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:id="@+id/constraintLayout" android:layout\_width="match\_parent" android:layout\_height="match\_parent" tools:context=".ui.main.PlaceholderFragment">

<WebView

android:id="@+id/TimetableView" android:layout\_width="match\_parent" android:layout\_height="match\_parent" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" /> </androidx.constraintlayout.widget.ConstraintLayout>

# Перелік посилань на сайт КНТЕУ для наповнення додатку

1. Екран «Загальна інформація про КНТЕУ»:

https://knute.edu.ua/blog/read/?pid=5&uk

2. Екран «Факультети, кафедри»: містить список, вкладений у тег body > div.container > div > div > div.row > div > div > div > div > ul > li.dropdown.open > ul. Назви факультетів відповідають тексту у тегах ul > li:nth-child(\*) > a > span, a назви кафедр – ul > li:nth-child(\*) > a. Посилання на сторінки факультетів/кафедр вказані в атрибуті href відповідних тегів a.

3. Екран «Мапа території КНТЕУ»:

https://knute.edu.ua/image/MzI3MA==/ef74781780f560103c2719722e2f0c23.png

4. Екран «Телефонний довідник КНТЕУ»:

https://knute.edu.ua/blog/read/?pid=16857&uk

- 5. Екран «Розклад занять та екзаменаційної сесії»:
  - 5.1. Розклад дзвінків:

https://knute.edu.ua/file/MjIxNw==/fff867341b9ecbae68301a3b3ea29598.pdf

5.2. Розклад занять:

https://knute.edu.ua/blog/read/?pid=1038&uk

5.3. Розклад екзаменаційної сесії:

https://knute.edu.ua/blog/read/?pid=1038&uk

6. Екран «Довідник студента»:

https://knute.edu.ua/blog/read/?pid=1086&uk