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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка програмного забезпечення «Learn Phyton base» на ОС

Android»

Студента 4 курсу, 7 групи, спеціальності 121 «Інженерія програмного забезпечення»

Суржика Максима Олександровича

підпис студента

Науковий керівник кандидат технічних наук, доцент

Рзаєва Світлана Леонідівна

підпис керівника

Гарант освітньої програми кандидат технічних наук, доцент

Цензура Микола Олександрович

підпис керівника

КИЇВ – 2020

3MICT

ВСТУП
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ ТЕСТОВИХ СИСТЕМ ТА ТЕХНІЧНЕ
ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ5
1.1. Загальні відомості про освітні додатки для мобільних телефонів5
1.2. Огляд існуючих тестових системб
1.3. Технічне завдання
1.3. Висновки до розділу 112
РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ ДЛЯ
МОБІЛЬНОГО ДОДАТКУ13
2.1. Створення бази даних мобільного додатку та її таблиць 13
2.2. Розгляд полів та таблиць, створення запитів17
2.3 Опис коду бази даних мобільного додатку
2.4. Висновки до розділу 2
РОЗДІЛ 3. РОЗРОБКА ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ МОБІЛЬНОГО
ДОДАТКУ
3.1. Розробка інтерфейсу мобільного додатку27
3.2. Розробка навігації мобільного додатку
3.3. Створення налаштувань мобільного додатку
3.4. Створення головного меню курсів та словнику
3.5 Висновки до розділу 3
ВИСНОВКИ ТА ПРОПОЗИЦІЇ46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ
ДОДАТКИ

5.0	TE	NUL	EXA	JIL	КНТЕУ 12	21 07-1	6.БР	JKh	
Зм.	Аркуш	№ докум	Підпис	Дата	ENKRIENKH	EK.	HIL	EK	
Зав.	ив. кафедри	Криворучко О.В.	TE	NU		Стадія	Аркуш	Аркушів	
Кері	вник	Рзаєва С.Л.	ST.E	N. X	Розробка додатку «Easy Phyton» на OC Android	3	2	47	
Гара	нт	Цензура М.О.	1115	1	TEN RITEVKI	TE	1Kr	TE	
Розр	об.	Суржик М.О.	'HI'	1 VC	Quinn a	Факуль	Факультет інформаційних		
2	EY	TEC	1 1.1	EJ	Smich	технологи, 4 курс, 7 група			

ВСТУП

З кожним роком все більше і більше людей потребують знань в галузі програмування, а сама галузь з кожним роком росте зі швидкістю геометричної прогресії. В наш час інформація та різні способи її передачі набули величезного значення. Відстань, на яку можна передати інформацію розрослась майже на весь світ. a технології які для цього застосовуються стають більш високотехнологічними. Саме тому спеціалісти в галузі IT працюють над покращенням якості зберігання, обробки і передачі інформації, створенням нових технологій.

Сьогодні вже важко уявити наше життя без комп'ютерів і комп'ютерних програм, які керують комп'ютерами і виконуються на них. Мало кого цікавить, як написані ці програми, яка при цьому використовувалась мова програмування. З цієї причини було обрано розглянути саме мови програмування в сучасному світі.

За популярністю станом на 2019 рік лідерами списку на GitHub стали JavaScript, Phyton, Java. Їх популярність обумовлена тим, що ці мови програмування легко вивчити(особливо Phyton), вони використовуються при веб розробці а також в якості мов програмування для створення мобільних додатків.

Багато поважних діячів від колишнього президента США Білла Клінтона до творця Facebook Mapka Цукерберга і фізика і космолога Стівена Хокінга публічно заявляли про те, що базові навички програмування необхідні всім в сучасному світі.

Виконавчий директор Mozilla Foundation Марк Серман вважає що уміння програмувати стало четвертою складовою грамотності. Він говорить, що кожен повинен знати, як наш цифровий світ працює, не тільки інженери.

5	TE	ANUT A	EXA	JIL	КНТЕУ 12	21 07-16.БР				
Зм.	Аркуш	№ докум	Підпис	Дата	ELKATEKK	EN	HIL	EK		
Зав.	кафедри	Криворучко О.В.	JTE'	NU.		Стадія	Стадія Аркуш Арк В З			
Кері	вник	Рзаєва С.Л.	ST.E	12	ОС Android	В				
Гара	нт	Цензура М.О.	1115	1	TEN RITEVKI	TE	IKI	TE		
Розр	об.	Суржик М.О.	'HI'	- VL	Pomura	Факуль	лет інфори	маційних		
	E.Y	TEE	1.1	EJ	Bernyn	Техноле	технологи, 4 курс, 7 гр			

Мова програмування Python відома тим, що за допомогою її можна зробити Веб-додатки, призначені для користувача інтерфейси, проаналізувати великий обсях даних, зробити детальну статистику. Її особливістю є те що, для будь-якої необхідної задачі в якій потрібно знайти рішення, швидше за все, знайдеться відповідний фреймфорк. Вчені прийшли до висновку, що Python можна використовувати в якості основного інструменту для обробки гігантських обсягів даних практично в будь-якій галузі.

Тому, обрана мною тема «Створення програмного забезпечення «Learn Phyton base» на операційній системі Android» є актуальною.

Метою дослідження випускної кваліфікаційної роботи є дослідження освітніх баз даних та інтеграції їх у мобільні додатки, а також розробка мобільного додатку засобами Android Studio для створення освітнього додатку «Easy Phyton».

Об'єктом дослідження є освітній процес.

Предметом дослідження є освітній додаток «Easy Phyton».

Завдання дослідження:

- створення дизайну додатку «Easy Phyton»;
- розробка технічного завдання;
- дослідження існуючих інструментів та платформ для розробки освітнього додатку;
- проектування логічної та фізичної моделей бази даних освітнього додатку;
- підключення спроектованої бази даних до освітнього додатку;
- розробка алгоритмів налаштувань всередині додатку.

Методи дослідження: метод аналізу, метод систематизаці даних, метод синтезу даних, метод класифікації даних.

E	11	I TEY	1111	0	TEN RITE KITE KM	Аркуш
	14	KH1 K	N. H	14	КНТЕУ 121 07-16.БР	
Зм.	Аркуш	№ докум	Підпис	Дата	MARTER KATER KATER K	0

РОЗДІЛ 1.

ОГЛЯД ІСНУЮЧИХ ТЕСТОВИХ СИСТЕМ ТА ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Загальні відомості про освітні додатки для мобільних телефонів

З кожним днем все більше збільшується використання мобільного інтернету і кількість користувачів мобільних пристроїв. Такі зміни цілком обумовлені, зараз всі прагнуть до зручності і мобільності у всіх сферах життя. З появою комунікаторів і планшетів, вже немає необхідності брати з собою ноутбуки та інші пристрої в дорогу, в відрядження, на зустрічі і для інших цілей. Однак мобільні пристрої не приносили б стільки користі без спеціальних програм – мобільних додатків. Значимість мобільних додатків в освіті зростає, а основна причина цього - можливості, які вони дають: навчання в дорозі, на великих дистанціях,

- можливості, які вони дають: навчання в дорозі, на великих дистанціях, функціонал який на даний час вже не вступає комп'ютерам, колективне навчання.

Не відступаючи від теми освіти, затронемо тему програмування більш детально, а саме закладанню основ програмування з мови Phyton яка є однією з найпопулярніших мов програмуванння на даний час, а також мові програмування Java за допомогою якої буде зроблено мобільний додаток. Android Studio пропонує низку інструментів для побудови мобільного додатку за допомогою мови програмування Java або Kotlin. З допомогою Java на Android Studio ми зможемо легко створювати мобільні додатки, а також інтегрувати SQLite в майбутні проекти.

Мобільні навчальні програми повинні відповідати наступним вимогам, а саме:

N								N
5	TE	NUTE	EXT	JTE	КНТЕУ 12	21 07-1	6.БР	KH
Зм.	Аркуш	№ докум	Підпис	Дата	EK HIEKE HTE	- KR	TE	224
Зав. к	афедри	Криворучко О.В.	L'EN	111		Стадія	Аркуш	Аркушів
Керів	ник	Рзаєва С.Л.	HIL	R	Розробка оодатку «Easy Phyton» на OC Android	P1	5	47
Гара	ım	Цензура М.О.	TE	ZK	TENKITENKI	Факульт	тет обліку	аулиту та
Розра	об.	Суржик М.О.	N. Y	E'N	Огляд існуючих тестових систем та	інформаг	ційних сист	тем, 4 курс,
rospoo.		TEN	IKT	FR	техничне завоання на розрооку програмного забезпечення	7 група		

Компактність - компоненти мобільного навчання повинні бути короткими
 за тривалістю, з огляду на те, що вони доступні в середовищі, в якій вірогідні
 потенційні перерви в зв'язку.

 Високий рівень мікроергономіки - висока якість зображення / звуку при малому розмірі екрану. Малий розмір вихідного файлу (швидкість завантаження).

- Повсюдність і доступність - мобільний освітній додаток можна отримати в будь-якому місці, незалежно від місцезнаходження. Все більший радіус дії провайдерів мобільних мереж і наявність мобільних пристроїв забезпечує повсюдне присутність сервісів мобільного навчання, в будь-який зручний для користувача час.

 Доступ на вимогу - за своєю природою мобільний пристрій забезпечує доступ на вимогу для користувача, максимально використовуючи потенціал доставки цінного змісту в момент необхідності.

1.2. Огляд існуючих тестових систем

Тест – це сукупність завдань, які треба виконати при проходженні тестування. Тести, призначені для самоперевірки, містять незмінний набір завдань, що сформував викладач при створенні 101 підручника. Такі тести мають на меті не стільки встановлення рівня знань студента, скільки допомогу ретельніше розібратися в навчальному матеріалі. Набагато більше значення мають тести для проведення рубіжного контролю успішності навчання. Їх застосовують при встановленні рівня засвоєння матеріалу після закінчення роботи над навчальним модулем, а також на семестрових іспитах і заліках. За результатами таких тестувань студент отримує оцінку в балах, яка впливає на порядок подальшого виконання навчального курсу й рейтинг студента.

Для впровадження комп'ютерної перевірки знань необхідно мати спеціалізоване програмне забезпечення тестування (ПЗТ).

1	K	EKI	N'C	Y	KHI KA HI KA JIE KA TE	Аркуш
E	11-	TEY	11/1	10	КНТЕУ 121 07-16.БР	6
Зм.	Аркуш	№ докум	Підпис	Дата	A TEXPOTE NO TE NO	0

Його можна умовно поділити на дві частини:

першу – застосовують при створенні тестових завдань;

головна функція другої частини – встановлення правильності виконання тестових завдань і на цей підставі – оцінювання рівня знань студента.

Тестові завдання мають обмежений набір форм. Формалізована перевірка виконання завдань відбувається так, що ПЗТ порівнює коди відповіді студента з кодами, що закладені як правильні. ПЗТ виробляє команду на зарахування правильної чи неправильної відповіді залежно від того, було або не було зафіксовано співпадіння кодів.

Створення комп'ютерної програми за таким алгоритмом не є важким завданням для кваліфікованого програміста. Тому багато навчальних закладів створювали власні програми тестування. У той самий час ПЗТ розробляли й спеціалізовані комерційні організації. Такі програми випускали як у вигляді окремого програмного продукту, так і як програмний модуль, що входить до загального програмного забезпечення системи управління навчанням.

На даний час існує не так багато тестових систем для навчання на мобільній платформі.

Самим відомим на дний час є Learn Phyton, що охоплює в собі безліч інших навчальних додатків. В ньому є безліч функцій, в тому числі створення власного аккаунту для отримання досягнень, регулювання свого прогресу а також вмонтована консоль Phyton. Але в ньому э декілька недостатків – це платний контент та відсутнысть словника для термінів. На даний час цей додаток є найбільш завантажуємим додатком з Play Market.

На другому місці є додаток Learn Phyton – Pycoders. Він має обмежену кулькість функціоналу але гарно підготовлений матеріал. В ньому ви можете лише читати матеріал та проходити загальний тест, всі теми відкриті відразу. Серед його мінусів – повна відсутність будь якого незвичного функціоналу. Цей додаток схожий на звичайну книжку з однією лиш відмінністью в загальному тесті.

L'	110	TE?	111	10
Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 07-16.БР

Також варто виділити клон Learn Phyton – Изучите програмированние на Phyton. Його особливістю э облегшений інтерфейс. З мінусів можливо віднести те, що консоль там працює не завжди справно, а також є недоступними деякі важливі біліотеки.

1.3. Технічне завдання

- 1. Загальні відомості
 - 1.1.Найменування додатку «ЕРҮ».
 - 1.1.1. Повне найменування додатку «Easy Phyton»
 - 1.1.2. Скорочене найменування додатку «ЕРҮ»
 - 1.2.Планові терміни початку та закінчення робіт
 - 1.2.1. Початок робіт 01.03.2020
 - 1.2.2. Закінчення робіт 01.05.2020
 - 1.3.Порядок та пред'явлення результатів робіт пред'явити готовий додаток керівнику
 - 1.4.Потенційні користувачі системи люди віком від 10 років, які бажають отримати знання з мови програмування Phyton, а також не мают проблем з зором.
- 2. Мета та призначення створення додатку
 - 2.1. Призначення додатку надання знань з мови програмування Phyton
 - 2.2. Мета створення додатку створити додаток для проглиблення і розширення знань з програмування
- 3. Вимоги до додатку
 - 3.1. Вимоги до додатку в цілому у додатку повинна бути реалізована можливість прохождення різних тем з програмування, прохождення тестів після кожної теми, відкриття нових тем за уже виконані тести, наявність словника термінів та живого пошуку в ньому, зміни нашлаштувань додатку.

	L.	ET.I	1.6	N.		Аркуш
E	11	TEY	111	5	КНТЕУ 121 07-16.БР	0
Зм.	Аркуш	№ докум	Підпис	Дата	AN TEXANTE AN TE AN	0

- 3.1.1. Вимоги до структури та функціонування додатку, перелік підсистем додаток базується на мові програмування Java виконаного в Android Studio з підключенням внутрішніх бібліотек та особливостей Gradle.
 - 3.1.1.1. Вимоги до засобів інформаційного обміну між компонентами додатку – зв'язок між компонентами додатку буде виконувати 2 окремі структури: Navigation та SQLite.
 - 3.1.1.2. Вимоги до режимів функціонування додатку додаток повинен функціонувати у режимі оффлайн.
 - 3.1.1.3. Вимоги до діагностування додатку додаток повинен працювати безперебійно та успішно виконувати усі процеси.
 - 3.1.1.4. Вимоги до режимів управління додатком повинно бути реалізовано режим користувача.
- 3.1.2. Показники призначення
 - 3.1.2.1. Параметри, що характеризують ступінь відповідності системи призначенням безперебійність роботи системи, відсутність помилок.
 - 3.1.2.2. Вимоги до пристосованості додатку до змін в додатку буде реалізовано живий пошук в словнику а також можливість додавати та видаляти слова зі словника. Буде створено меню налаштувань, яке буде змінювати роботу всієї системи.
 - 3.1.2.3. Вимоги до збереження працездатності додатку в різних ймовірних умовах – додаток буде зберігати дані після виходу, всі завершені теми зберіграються, налашування після виходу з додатку зберігаються, при зверненні додатку він стає на «паузу» а не вимикається.
- 3.1.3. Вимоги до надійності

	Kr	EKI	M.C	K	WHILE KE HIE KRUTE KRUTE	Аркуш
E	11	TEY	171		КНТЕУ 121 07-16.БР	5
Зм.	Аркуш	№ докум	Підпис	Дата	AN UTE KNUTE NO TE NO	9

- 3.1.3.1. Вимоги до методів оцінки і контролю показників надійності на різних стадіях створення додатку – на стадії тестування дозволена поява помилок, на стадії здачі додатку не дозволена поява помилок.
- 3.1.4. Вимоги до ергономіки та технічної естетики зі знарядь праці необхідний комп'ютер на робоче місце а також наявність телефону з операційною системою Android не нижче версії 4.0.
- 3.1.5. Вимоги до захисту інформації від несанкціонованого доступу доступ до інформації надається тільки користувачу додатку.
- 3.1.6. Вимоги до захисту від впливу зовнішніх факторів додаток приймає зміни тільки безпосередньо від операційної системи (встановлення системної теми).
- 3.1.7. Вимоги безпеки додаток повинен бути безпечним для його користувачів.
- 3.2. Перелік підсистем додатку SQLite.
- 3.3. Вимоги до видів забезпечення:
 - 3.3.1. Вимоги до інформаційного забезпечення інформаційне забезпечення повинне бути цілісним і захищеним від несанкціонованого доступу.
 - 3.3.1.1. Вимоги до складу, структури і способів організації даних в додатку – усі дані повинні зберігатись з-за допомогою баз даних і управлятись СУБД.
 - 3.3.1.2. Вимоги до інформаційного обміну між компонентами системи передача інформації між компонентами системи має виконуватись стандартними протоколами на рівні програмного забезпечення або на рівні платформи (системи керування БД).
 - 3.3.1.3. Вимоги щодо застосування систем управління базами даних потрібно створити адаптери для кожної таблиці бази даних, з яких будуть вноситись дані.

	K	EKI	5.2	Y	WHILE KE HILE KRUTE KRUTE	Аркуш
E	11-	I FE	ITL	1	КНТЕУ 121 07-16.БР	10
Зм.	Аркуш	№ докум	Підпис	Дата	A STERAD TE AD TE AD	10

- 3.3.1.4. Вимоги до структури процесу збору, обробки, передачі даних в системі представлення даних – процес збору, оброки та передачі даних в системі повинен безпечним та швидким.
- 3.3.1.5. Вимоги до захисту даних від руйнувань при аваріях і збоях в електроживленні системи – аварійний вихід з додатку зі збереженням даних в пам'яті телефону.
- 4. За допомогою додатку користувачі повинні мати змогу обирати необхідну тему, отримувати з неї інформацію, в кінці теми проходити тести, які необхідні для зрізу знань та відкриття нових тем. Також користувачі повинні мати змогу використовувати бокове меню і словник в якому будуть всі необхідні терміни, в стрічці пошуку буде реалізовано живий пошук по словнику, а також додавання та видалення елементів зі словника. Користувач зможе в меню налаштувань налаштовувати систему під себе, змінювати тему а також розмір шрифту у всьому додатку, та скидувати налаштування до початкових. Програмне забезпечення повинно базуватись на використанні:

4.1. Android Studio 3.6.2

- 4.2.Java
- 4.3.SQLite

4.4.Бібліотеки Google Gradle

- Для технічного забезпечення рекомендовано використовувати мобільний пристрій з такими характеристиками:
 - 5.1.Android 4.0 та вище
 - 5.2.Qualcomm Snapdragon S1 та вище
 - 5.3.1 гігабайт оперативної пам'яті
- Вимоги до методичного забезпечення не передбачаються, адже використання системи буде інтелектуально простим та зрозумілим.

1	K	ENI	5	Y	KHI KA HI KA JIE KA JE	Аркуш
E	11	TE	171	.0	КНТЕУ 121 07-16.БР	11
Зм.	Аркуш	№ докум	Підпис	Дата	AP TEXPORTE AD TE AD	

1.3. Висновки до розділу 1

У першому розділі було переглянуто переваги використання мобільних пристроїв для навчання та освіти користувачів. Було розглянуто аналоги створюваного додатку. Було виділено спеціальне програмне забезпечення тестування, та вирішено, що для досягенння поставлених цілей необхідно використовувати обидва типи програмне забезпечення тестування.

Також було створено та узагальнено технічне завдання. Під час створення технічного завдання було прийнято рішення про перегляд деяких першочергових підходів до створення програмного забезпечення. Було перевірено та встановлено мінімальні характеристики обладнання для роботи мобільного додатку, а також було вказано все програмне забезпечення, що використовувалось при створенні мобільного додатку.

1	K	EN	Nr.2	K	WHILE KNUTE KNUTE	Аркуш
E	11	TEY	1111	10	КНТЕУ 121 07-16.БР	12
Зм.	Аркуш	№ докум	Підпис	Дата	AN TEXAD ITE AD TE AD	12

РОЗДІЛ 2.

МОДЕЛЮВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ ДЛЯ МОБІЛЬНОГО ДОДАТКУ

2.1. Створення бази даних мобільного додатку та її таблиць

Призначення такої системи є зрозумілим, це можливість отримати знання з прогамування на мові Phyton, а також проходити тестування своїх знань для кращого засвоєння матеріалу.

Система навчання може бути модифікована і використовуватися в різних галузях, в залежності від наповнення бази даних. Найкращим аналогом буде створення додатку для вивчення англійської мови, адже в майбутньому додатку буде створено словник з живим пошуком та можливістю додавати, вилучати, та редагувати елементи.

В реляційних базах даних і плоских базах даних, таблиця це набір елементів даних (значень), які організовані з використанням моделі вертикальних стовпчиків (з різними іменами) і горизонтальних рядків. Таблиця має визначену кількість стовпчиків, в той час як кількість рядків може різнитися в різні моменти.

СУБД розроблена на SQLite та складається з 7 таблиць:

- 1. Теми main (ID теми, назва теми, лічильник зарахованих тестів, блокатор теми);
- Заняття lessons (ID заняття, назва заняття, ID теми до якої підєднано заняття, блокатор заняття);
- 3. Текст для занять text (ID тексту, текст заняття, ID заняття);
- 4. Зображення для занять picture (ID зображення, посилання на зображення, ID заняття);

5	TE	NUTE	EXT	JE	КНТЕУ 12	1 07-1	6.БР	KHI	
Зм.	Аркуш	№ докум	Підпис	Дата	EKRYHTE KRYTE				
Зав. к	афедри	Криворучко О.В.	1 SEY	111		Стадія	Аркуш	Аркушів	
Керів	ник	Рзаєва С.Л.	HIL	2	Розроока оооатку «Easy Phyton» на ОС Android	P2	13	47	
Гаран	ım	Цензура М.О.	TE	X	TENKATENKY	Факультет обліку, аулиту та			
Розро	б.	Суржик М.О.	7.0	FN	Моделювання та створення бази даних	інформац	ійних сист	ем, 4 курс,	
V		NER	Kh	L K	для мобільного додатку	7 група		TE	

- 5. Запитання question (ID запитання, ID заняття, запитання);
- 6. Відповіді answers (ID відповіді, ID запитання, перевірка на правильність, перевірка на активність кнопки, відповідь);
- 7. Словник dict (ID елементу словника, елемент словнику, опис елементу словнику);

Концептуальна(рис. 2.1), логічна(рис. 2.2) та фізична(рис. 2.3) моделі бази даних мобільного додатку:



Концептуальна модель включає в себе сім таблиць які показують основні принципи роботи даної бази даних. Дані для тем підгружаються з таблиці занять. Кожне заняття отримує свою інформацію з двух таблиць зображення для занять та текст для занять. Для відкриття нової теми необхідно пройти тест і тому таблиця заняття перенаправляє користувача на тест до якого підгружаються таблиця запитань та таблиця відповідей. Дана база даних має 2 віртуальні таблиці, які створюються лише на момент відкриття необхідного вікна в мобільному додатку.



Рис. 2.2. Логічна модель бази даних мобільного додатку

Логічна модель бази даних має більш розширену структуру, тут можливо побачити які рядки є присутніми в кожній таблиці, а також їх Primary та Foregin keys. В базі даних присутні звязки один до багатьох та один до одного. Це все

1	Kr	EK.	KH:	Y	WHILE KAUTE KAUTE KAUTE	Аркуі
E	11	1 FEY	ITT	10	КНТЕУ 121 07-16.БР	15
Зм.	Аркуш	№ докум	Підпис	Дата		15

обумовлено тим, що під час створення програми вдалося спростити базу даних, а саме вхідні дані з таблиці «text». Додаток отримує відразу весь урок без зображень, потім ділить текст на частини по спеціальному знаку, та створює в памяті свій власний віртуальний список, де заміть спеціального знаку він вставляє зображення отримане з бази даних. Не беручи до уваги зв'язок між таблицями «Заняття» та «Текст» в решті таблиць використовується стандартне зєднання один до багатьох. Таблиця «Словник» є відокремленою від інших таблиць і ніяк з ними не звязана.



Рис. 2.3. Фізична модель бази даних мобільного додатку

Фізична модель бази даних вказує на тип даних що находяться в таблиці. Перевірку на проходження тесту чи перетворення активної кнопки в неактивну було вирішено замінити перевіркою на значення «1» та «0» в базі даних. При розблокуванні елементу дані в базі даних змінюються та записуються.

2	Kr	EK	H	Y	WHILE KRUTE KRUTE	Аркуш
E	11	15EY	ITT	10	КНТЕУ 121 07-16.БР	16
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE AV TE IN	10

2.2. Розгляд полів та таблиць, створення запитів

Поле — це мінімальна (але найважливіша) порція інформації в записі, над якою визначені операції введення, виведення, перетворення тощо. Воно має ім'я, значення, характеризується типом і низкою додаткових властивостей.

Назви полям дає користувач, назви типів є стандартні, а значення полів випливають зі змісту конкретної задачі.

Очевидною унікальною властивістю кожного поля є його Ім'я. Крім імені в полі є ще властивість Підпис. Підпис – це та інформація, що відображається в заголовку стовпця. Її не треба плутати з іменем поля, хоча якщо підпис не заданий, то в заголовку відображається ім'я поля. Різним полям, наприклад, можна задати однакові підписи. Це не перешкодить роботі комп'ютера, оскільки поля при цьому як і раніше зберігають різні імена.

Перед створенням таблиці треба визначитись для чого призначаються різні типи полів. Основна властивість текстового поля – розмір.

Числове поле служить для введення числових даних. Воно теж має розмір, але числові поля бувають різними, наприклад для введення цілих чисел і для введення дійсних чисел. В останньому випадку крім розміру поля задається також розмір десяткової частини числа.

Для введення логічних даних, що мають тільки два значення (Так чи Ні; 0 чи 1; Істина чи Неправда і т.п.), служить спеціальний тип – Логічне поле. Неважко догадатися, що довжина такого поля завжди дорівнює 1 байту, оскільки цього більш ніж досить, щоб виразити логічне значення.

Дуже цікаве поле Лічильник. На перший погляд це звичайне числове поле, але воно має властивість автоматичного нарощування. Якщо в базі є таке поле, то при введенні нового запису в нього автоматично вводиться число, на одиницю

	Kr	EKI	Y	Y	HILL KRUTE KRUTE KRUTE	Аркуі
E	11	ILEY	ITE	6	КНТЕУ 121 07-16.БР	17
Зм.	Аркуш	№ докум	Підпис	Дата	APUTE AVITE AVIE AV	17

більше, ніж значення того ж поля в попередньому записі. Це поле зручне для нумерації записів.

Проаналізувавши поставлену задачу, були створені наступні таблиці:

«main» – це таблиця, в якій міститься інформація про всі основні теми курсу програмування, інформація про кількість підтем в темі, кількість пройдених підтем в темі а також вказано заблоковано чи розблоковано наступну тему. Ця таблиця пов'язана з таблицею «lessons».

«lessons» – це таблиця, в якій міститься інформація про всі підтеми, що наявні в программі, також тут указано ID_main для зв'язування підтем з необхідними темами, а також вказано заблоковано чи розблоковано підтему. Ця таблиця пов'язана з таблицями «text», «picture», «question».

«text» – це таблиця в якій указано всю текстову інформацію необхідну для всіх підтем, а також ID необхідної підтеми. Ця таблиця пов'язана з таблицею «lessons».

«picture» – це таблиця зі списком зображень. В таблиці вказано посилання на необхідні зображення, їх порядковий номер та ID необхідної підтеми. Ця таблиця пов'язана з таблицею «lessons».

«question» – це таблиця зі списком запитань для тестів. В ній наявна інформація про ID необхідної підтеми, запитання в текстовому вигляді та порядковий номер кожного запитання. Ця таблиця пов'язана з таблицею «lessons».

«answers» – це таблиця зі списком всіх відповідей для тестів, що є в программі. В ній наявна інформація про ID необхідного запитання, перевірка на вірність чи невірність відповіді, а також поле для зміни правильної відповіді на неактивну кнопку. Ця таблиця пов'язана з таблицею «question».

«dict» – це незажежна від інших таблиця із списком термінів, в яку користувач може вносити свої зміни. В ній наявна інформація про назву терміну та його описання. Ця таблиця не повязана з іншими таблицями.

1	Kr	2K	Y.	Y	WHILK WITH KAUTE KAUTE	Аркуи
E	11	IL EY	ITT	10	КНТЕУ 121 07-16.БР	10
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE AV TE AV	10

Запити дають широкі можливості для вибору, сортування і обчислення з використанням даних однієї таблиці. Можна налаштувати конкретний пошук будь чого в нашій базі даних, наприклад, теми та звязані з ними підтеми.

Запит на вибірку можна використовувати не тільки для відбору даних, але і для їх поновлення. Запит на вибірку має ряд властивостей, які можна використовувати для зміни роботи запиту.

В режимі таблиці доступні самі різні операції з даними - огляд, сортування, фільтрація, поновлення і друк. Але достатньо часто приходиться проводити обчислення і огляд даних з декількох таблиць. Відобразити потрібні дані можна за допомогою запитів.

Після виконання запита на вибірку (який відбирає інформацію з таблиць і інших запитів бази даних, в той час як при виконанні запиту на зміну дані вставляються, поновлюються або видаляються) SQLite створює набір записів, які містять відібрані дані. В більшості випадків з набором записів можна працювати так само, як з таблицею: можна проглянути і відібрати інформацію, роздрукувати і поновити дані. Але на відміну від реальної таблиці, цей набір записів фізично не існує в базі даних. Ассеss створює набір записів з даних таблиць тільки під час виконання запиту. Якщо змінити дані в наборі записів, SQLite внесе відповідні зміни в таблицю, на базі яких побудований запит.

Приклад написання коду запитів для незалежної таблиці «dict» можливо переглянути на рисунку 2.4

1		1 V AU		1		1 1
1	Kr	EKI	Y	Y	WHILE KAUTE KAUTE KAUTE	Аркуш
E	11	L'EN	ITT	6	КНТЕУ 121 07-16.БР	10
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE NUTE IN	19

perioge commentarity omen independed
<pre>import androidx.lifecycle.LiveData;</pre>
<pre>import androidx.room.Dao;</pre>
<pre>import androidx.room.Delete;</pre>
<pre>import androidx.room.Insert;</pre>
<pre>import androidx.room.Query;</pre>
<pre>import androidx.room.Update;</pre>
<pre>import com.example.diplomepy.databse.entities.DictModel;</pre>
import java.util.List;
@Dao
<pre>public interface DictDao {</pre>
@Update
<pre>void update(DictModel note);</pre>
@Insert
<pre>void insert(DictModel note);</pre>
@Delete
<pre>void delete(DictModel note);</pre>
@Query("SELECT * FROM dict ORDER BY id dict ")
LiveData <list<dictmodel>> getAllDict();</list<dictmodel>
END AND AN AR AR KEL

Рис. 2.4. Створення інтерфейсу запитів для таблиці «dict»

2.3 Опис коду бази даних мобільного додатку

Щоб керувати та синхронно користуватися програмою і базою даних SQLite ми повинні підключити цю базу даних до нашого мобільного додатку. Для цього необхідно створити окремий клас, який ми назвемо MainDatabase(рис. 2.6).

Перед тим як створити базу даних потрібно закинути вже створену базу даних в ассети додатку(рис. 2.5).



Рис. 2.5 Перенесення створеної бази даних до додатку

1	Kr	EKI	Y	Y	THIS KALLS KAUTE KAUTE	Аркуш
E	11	1 FEY	ITE	10	КНТЕУ 121 07-16.БР	20
Зм.	Аркуш	№ докум	Підпис	Дата	LANJIE AN TEN AN TEN AN	20



Рис. 2.6. Створення класу MainDatabase

Для безперебійного використання бази даних потрібно створити «ентічки» кожної таблиці(рис. 2.7).

Приклад опису одного з классів бази даних (передача полів таблиці «main») вказано на рис. 2.8.

	KL	EKI	N'Z	K		Аркуш
E	11	1-E	ITE	0	КНТЕУ 121 07-16.БР	21
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE AN TE IN	21

```
package com.example.diplomepy.databse.entities;
import androidx.room.Entity;
import androidx.room.PrimaryKey;
@Entity(tableName = "main")
public class MainTest {
    @PrimaryKey(autoGenerate = true)
    private int id_main;
    private int id_main;
    private String name;
    private int counter;
    private int blocked;
    public MainTest(String name, int counter, int blocked, int id_main) {
        this.name = name;
        this.counter = counter;
        this.blocked = blocked;
        this.blocked = blocked;
        this.blocked = blocked;
        this.id_main = id_main;
    }
    public int getId_main() { return id_main; }
    public String getName() { return name; }
    public int getBlocked(){return blocked;}
    public int getBlocked(){return blocked;}
    public void setCounter(int counter) { this.counter = counter; }
    }
```

Рис. 2.8. Приклад передачі таблиці «main»

Наступним кроком э передача інтерфейсів(запитів) для кожної з цих таблиць. Було створено 7 таблиць та 9 інтерфейсів. Це все обумовлено тим, що інтерфейси відповідають за надання необхідних даних користувачу. Саме тому було створено 2 додаткових інтерфейси які обєднують декілька таблиць в одну і виводять лише необхідні дані. Створені інтерфейси необхідно помістити в окремий раскаде та назвати dao(рис. 2.9).

1	Kr	KKI	Y	Y	HILK HILK KNUTE KNUTE	Аркуш
E	11	LE	ITE	5	КНТЕУ 121 07-16.БР	22
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE NO TE NO	22

Рис. 2.9. Створені інтерфейси

Приклад створення одного з інтерфейсів вказано на рис. 2.4.

Наступним і останнім кроком для повноцінного функціонування бази даних є створення Адаптеру для кожної таблиці. Адаптер приймає значення з таблиці, в ньому вказується що необхідно робити з цими даними і які дані необхідно взяти з таблиці. Також він заносить дані до таблиці. Приклад створення такого адаптеру для таблиці «question» можливо переглянути на рис. 2.10. Це найменший з адаптерів, адже він тільки приймає та передає дані. Решта адаптерів є складними і включають в себе від 100 стрічок коду.

Також через необхідність отримувати та виводити на один фрагмент дані з різних таблиць необхідно було створити особливий класс. В розробці мобільних додатків він має назву «сінглтон».(рис. 2.11). При створенні фрагменту, що виводить інформацію на екран неможливо в одному класі фрагменту виводити декілька списків таблиць. Саме тому створюється класс «сінглтон» який бере необхідні дані з таблиць і передає в різні фрагменти.

package com.example.diplomepy.ui	.question;
import	
and so we are	
public class Singleton {	
private static volatile Sing	leton instance;
private int answeredCounter	
<pre>private int currentTestId = (</pre>	0;
private List <questiontestmod< td=""><td>el> lists;</td></questiontestmod<>	el> lists;
private List <maintestpart2></maintestpart2>	testList;
private List <mainlest> mainle</mainlest>	estList;
private int testMainid;	
public static Singleton getI	nstance() {
Singleton <u>localInstance</u>	
<pre>if (localInstance == nul</pre>	1) {
synchronized (Single	ton.class) {
<u>localInstance</u> =	
if (<u>localInstanc</u>	<u>e</u> == null) {
$instance = \underline{l}$	<pre>ocalInstance = new Singleton();</pre>
}	
return <u>localinstance;</u>	
<pre>void setListOuestions(List<0</pre>	uestionTestModel> guestions) { lists = guestions; }
<pre>public List<questiontestmode< pre=""></questiontestmode<></pre>	<pre>l> getListQuestion() { return lists; }</pre>
<pre>public void setListPart2(Lis</pre>	t <maintestpart2> mainTestPart2s) { testList = mainTestPart2s; }</maintestpart2>
public int getSizeListPart2()) { return testlist size(): }
papire ine Berstreiter al ri(
public MainTestPart2 getTest	<pre>ByIndex(int index) { return testList.get(index); }</pre>
<pre>public void setCurrentTestId</pre>	<pre>(int id){</pre>
<pre>if(id + 1 < testList.siz</pre>	e()) {
currentTestId = id +	
	A PARTIE RANGER
Drtak	2 11 Harring Kugay (Singlaton)
РИС.	2.11. частина класу «Singleton»
ALAN	
	KUSKINSKINSK
TEY WITHEN	КНТЕУ 121 07-16.БР
No докум Підпис Пата	

Зм.

4ркуш 24 Під час роботи з базою даних було створено 7 класів таблиць, 9 інтерфейсів для кожної з цих таблиць, та 2 віртуальних, життєвий цикл яких триває тільки під час запуску окремих функцій, 7 адаптерів для кожної з таблиць, особливий клас, що приймає та перенаправляє дані з таблиць, головний клас бази даних. Були створені запити для кожної з таблиць. У всіх таблицях окрім «dict» використовуються лише 2 методи SELECT та UPDATE. В таблиці «dict» користувач може вносити свої зміни саме тому там використовуються 4 методи: SELECT, INSERT, UPDATE та DELETE.

	1XT	KAN	HT	XY.	ATE AN ATE AN ATE AN	Аркуш
É		IE-EY	TT		КНТЕУ 121 07-16.БР	25
Зм.	Аркуш	№ докум	Підпис	Дата	NUTE NO TE NUTE IN	25
1			1 1.	121	KI CINT IN ZINALIN	

2.4. Висновки до розділу 2

У цьому розділі було детально описано структуру бази даних. Було описано кожну таблицю та призначення кожного стовпчику і взаємодію між ними. Розглянуто взаємодію SQLite та Android Studio на прикладі створення власного мобільного додатку. Було досліджено всі особливості створення бази даних для мобільного додатку, а також створення окремих класів. Для безперебійного функціонування бази даних на мобільних пристроях необхідно створити *Adanmep*, а для використання даних з різних таблиць в одному фрагменті клас «Singletone».

Було наведено приклади створення звязку кожної таблиці та класу, що відповідає цій таблиці. Наведено приклад створення інтерфейсу (запитів до бази даних за допомогою Android Studio).

Окремо було показано створення класу бази даних для підключення існуючої бази даних в мобільний додаток.

KIE	K' K'	· KI st		Аркуш
ENINE	EXIT		КНТЕУ 121 07-16.БР	26
Зм. Аркуш №	докум Підпис	Дата		20

РОЗДІЛ З.

РОЗРОБКА ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ МОБІЛЬНОГО ДОДАТКУ

3.1. Розробка інтерфейсу мобільного додатку

Для створення інтерфейсу программи вирішено було використовувати xml фрагменти в середовищі розробки Android Studio. Найкращим рішенням поставлених задач є перезаписання деяких фрагментів для повторного використання. Дані, що приходитимуть з бази даних будуть відправлятися на один і тей же фрагмент де вони будуть перезаписуватися відповідно до обраного індексу.

Перед початком роботи потрібно визначитися з поставленими задачами. Необхідно створити бокове меню, а так же необхідні вікна додатку (головна, курс, словник, налаштування, сайти та про нас). Для початку було створено бокове меню засобами Android studio(рис. 3.1).

Рис. 3.1. Створення проекту за допомогою шаблону Navigation drawer activity

5	TE	NUTE	E A	JTE	КНТЕУ 12	21 07-1	6.БР	KHI	
Зм.	Аркуш	№ докум	Підпис	Дата	KATT KRUTT				
Зав. к	афедри	Криворучко О.В.	1 SEY	T_{II}		Стадія	Аркуш	Аркушів	
Керів	ник	Рзаєва С.Л.		2	Розроока обоатку «Easy Phyton» на ОС Android	P3	27	47	
Гаран	ım	Цензура М.О.	TE	Y Z	TE KATENK	Факультет обліку, аулиту т		аулиту та	
Розроб.		Суржик М.О.		E N	Розробка інтерфейсу та функціоналу	інформаційних систем, 4 курс,			
E	51	NEN	1Kh	EK.	мобільного додатку	7 група			

Наступним кроком створення проекту є робота з окремими вікнами (далі – фрагменти). Для початку потрібно створити прості вікна: Головна, сайти, про нас. Для створення кожного з цих фрагментів потрібно створити окремий xml-файл – розмітку нашого вікна(саме ця складова відповідає за зовнішню складову нашого фрагменту). Приклад створених xml-файлів показано на рис. 3.2.

Рис. 3.2. Приклад створення xml-файлу для вікна корисної інформації.

Наступним кроком в створенні фрагменту є його програмний опис. Для цього потрібно створити окремий клас, який буде відповідати за цей фрагмент, а також за всі кнопки та інші складові з якими можлива взаємодія. Для створення класу потрібно перейти в директорію в якій зберігаються всі класи проекту. Нижче буде приведено приклад опису класу «Головна» для виведення інформації користувачу при переході на обраний фрагмент:

1	Kr	EK.	Y	Y	HI KANTE KANTE KANTE	Аркуш
E	11	LEY	ITE	1	КНТЕУ 121 07-16.БР	20
Зм.	Аркуш	№ докум	Підпис	Дата	TE AD TE DO TE AD	20

потрібно імпортувати бібліотеку UML та парсингу для посилань. Для доступу до створених кнопок необхыдно використовувати Bundle.

🗯 Eile Edit View Navigate Code Analyze Refact								a x
📲 DiplomEPY 🔤 app 🔚 src 🗎 main 🕽 📷 java 🖓 🖬	com) 🖬 ecample 🖬 diplomepy 🖿 ui)	🖬 sites) 😌 SitesFragment		app • D Pixel 2 API 24			0 5 🖬 🖬 🤻	Q 9 Q]
🛫 🕍 Android 💌	😳 😤 🌣 — 🕒 Settings java 🖉 🛔	🛓 fragment_sites.xml 👋 💿 AboutFragment.java 🖄 🧕	SitesFragment.java 🔧 🛔 mobile_navigation.xml 🗵	dement_dict.xml ×	fragment_about.xml 🛞	fragment_glossary.xml ×	fragment_home.xml ×	sett 🗬
E Ins app								Grad
manifests								
java								
E Com								
e example	public clas	ss SitesFragment extends Fragment (
2 Till diplomepy								
e bildetabse		View onCreateView(@NonNull LayoutInflater int						
2 P D helper		ViewGroup container, Bundle						
A Models		<pre>ew root = inflater.inflate(R.layout.frugment_s</pre>	ites, container, attachToRoot false);					
repositories		ot.findviewById(R.id.fromel).setOnClickListene						
		Intent 1 = new Intent(Intent.Ac/ION_VIEW, Or	1.porse(https://ru.wikipedia.org/wiki/pyc					
La about								
G AboutFragment		the drive Butd(B) id Second) contractions in						
		Totast T - new Totast/Totast //Totast	Construction of the second se Second second seco					
Sectors Sectors		chart is now intent (intent.ac/inv_view, ur	1.parse(<u>https://bython-scripts.com/</u>));					
A La nome								
homer ragment		<pre>st find/imsbyId(R id formal) satUn(lick)istane</pre>	relation as I					
R autorian		Intent I = new Intent/Intent & TTON VIEW	A parte ("https://www.pethon.org/doc/")					
T Pa uter		startArtivity(T):						
Sites Franment								
E fate		ot findViewBvTd(R id Framed) setOnClickListene						
G Singlel ineData		Intent I = new Intent(Intent ACTION VIEW, Un	i.parse("https://withonworld.ru/osnovy/dak					
California California		startActivity(I):						
G StartAno								
Dir com (antimid fert)	100							
 Di com (tenti) 	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	ot.findViewById(R.id.frame5).setOnClickListene	r(view -> {					
Image in the intervented intervented in the intervented in the intervented interven	38							
E Massets		startActivity(I):						
- Dil drawabie								
V Dillayout								
activity_add_dict.xml		startActivity(I);						
activity main.xml								
answer_button_true.xml								
answer_button.xml								
app_bar_main.xml								
- button_test.aml								
button_test_no_active.xml								
👌 🕌 button_test_part2.xml								
button_test_part2_no_active.xml								
content_main.xml								
derent_dict.xml								
g fragment_about.xml								
🚪 👬 fragment_glossary.kml								
fragment_home.aml								불
★ fragment_lesson.xml	SiteiFranmen							
E 1000 Lt R Version Control E 6 Jacob 6	Ind Different							() Event Lon
Gradie over Enicked in 51 c 416 mc (vertred v 22/20)	termer and recomment							t and the state of
(yesterday 22:26)						30193	CALL DIFG Aspaces G	in mayter 🖬 👳

Рис. 3.3. Створення класу «FragmentSites» та парсинг всідних посилань

Основні моменти зі створенням фрагменту закінчено. Для більш детального та функціонального створення фрагменту потрібно використовувати різні класи та стилі (як приклад створення меню налаштувань та словнику, курсів).

~	Kr	2K	H	V	HILKA THE KANTE KANTE	Аркуш
E	11	TEES	ITT	10	КНТЕУ 121 07-16.БР	20
Зм.	Аркуш	№ докум	Підпис	Дата	LAN TE AN TE NO TE NU	29

3.2. Розробка навігації мобільного додатку

Під час створення додатку з шаблону було створено стандартне меню з навігації. Воно потребує модифікацій та змін. Для початку було створено всі головні фрагменти мобільного додатку. Для того, щоб їх звязати необхідно в клас «MainActivity» внести зміни (рис. 3.4).

Рис. 3.4. Класс «MainActivity» який э головним координуючим класом в мобільному додатку

В цьому класі необхідно імпортувати налаштування навігатору, а також вказати всі створені раніше фрагменти по їх іd. Після цього потрібно оголости сам навігатор та створити(або перейти) до створеного меню навігації (рис 3.5).

1	Kr	K	H	Y	WHILE KRUTE KRUTE KRUTE	Аркуш
E	11	L'E	ITT	1	КНТЕУ 121 07-16.БР	20
Зм.	Аркуш	№ докум	Підпис	Дата		30

Рис. 3.5. Меню навігації

В меню навігації необхідно оголосити всі фрагменти, а також дати їм назву, яка буде відображатися вверху вікна. Також потрібно прописати всі звязки з додатковими вікнами, які були створені. Це необхідно зробити для того, щоб користувач мав змогу повертатися на попередні фрагменти.

Для безперебійного використання бокового меню потрібно створити окремий віртуальний інтерфейс, який буде виконуватися лише при натисканні на необхідну кнопку в створеному «тулбарі». Також цей інтерфейс потрібно звязати з головною сторінкою, для того, що би після запуску додатку користувач відразу міг використовувати бокове меню.

Також до навігації відноситься вікно запуску додатку, яке зявляється перед користувачем від час першого запуску додатку. Воно має назву – стартовий фрагмент. Для його створення потрібно створити окремий xml-файл та класс, що наслідується від материнського класу Activity. Приклад створення класу та xml-файлу показано на рис. 3.6. та рис 3.7.

	KI	EN.		1 M		Аркуш
E	11	1 LEY	ITT	10	КНТЕУ 121 07-16.БР	21
Зм.	Аркуш	№ докум	Підпис	Дата	APUTE AUTE AUTE AU	31

Також потрібно налаштувати сам запуск додатку, та порядок виконання Activity при його запуску. Стандартні налаштування спочатку запускають головну Activity => MainActivity. Для того, щоб це виправити потрібно внести зміни до маніфесту створюваного додатку, та вказати, що першим повинен запускатися стартовий фрагмент(рис. 3.8).

xml version="1.0" encoding="utf-8"?
<pre><manifest <="" pre="" xmlns:android="http://schemas.android.com/apk/res/android"></manifest></pre>
<pre>package="com.example.diplomepy"></pre>
TOW SET WAR BE UN STAN STAN
<pre><application< pre=""></application<></pre>
android:name=".StartApp"
android:allowBackup="true"
android:icon="@drawable/logo2"
android:label="EPY"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".ui.activities.LogoActivity"></activity>
<intent-filter></intent-filter>
<action android:name="android.intent.action.MAIN"></action>
<category android:name="android.intent.category.LAUNCHER"></category>
<activity< td=""></activity<>
android:name=".ui.activities.MainActivity"
android:label="EPY"
<pre>android:theme="@style/AppTheme.NoActionBar" /></pre>
<pre> </pre>
EN WILL NOW AND AN AND AN ARE AND A

Рис. 3.8. Маніфест створюваного додатку

3.3. Створення налаштувань мобільного додатку

Створення налаштувань один з найбільш складних процесів. Адже після закриття додатку чи переходу з одного фрагменту до іншого всі налаштування

1	Kr	EKI	H	Y	WHILE KAUTE KAUTE KAUTE	Аркуш
E	11	THEY	ITT	10	КНТЕУ 121 07-16.БР	22
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE AN TE AN	- 33

повинні зберігатись до наступного входу, та використовуватись додатком при його активації. Для цього в Android Studio було використано спеціальні налаштування, які мають назву SharedPreferenses. Це спеціальна бібліотека, що записує зміни додатку окремим файлом, та зчитує його при кожному завантаженні додатку.

Для початку потрібно в папці res створити окрему директорію, яка буде наслідувати SharedPreferenses. Потім створюється xml-файл в якому вказується інтерфейс майбутніх налаштувань. Різниця звичайного файлу і файлу налаштувань полягає в внутрішньому конструкторі який надається Android Studio. Приклад створеного файлу налаштувань вказано на рис. 3.9.

Рис. 3.9. xml-файл налаштувань мобільного додатку

Наступним кроком є створення окремого класу, який буде наслідувати SharedPreferenses а також буде материнським для двух інших класів(SharedPreferences.OnSharedPreferenceChangeListener, TextHelper.RecreateApp). В цьому класі буде описано основні моменти

	Kr	EK	Kriz	K		Аркуш
E	. 11	ILEY	ITT	10	КНТЕУ 121 07-16.БР	21
Зм.	Аркуш	№ докум	Підпис	Дата	NUTE NO TE NO TE NO	34

налаштувань в додатку. Так наприклад головний клас налаштувань отримує методи з двух дочірніх класів для вибору необхідної теми та для необхідного розміру тексту, який вказує користувач при користуванні додатком. Приклад головного класу налашувань вказано на рис 3.10.

🗯 Eile Edit View Navigate Code Analyze Befa	tor Build Rym Tools: VCS Yandow: Help DiplomEPYTC/Uter/Uter/Uter/Uter/Uter/Uter/Uter/Uter
🗮 DiplomEPY 🛛 🕅 app 🖉 🖿 src 🖉 🖬 main 🕽 🔤 java 🕬	🕱 com) Dif example Dif deplomenty 🕲 definings - 🕹 👘 🗇 🖧 📰 🗇 🏘 🛄 🖓 👰 🛄 🖓 🤱 🖓 🦉 🖉
🛫 🚜 Android 👻	
Com	14 public class Settings extends PreferenceGnampeListener, TextSelper.RecreateApp { 15 SharedPreferences.OtSharedPreferenceGnampeListener, TextSelper.RecreateApp { 16 17 17 17 17 17 17 17 17 17 17 17 17 17
k 🖿 🖿 helper	
Image:	19 0f public void onCreatePreferences(Bundle savedInstanceState, String rootKey) (
► Da repositories ▼ Da ui	
 Em about AboutFragment 	
 Eactivities LogoActivity MainActivity 	Defense button = findPreference("Compta porpsc"); 10 button_setDoPreference(lickListemer(preference -> (26 26
► 🖿 glossary ▼ 🖿 home	27 alextDialogs.thew(getChild*cageentMenger(), (sp "som_text");
question	
▼ 🖾 sites	
SitesFragment	
► D∎ tests	
G SingleLiveData	35 Ø 🚇 op public void onSharedPreferenceChanged(SharedPreferences, sharedPreferences, String key) {
e Settings	
StartApp	<pre>37 final Context context = getContext();</pre>
5 E com (andreadTest)	
com (list)	39 B if (context i= null) {
 Di java (generated) 	40 final String theme = sharedPreferences.getString(key, "Bstring/theme_light");
g b assets	ThemeHelper.setTheme(theme, getContext());
d drawable	
a layour	
Eq. all scholage and	final contract contact - cationtact()
activity_main.xnu	
E answer botton and	15 (constant to wall 38 antichthicker) to wall /
ann har main yml	40 final String text = sharedPreferences.setString(key, "Mitring/text modium"):
A button test ami	TextHolor, setText(text, setContext(), recention:http://
button test no active ami	
button test part2xml	
2 Sutton test part2 no active amil	
🗧 🛁 content_main.xml	
😤 😽 element_dict.xml	
* # fragment_aboutami	Settings on Cristofederatical profesence
IE TODO 년 옷 Version Control 또 술 Logcat	
IDE and Plugin Updates: Android Studio is ready to up	281 CRUF UTF-8 Aspaces Git mader in

Рис. 3.10. Створення основного класу налаштувань для мобільного додатку

Для того, щоб у додатку було декілька тем потрібно в ресурсах створити нові файли з кольорами. На рис 3.11 – приклад створення такого файлу для темної теми.

	xml v<br <resour <cc <cc <th>version="1.0" rces> plor name="co plor name="co plor name="co</th><th><pre>encoding="utf-8"?> lorPrimary">#AA1919 lorPrimaryDark">#000000 lorAccent">#AA1919</pre></th><th>LE LIL</th></cc </cc </resour 	version="1.0" rces> plor name="co plor name="co plor name="co	<pre>encoding="utf-8"?> lorPrimary">#AA1919 lorPrimaryDark">#000000 lorAccent">#AA1919</pre>	LE LIL
Рис. 3	.11. Створ	ення новс	ого ресурсу для нічного варіанту теми	EKA
	JUTE	KI KY	КНТЕУ 121 07-16.БР	Аркуш 35

Для зміни тексту потрібно створити новий файл стилів, в якому необхідно зазначити необхідні назви розміру шрифту а також їх локальний розмір. На рис 3.12 – приклад створеного файлу для зберігання розмірів та позначень шрифту.

Рис. 3.12. Створення нових стилів для регулювання розміру шрифту за допомогою налаштувань

Також для того, щоб програма розуміла звідки потрібно брати ці ресурси, необхідно створити два допоміжні класи – для тем та для тексту. Назви ThemeHelper та TextHelper відповідно. В них потрібно сказати які ресурси беруться, та куди передаються. Останньою функцією налаштувань є функція скидування налаштувань до початкових. Для коректної роботи та уникнення випадків випадкового натискання цієї кнопки було розроблено ще один клас, що відповідає за виклик AlertDialog – діалогових вікон з підтвердженням операції. Приклад створення цього класу показано на рис. 3.13.

1	Kr	EK.	Y	Y	HILK HILK KNUTE KNUTE	Аркуш
E	11	LEY	ITE	10	КНТЕУ 121 07-16.БР	26
Зм.	Аркуш	№ докум	Підпис	Дата	TE AD TE AD TE AD	30

Рис. 3.13. Створення класу AlertDialog

3.4. Створення головного меню курсів та словнику

Набільш трудоємким є створення створення двох фрагментів, які є повязаними з базою даних. Раніше уже біло створено фрамент словнику, а отже потрібно додати туди живий пошук, а також декулька функцій для управління все підключеною базою даних. Для створення живого пошуку потрібно використати внутрішню складову «пошук» при створенні xml-файлу фрагменту, а також додати туди список з елементів, що є в базі даних. Також було створено кпоку додавання нових елементів. Потрібно було зробити так, щоб пошук був живим, а значить обновлювався в реальному часі та записував зміни до бази даних. При цьому потрібно було виводити новий відфільтрований список при будь яких змінах в рядку пошуку. Приклад створення фрагменту словнику показано на рис. 3.14.

1	Kr	EK.	Y	Y	WHILE KALLE KALLE KALLE	Аркуш
E	11	1 FEY	ITE	5	КНТЕУ 121 07-16.БР	27
Зм.	Аркуш	№ докум	Підпис	Дата	LANJE AN TE NUTE IN	37

Рис. 3.14. Створення xml-файлу словнику мобільного додатку

Аналогічно, як і раніше потрібно створити класс, що буде відповідати за виведення даних з бази даних в фрагмент а сам фрагмент користувачу. Приклад створення класу словнику вказано в рис. 3.15.

E DisponeEFY Ist app Mit soc Mit main Mi jour, Dit son Di g → Antoid → ① ☆ Z → Mitsep Z → Mitsep Antoid → ② ☆ W Hitsep Antoid → ○ ○	lazarapia Ed algebrange Ed al Balgebrange Ed al Da glastary & Disconforgerant All algebrange Ed algebrange Ed algebrange All a	
Vier on the service to use of the servi	<pre>Prove the second s</pre>	
NITEYNU	Рис. 3.15. Створення класу для словнику	CITE!
KIEKI	TER WHIEK HIEK WITEK	A
Аркуш № докум	КНТЕУ 121 07-16.БР Підпис Дата	KIUT

Пілся створення словнику потрібно створити ще 3 фрагменти, інформація в яких бузе змінюватися відповідно до обраного значення в головному фрагменті курсів. Вони будуть перезаписуватися, тому створити їх потрібно пустими, та звязати з базою даних. Розпочати потрібно в порядку спадання. Саме тому першим було створено xml-файл головного фрагменту курсів(рис. 3.16).

👗 Eile Edit View Navigate Code Analyze Refactor Bu	ld Ryn Took: VC\$ Window Help DiplomEPY[C\User\User\AndroidStudioProjecti\DiplomEPY]\app\src\main\res	Jayout Bragment, test.xml [app] - Android Studio —	σ×
🗮 DiplomEPY 🕅 app 🖿 src 🖿 main 🕅 res 🕅 layout		🔨 👗 app 👻 📴 Pixel 2 API 24 🛩 🕨 C1 📰 🏛 🖏 🕼 👘 🕼 🖌 🗸 💿 😏 📭 📼 🝂	1 🦦 Q 🗖
🛫 🛎 Android 👻 🚱		nent_test.xml 📧 🏭 fragment_glossary.xml 👘 🥥 ThemeHelper.java 👘 🏭 values\colors.xml 👘 🏭 night\colors.xml 👘 🏭 styles.xm	× .
Auron A	1 (?wal version="1.0" encoding="utf-5"?>		≡ ⊡ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
Cli home Di home Di lisson Di question Di stes Di stes Di stes OligiciveData OligiciveData	Construction to constructor appoint, sugget a constructor appoint and instantical "http://tcieness.android.com/public androidslappoint_abstructure.android.com/tools" androidslappoint_abstructure.android.com/tools" androidslappoint_abstructure.android.androids tools:construct: "ail Artist: Anstruct.android.porent"		
G Settings StartApp	candroids.recyclerview.widget.RecyclerView extended.tds=04.id/sel07est*	Something Some	
Cal Com (and and and and and and and and and and	11 android:layout_width="antri, parent" 12 android:layout_height="antri, parent" 13 tool:Silitite*"aity.purt/buttor_test">>>	Something Some	
♥ Big res ▶ Dil drawable ▼ Dil layout # articite add diet und	14 Crandroids.coordinatorlayout.vidget.Coordinatorlayout>	Some	
🚆 octivity_main.xml 🍓 answer_button_true.xml 🍓 answer_button.xml		Something Some	
े app_bar_main.xml		Some Some Some	
button_test_part2_no_active.xml def content_main.xml def clement_distanti def fragment_about xml		Some Some	
Image: Constraint of the second sec		Something Some	
Bragment, question.xml Bragment, interarml Bragment, interarml Bragment, itest, part2.xml Bragment, itest, part2.xml		Some Some	
de fragmet stest.xml de image.xml de image.xml de image.xml		something Some	
Carney Desperiment Xnu		Some Some Some	
E havigstion ► Di values r → ► Di values r → ► Di values ► Di values			
IIE TODO IP ⊉ Version Control III & Lopcat ≺ Build	98 Terminal prday 22:27)	14 CRLF UTF-8 Aspeces Git	🕕 Event Log : master 🔏 🚍

Рис. 3.16. Створення xml-файлу головного фрагменту курсів

Після створення xml-файлу головного фрагменту курсів його потрібно оголостити за допомогою классу TestFragment(рис. 3.17). В ньому вказано основні звязки фрагменту з базою даних, а також показано перехід на наступний фрагмент TestFragment2 – в якому знаходяться підтеми відповідних тем. Коли користувач обирає необхідну тему в головному фрагменті, то в дочірній фрагмент вносяться відповідні дані з бахи даних, коли користувач виходить з цього фрагменту дані зницають і фрагмент знову стає пустотілим. Так же само працюють і наступні фрагменти створені після фрагменту підтем.

Зм. Аркуш № докум Підпис Дата		КНТЕУ 121 07-16.БР		TE	(EEY)		E
	Y in	AN TE AN TE AN TE	Дата	Підпис	№ докум	Аркуш	Зм.

4ркуи **39**

Рис. 3.17. Створення класу TestFragment

Клас та xml-файл для дочірнього фрагменту підтем є схожими тому розглядати його створення не потрібно. Фрагмент уроків приховує в собі деякі особливості. Тут також потрібно створити свій xml-файл(рис. 3.18).

- the question			= 👪 styles.com) - 🖌
► Dil sites ▼ Dil tests	1 k?wel version="1.0" encoding="utf-#"?}		
Call tests Call content Themes	2 @ CLinearLayout xmlns:android="http://schemai.android.com/apk/res/ar	noid" 🖉 🗞 U Pixel 2 × 🛲 21 × @ AppTheme × 🕲 Default (en-us) ×	U H
			Tribe
	android:layout_width="match_parent"		
Spotal kaData	android:orientation="vertical"	Item 0	
© Settings		Item 1	
		tien 2	A DECEMBER OF
		item 4	
 Diff cont ((e)) 	android:1d="y+id/list_lesson"	item 5	
Reg (ava ()(interation))	android:layout_width= match_parent	Item 6	
T late	14. android:layout_weight="1">	Item 7	
		Item 9	
activity_main.aml			
answer_button_treasmi	kandroidx.cardview.widget.Cardview		
App bar main sml	21 android:layout width="match parent"		
abutton_test.ont	22 android:layout_height="45dp"		
autom_test_no_active.xml			
			O THE REAL
element_dict.cml			A MARKEN
fragment_about.ami	andraid lawaut widths "natch narrest"		
Tagment justicary and	android:layout height="natch parent"		
fragment_lesses.aml	31 android:background="@drawable/button2 back"		A IN THE
fragment question.cml	android:gravity="center_vertical right"		
			ACCESSION IN
fragment_test_part2.xml			
fragmet_tedtami	sudepid-lawat widths"bis"		
	android:layout_width="match_ma		
aner, brader, main.cml	35 android:layout_marginEnd="50do"		
			+ -
Di menu			11 2
> Di ravigation	android:text-llepeirs go recty		10 8
No. 1			
Bergeneterer			i i
III TODO US 9 Version Cantrol III & Loans & Build	H Terminal		C Durat Lon
IDE and Plugin Updates: Android Studio is ready to undate livesterd	y 22:27)		ces Git: master 🎓 👼
	Due 210 Concession	a uml thating unarin	
	гис. э. 16. Створенн	я хип-фаилу уроків	
TENL		N'IV. N'KE'N	5
AT KAL			
KIEKI			
KILEKI	NAEL KATEN		114
KT E KIL	TE KITE	КНТЕУ 121 07-16.БР	HI
KT EKN	JTE KITE	КНТЕУ 121 07-16.БР	HIL

Після створення xml-файлу, потрібно створити клас, що буде передавати необхідні значення в створений xml-файл, та показувати його користувачу. Його особливістю є те що він приймає з баз даних 1 значення з текстом та безліч картинок. Для того, щоб картинки біли поставлені на свої місця в тексті необхідно ставити спеціальний знак. Під час приймання тексту в класі через цикл текст перевіряється, і там де знаходиться спеціальний знак ділиться на 2 частини, а замість другого спеіального знаку на те місце ставиться каритнка. Таким чином виходить новий список, який пізніше завантажується в RecyclerView. Зображення зявляються за допомогою посилання яке знаходиться в базі даних, а самі зображення знаходяться в директорії assets, там же де й база даних. Приклад створення класу для уроків зображено на рис. 3.19.

👗 Eile Edit View Navigate Code Analyze Re	actor Build Ran Josh VCS Window Help DiplomEPY [CUlture:UbsrlAndhold:ZudhaPhayects)DiplomEPY [Speptic:Chaningara,Com/neurophildiplomepy/Lultureson/LeuronFragment Java Fappi - Android Studio — 🗗 🗙
🗮 DiplomEPY 🕅 app 🔤 src 🖬 main 📄 java	ති (com) Dit example) Dit diplomepy) Dit wil Dit lesson 🖉 Lesson/Regment 🔨 👗 sp 💌 🕞 Find 2 249,24 🔍 💌 🗇 🕼 🚛 🖉 🗸 🔍 🔾 🤹
🙂 🗯 Android 👻	😌 ≑ 💠 — 💿 TeatHelperjava 🔍 AlexDialogijava 🕐 Glossan-Fragment,java 🍰 fragment, text.ml 💿 TestsFragment, jeva 🚔 fragment, jeva 👘 fragment, jeva v
er v in app E v in app T v in manifests	1 package com.example.diplomepy.ui.lesson;
AndroidManifest.xml	
	33 B autility of the Landau External Autority Engeneration
S S D diplomany	
i bi databse	23 private RecyclerView respectively:
* T in helper	23 private LessonRecyclerViewidapter lessonRecyclerViewidapter;
AlertDialogs	
G TextHelper	
O ThemeHelper	
Image:	
Image: Provide the image of	
	30 View root = inflater.inflate(R.layout.frogment_Lesson; container, intachToRoot:false);
Di about	
activities	<pre>int id = getArguments().getInt(kng: "id", defaultValue @);</pre>
P glossary	
P La home	14 Lessonsviewmodel = new viewmodelProvider(owner: this, new LessonsviewmodelFactory(getActivity(), getActivity(), getActivity(), 10)).get(Lessonsviewmodel.class);
En resson	2 Jacoberurlertindiater = nev lasenderurlertieudeter()
Q LessonsViewModel	33 recvclerView = root.findViewBvtd/R.id./ist (esson):
G LessonsViewModeFac	pry 39 recyclerview.sethasFixedSize(true);
Computer question	40 recyclerView.setLayoutHanager(new LinearLayoutHanager(getContext()));
> Dill sites	
e Di tests	
🗧 🛛 🕝 SingleLiveData	
Settings	
G StartApp	
com (androidTest)	Bundle bundle new Bundle();
g Com Grit)	
ava (generated)	<pre>bundle.putInt('10', <u>10</u>);</pre>
3 assets	10 HundrackenThen and action Teddite Station control (and activity) 1. Id any here for more the
	and and a second vice is a marganized to a provide and the first and a second vice is a second vice of the second vice is a second vice of the sec
activity add dictami	
E activity_main.xml	
answer_button_true.cml	
answer_button.cml	
🚆 🛛 🏭 app_bar_main.xml	
hutton_test.aml	
2 button_test_no_active.xml	
button_test_part2.xml	
button_test_part2_no_active.xml	si lessonsviewFodel.getLessoniist().observe(getViewLifecycleOwner().baseNodels -> (
* Gontent_main.xml	LessontFragment
III TODO 1 Strision Control I & Logcat	A full 28 Terminal
DE and Plugin Updates: Android Studio is ready to up	21:14 CRLF UTF-3 Aspaces Git: master 😘 🙍

Рис. 3.19. Створення класу LessonsFragment

Наступним, самим складним та необхіднім є створення фрагменту з прохожденням тесту. Саме цей фрагмент є відповідальним за успішне або

	K	KK	N'r	K		Аркуш
E	. 11	1-EY	ITT	5	КНТЕУ 121 07-16.БР	11
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE AN TE AN	41
X.	~			11		

неуспішне прохождення тесту, а також за розблокування нових тем та підтем у двух попередніх фрагментах. Він є відповідальним за зміну лічильника в головному класі курсу. Для його створення необхідно використовувати безліч підкласів, а сам адаптер та фрагмент є найбільш нагруженими серед всіх адаптерів та класів в даному додатку. Для початку необхідно створити xml-файл тестів(рис. 3.20).

Рис. 3.20. Створення xml-файлу тестування

Створений клас для даного xml-файлу не є перезавантаженим адже всі операції проходять в адаптерах, яких було створено 2 для рівномірного розподілення навантаження на класи, а також для асинхронного виконання операцій. Користувач повинен мати змогу переглядати уже пройдені тести. Саме тому під час прохождення тесту всі правильні відповіді зберігаються і користувач може переглянути їх в будь який момент, навіть після виходу з додатку. Приклад створення класу для тесту наведено на рис. 3.21.

	Kr	ZKI	H	Y	WHILE KAUTE KAUTE	Арку
E	11	ILEY	ITT	10	КНТЕУ 121 07-16.БР	12
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE NUTE NU	42

Рис. 3.21. Ствоерння QuestionFragment

Оскільки роботу з базами даних було розглянуто в розділі 2, то тут варто додати, що блокатор даного додатку працює виключно від даних з бази даних. Саме тому нові уровки розблокуються лише тіольки тоді коли кількість правильних відповідей на тести буде дорівнювати кількості запитань на тесті. Також для розблокування нових тем необхідно, щоб кількість пройдених тестів дорівнювала кількості підтем в попередній темі. Всі ці аспекти були виконані в адаптері відповідей. Всі функції та методи виконуються виключно дивлячись на необхідній індекс, що знаходиться в базі даних. Для того, щоб в одному адаптері мати змогу використовувати дані з різних таблиць потрібно створити сінглтон. Приклад створення сінглтону приведено на рис 3.22.

1	Kr	KK	Y	Y	THE KRITE KRITE KRITE	Аркуш
E	11	LE	ITE	10	КНТЕУ 121 07-16.БР	12
Зм.	Аркуш	№ докум	Підпис	Дата	APUTE APUTE APTEND	43

🗯 Eile Edit View Navigate Code Analyze Befacts						a ×
Na DiplomEPY In app 🖿 src 👘 main) 🖿 java) Da	l com) 🖬 ecample) 🖬 diplo	mepy) 🕅 ui) 🖬 question) 🕝 Singleton				a 🖬
별 🛎 Android 💌		🛔 fragment_test.xml 👋 🔞 TestsFragment.java 👋 🧔 Sir				javi -=: #
🐔 🔻 🖿 java						Sing
T Cam						a.
T 🖿 example						
👻 🔽 diplomepy						
T Di helper						
G AlertDialogs						
j G TextHelper						
G ThemeHelper						
Image:						
repositories						
🔻 🖿 ui						
Babout						
Dil activities						
Biglossary						
bit home						
Di lesson						
V Du question						
🔻 🛅 adapter						12
G AnswerAdapter						
🕝 QuestionAdapter		if (localInstance == null) {				
▼ 🖾 adapterViewFragment						
G ArrayListFragment						
G ArrayListViewModel						
G ArrayListViewModelFar						
V 🖬 dialogs						
G AlertDialogFalse						
AlertDialogTrue						
G QuestionFragment		vold setListQuestions(List <questiontestmodel)< td=""><td></td><td></td><td></td><td></td></questiontestmodel)<>				
C QuestionViewModel						
Singleton		public List <questiontestmodel> getListQuestion</questiontestmodel>				
► Di sites	A STATE OF A					
E tests						
5 G SingleLiveData						
E G Settings		public int getSizeListPart2() (return test)				
G StartApp						
📭 🕨 🖿 com (Indiaid Test)		public MainTestPart2 getTestByIndex(int inde:				
E com (test)						
É ► Ex iava (generated)						
E massets						
V Intes		currentTestId = id + 1;				
drawable						
T Di lavout						
* Activity add dictared						
activity main.xml		mublic int setCurrentTestId() (return current				
A answer, button true yeal		partice the Becca terrestration and				
III TODO 🔰 👷 Version Control 📰 🚊 Logcat 🗳						
IDE and Plugin Updates: Android Studio is ready to updat						

Рис. 3.22. Створення класу Singleton

Всі необхідні зображення для даного проекту необхідно завантажити в директорію drawable(рис. 3.23).

Рис. 3.23. Директорія drawable

N		1 V AU		1		. 1 8
1	Kr	EKI	YE	Y	HI KALTEKA JEKA	Аркуи
E	11	L'EN	ITE	5	КНТЕУ 121 07-16.БР	11
Зм.	Аркуш	№ докум	Підпис	Дата	ANUTE AN TE AUTE AU	44

3.5 Висновки до розділу 3

В цьому розділі детально описано створення мобільного додатку, а також всіх його складових.

Інтерфейс не є складним, і користувач легко сможе орієнтуватися в ньому. Кольори біли подібрані не «токсичні», саме тому подходять для всіх категорій людей.

Навігація була розроблена згідно офіційних вимог, працює безперебійно та не має прихованих багів. Користувач може вільно пересуватися всередині додатку.

Налаштування були мінімізовані, що дозволило вивести користувачу загальні налаштування додатку, для спрощення його використання. Зміни тем та шрифту функціонують добре. (Перевірено моделями телефонів Samsung S9, Meizu M3 note, Xiaomi Redmi 4X). Переглянути результат налаштувань зміни теми можливо в Додатку А, зміни розміру тексту в Додатку Б.

Всі фрагменти перевірено на безперебійну роботу, також перевірено корректну роботу головного фрагменту курсів. Виправлено всі баги, які було знайдено під час перевірки.

	Kr	2KT	H	1V	HIE KN JIE KN TE MY	Аркуш
E	11	THEY	TT	1	КНТЕУ 121 07-16.БР	AF
Зм.	Аркуш	№ докум	Підпис	Дата	AN TE AN TE AV TE AV	45

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У результаті дослідження було розглянуто переваги використання мобільних пристроїв для освіти та навчання користувачів. Було розглянуто існуючі на ринку освітніх послуг аналоги створюваного програмного додатку «Easy Phyton». Для створення мобільного додатку вирішено було використовувати Android Studio на мові програмування Java та SQLite для створення внутрішньої бази даних. Автором створено технічне завдання.

Для повноцінного функціонування програмного додатку розроблено та детально описано структуру бази даних, а саме: описано кожну таблицю та призначення кожного стовпчика і зв'язки між ключовими полями. Розглянуто взаємодію SQLite та Android Studio на прикладі створення власного мобільного додатку «Easy Phyton». Було досліджено всі особливості створення баз даних для мобільних додатків, а також створення окремих класів. Окремо було відображено створення класу бази даних для підключення існуючої бази даних в мобільний додаток.

Розроблено прорамну реалізацію мобільного додатку «Easy Phyton», призначеного для всіх категорій населення, метою якої є освітній процес користувачів. Всі фрагменти перевірено на безперебійну роботу, також перевірено коректну роботу головного фрагменту курсів. Виправлено всі баги, які було знайдено під час перевірки.

На основі виконаних завдань було створено власну програму «ЕРҮ», а також проведено дослідження актуальності створеної програми на сьогодення. З цього можливо зробити висновки, що всі поставлені завдання виконано і дослідження проведено.

5	TE	NUTE	EX	JTE	КНТЕУ 121 07-16.БР				
Зм.	Аркуш	№ докум	Підпис	Дата	KANTEKAUT				
Зав. кафедри		Криворучко О.В.	15EY	n_{i}		Стадія	Аркуш	Аркушів	
Керівник		Рзаєва С.Л.	HIL	CH V	Розробка додатку «Easy Phyton» на OC Android	ВП	46	47	
Гарант		Цензура М.О.	TE	K	TEL KATELK	Факультет обліку, аулиту та			
Розроб.		Суржик М.О.	10.1		Висновки та пропозиції	інформаційних систем, 4 курс, 7 група			
		NEN	, Kh.						

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems. 8-е изд. / М.: Вильямс, 2005. 843 с.
- Кренке Д. Теория и практика построения баз даннях. / Д. Кренке Спб.: Питер, 2005. – 844 с.
- Герберт Шилдт. «Java 8. Руководство для начинающих». / Шилтд Г., 2014 139 с
- 4. Брюс Еккель. «Философия Java». / Еккель Б., 2016 40 с

Інтернет ресурси:

- 1. GitHub [Електронний ресурс] Режим доступу до ресурсу: https://github.com/Maks4u
- AndroidDocumentation [Електронний ресурс] Режим доступу до ресурсу: <u>https://developer.android.com/reference/android/content/res/Configuration.html#f</u> <u>ontScale</u>
- 3. CodingInFlow [Електронний ресурс] Режим доступу до ресурсу: <u>https://codinginflow.com/</u>
- Освой Android играючи [Електронний ресурс] Режим доступу до ресурсу: http://developer.alexanderklimov.ru/android/
- 5. Stackoverflow [Електронний ресурс] Режим доступу до ресурсу: https://stackoverflow.com/

5	TE	NUTE	EXK	JTE	КНТЕУ 121 07-16.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	EKRUTEKRUT	- KR	TE	3 ch
Зав. кафедри		Криворучко О.В.	1 LEY	111	Розробка додатку «Easy Phyton» на ОС Android	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.	HIL			ВД	47	47
Гарант		Цензура М.О.	TE	XK	Факультет обліку, аули			аулиту та
Розроб.		Суржик М.О.	JU.T	L'L'	Список використаних джерел	інформаційних систем, 4 курс, 7 група		

додатки

Додаток А

Візуальне зображення екрану завантаження додатку при зміні теми

Візуальне зображення зміни головного меню курсів та бокового меню при зміні

теми

Додаток Б

Візуальне зображення зміни розміру тексту в залежності від обраних налаштувань

користувачем