

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ЗОРИЧ СЕРГІЙ ДМИТРОВИЧ

Допускається до захисту:

завідувач кафедри

інформаційних технологій,

канд. техн. наук, доцент

_____ Т. В. Нескородева

« _____ » _____ 2021р.

РОЗРОБКА КЛІЄНТСЬКОЇ ЧАСТИНИ ДЛЯ НАВЧАЛЬНОЇ СИСТЕМИ

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Римар П. В., старший викладач

кафедри інформаційних технологій

Оцінка: _____ / _____ / _____

(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2021

АНОТАЦІЯ

Зорич С.Д. «Розробка клієнтської частини для навчальної системи»
 Спеціальність 122 «Комп'ютерні науки» Донецький національний університет
 імені Василя Стуса, Вінниця, 2021.

У кваліфікаційній (бакалаврській) роботі досліджено технологію розробки мобільних додатків, а саме клієнтської частини. За допомогою мови програмування Dart та фреймворка Flutter було розроблено мобільний додаток «Get Simple», який дозволяє користувачеві проходити онлайн-навчання за різноманітними курсами. Додаток зроблено як для викладача, так і для студента. Відповідно, можна реєструвати нові курси або бачити перелік курсів, на які вже записано зі статистикою проходження.

Ключові слова: мобільний додаток, Figma, прототип, дизайн, компоненти, Flutter, GetX, , Firebase, Stacked Themes, віджети.

47с., 31 рис., 20 джерел.

ANNOTATION

Zorych Sergii «Development of the client part for the educational system»
 Specialty 122 «Computer Science» Vasyl Stus Donetsk National University,
 Vinnytsia, 2021.

In the qualification (bachelor's) work the technology of development of mobile applications, namely the client part is investigated. Using the Dart programming language and the Flutter framework, the Get Simple mobile application has been developed, which allows the user to take online training in a variety of courses. The application is made for both teacher and student. Accordingly, you can register new courses or see a list of courses for which you have already registered with the statistics of passing. Keywords: mobile application, Figma, prototype, design, components, Flutter, GetX,, Firebase, Stacked Themes, widgets.

47p., 31 pic., 20 sources.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ПОНЯТТЯ МОБІЛЬНОГО ДОДАТКУ.....	8
1.1 Поняття мобільного додатку.....	8
1.2 Сучасні мобільні додатки.....	8
Висновок до розділу 1	12
РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ ТА ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ	13
2.1 Постановка задачі	13
2.2 Огляд існуючих аналогів.....	14
2.3 Огляд дизайнерських інструментів.....	15
2.4 Мова програмування Dart	16
2.5 Інструменти для розробки.....	17
Висновок до розділу 2	22
РОЗДІЛ 3. РОЗРОБКА КЛІЄНТСЬКОЇ ЧАСТИНИ МОБІЛЬНОГО ДОДАТКУ .	23
3.1 Створення прототипу.....	23
3.2 Створення дизайну проекту.....	25
3.3 Створення архітектури проекту	31
3.4 Створення UI мобільного додатку	32
Висновок до розділу 3	42
ВИСНОВКИ.....	43
СПИСОК ЛІТЕРАТУРИ.....	45

ВСТУП

Смартфони зробили життя людини набагато простіше і швидше. У міру розвитку технологій, що лежать в основі смартфонів, індустрія мобільних додатків також постійно розвивається. З більш ніж 2,96 млн додатків (і їх кількість зростає) в магазині Google Play, Android з моменту своєї появи набрав обертів і випередив конкурентів з часткою ринку більше 85%. З кожним роком Android продовжує пропонувати нові ідеї, інновації, методи і інструменти в розробці мобільних додатків. Майбутнє розробки додатків для Android дуже багатообіцяючі, що допоможе розробникам випередити своїх конкурентів.

Впровадження Android зростало семимильними кроками, в основному завдяки покращеному інтерфейсу. Щоб залишатися попереду в постійно мінливій індустрії розробки мобільних додатків, вкрай важливо втілювати в собі тенденції завтрашнього дня. Перш ніж вирішите найняти професійного розробника мобільних додатків для Android, необхідно краще зрозуміти останні тенденції розробки додатків для Android.

Навіщо бізнесу мобільні додатки:

- Організація продажів і стимулювання повторних покупок.
- Підвищення лояльності покупців і клієнтів.
- Автоматизація бізнес-процесів.
- Аналіз аудиторії на основі різних даних.
- Прийом платежів.
- Надання сервісу нового рівня і т.д.

Через карантин та пандемію COVID-19 зросла кількість користувачів мобільних додатків. Тижні та місяці в ізоляції в карантині змусили бізнес та навчання перейти в онлайн, що сильно вплинули на звички та спосіб життя людей. Люди частіше купують в інтернеті, замовляють їжу онлайн, більше проводять часу у смартфонах й витрачають набагато більше часу в іграх. Саме це призвело до зростання ринку мобільних додатків. Порівнюючи з 2019 роком, у 2020 році на 40% зріс час, який люди витрачають на мобільні додатки

щомісяця. Ці тенденція зберігається й навіть зростає у 2021 році. Також зросла кількість завантажень мобільних додатків, а саме: мобільні ігри (43%), додатки для бізнесу (105%) та доставка їжі (73%). Також у 2020-й зросла кількість відеоконференцій. Завантаження додатку Zoom для відеоконференцій зросла на 2000% порівнюючи з 2019 роком. Мобільні додатки перетворилися в найбільшу споживчу систему що існує на планеті. За прогнозами, у 2021 році світова виручка від мобільних додатків має досягнути \$6,3 трлн.[1]

Створення платформи для навчання в умовах COVID-19 є дуже актуальною темою. За допомогою цієї платформи люди можуть не виходячи з дому проходити курси для саморозвитку, поліпшенню професійних навиків та навпаки, люди – що можуть надати корисну інформацію шляхом публікування своїх курсів на платформі.

Переваги онлайн освіти:

- Доступ до програм найкращих викладачів світу: онлайн-освіта дає можливість навчатися скрізь, де є інтернет.
- Найновіша інформація, технології, теорії: матеріал онлайн-курсів оновлюється динамічніше, ніж офлайн-програми.
- Безкоштовне чи доступніше за ціною навчання, ніж звичайне денне навчання в університеті.
- Можливість навчатись у будь-якому місці та у бідь-який час.
- Можливість поєднувати різні формати та підходи до навчання.

Недоліки онлайн освіти:

- Відсутність живого контакту з викладачем та групою.
- Самомотивація та самоорганізації є досить складною. Саме це є наслідком низького відсотку завершення курсів (у світі 7%, в Україні – 15%).
- Недостатня кількість спеціалізованих матеріалів вищого рівня складності: більшість матеріалів створені для вступного рівня – щоб охопити якомога більшу аудиторію.
- Закінчення курсів не гарантує наявності потрібних знань і навичок.

Актуальність теми дослідження. Сьогодні ринок мобільних додатків є найпопулярнішою тенденцією IT-ринку. Сучасні мобільні додатки мають кілька вирішальних переваг: це швидкість та зручність під час користування. Кількість технологій та варіацій реалізації постійно зростає і саме це робить непростим створення мобільного додатку.

Мета дослідження. Розробка мобільного додатку-платформи для навчання «Get Simple» з можливістю створення курсів, набором групи, отриманням сповіщень, переглядом всіх курсів, відправкою запита на проходження курсу (все залежить від обраного функціоналу при реєстрації). Додаток має бути візуально приємним, інтуїтивно зрозумілим та багато функціональним.

Завдання дослідження:

- Вивчити та з'ясувати, як працюють Figma, Dart, Flutter SDK, Android Studio, Android Emulator, Firebase, Firebase Authentication, Firebase Firestore, Firebase Storage.
- Визначити як розробляти мобільні додатки для Android та iOS, використовуючи Flutter SDK та мову програмування Dart.
- Створити мобільний додаток «Get Simple».

Об'єкт дослідження. Методи проектування, використання технології для створення мобільних додатків для Android та iOS.

Предмет дослідження. Інструменти, бібліотеки та фреймворки, що дозволяють створювати мобільні додатки для Android та iOS.

Задачі дослідження:

- Розглянути поняття, що таке мобільний додаток.
- Проаналізувати та обрати інструменти.
- Розробити прототип та дизайн додатку.
- Визначити моделі даних та інтегрувати бізнес логіку у проект.
- Створити мобільний додаток для ОС Android та IOS, що буде задовольняти меті роботи.

Робота складається зі вступу, трьох розділів зі своїми підрозділами, висновку та списку літературних джерел кількістю 20 найменувань та 31 рисунку. Загальний об'єм роботи 47 сторінок.



РОЗДІЛ 1

ПОНЯТТЯ МОБІЛЬНОГО ДОДАТКУ

Даний розділ містить опис поняття «мобільний додаток», окреслені їх переваги та недоліки.

1.1 Поняття мобільного додатку

Мобільний додатком є програмним забезпечення, призначене для роботи на телефонах, планшетах та інших мобільних пристроях. Багато мобільних додатків встановлено на самому пристрої або їх можна завантажити з онлайн магазинів мобільних додатків: Google Play, App Store, Windows Phone Store та інших, безкоштовно або за окрему плату.

Насамперед мобільні додатки використовували для швидкої та зручної перевірки електронної пошти, тощо. Про високий попит призвів до розширення їх функціоналу та призначень зовсім в інших галузях, приклад ігри, GPS, спілкування, перегляд відео та доступ до інтернету.

Мобільні додатки переслідують, щонайменше дві явні цілі: не дати користувачу нудьгувати та спростити побут.

Необхідність мобільного додатку – недоречно навіть обговорювати, через те, що поширення доступних мобільних пристроїв, зручного сенсорного екрану, вільний, або взагалі безкоштовний, вихід у мережу інтернет, легкий та швидкий обмін інформацією – збільшило аудиторію потенційних клієнтів будь-якого бізнесу.

1.2 Сучасні мобільні додатки

Всі мобільні пристроїв продаються з уже встановленим набором додатків, наприклад набір додатків від Google, а саме Gmail, Google Drive, Youtube, Google Chrome та інші. Попередньо встановлені додатки, можуть бути вилучені з мобільного пристрою, але не завжди. За це відповідає компанія, що виробляє мобільний пристрій.

Користувач може сам встановлювати необхідні йому додатки через платформи їх розповсюдження. Платформою розповсюдження є магазин додатків. Перші магазини з'явилися у 2008 році, та зазвичай, керуються компанією, що розробляє та володіє мобільною операційною системою, наприклад: Apple - App Store, Google - Google Play, Windows - Windows Phone Store. Але існують незалежні магазини додатків, такі як Cydia на iOS, Get Jar або F-Droid на Android. Існують платні та безкоштовні мобільні додатку. Більшість додатків є безкоштовними для користувачів.

Переваги мобільних додатків:

- **Взаємодія з користувачами.** У мобільним додатку є можливість відправляти Push-повідомлення, на відміну від сайтів. Недолік сайтів полягає у тому що, користувачі дізнаються про щось нове (наприклад новий товар) лише тоді, коли зайдуть на сайт. У мобільних додатках цю проблему вирішують push – повідомлення. Вони надають змогу відправляти усім користувачам безкоштовні повідомлення, без обмеження по символам.
- **Локальні сповіщення** працюють однаково, лише встановлюються на самому пристрої. Наприклад, якщо користувач зробив якусь дію у додатку, яка очікує від нього реакції через певний час, додаток може нагадати йому про це.
- **Широкі можливості зворотного зв'язку** – користувачі можуть залишити повідомлення, відгуки у магазинах додатків, відправляти повідомлення через додаток або у соцмережах і вони автоматично транслюватимуться на усі ресурси.
- **Інтерфейс.** Всі елементи управління: кнопки, текстові поля, посилання мають бути зручними для натискання на мобільному. Мобільні операційні системи мають для цього свою спеціальну логіку.
- **Навігація.** Мобільна операційна система має свою логіку переходів між екранами у додатках. Користувач кожної мобільної операційної системи звик до тієї ж поведінки та логіки переходів у кожному додатку. Проте навігація ж на кожному сайті зроблена по-своєму, і потративши на черговий сайт, потрібно

кожного разу шукати кнопки «Ок», «Назад», «Скасування», тощо, яких взагалі може й не бути.

- Рівень персоналізації. Однією з ключових причин бурхливого зростання мобільних додатків є те, що мобільні телефони знають про свого власника дуже багато, і використовують цю інформацію для покращення рівня сервісу. Мобільні додатки мають можливість запам'ятовувати дані користувача і змінювати інтерфейс додатку в залежності від його потреб.

- Дозволи. Користувач сам вирішує якому додатку надати дозволи, а якому ні. Наприклад дозволи до доступу мережі інтернету, камери, акселерометра, компаса, барометра, тощо.

- Локація. Мобільний додаток може визначити поточне місце розташування користувача і, наприклад, відобразити список ресторанів, до яких користувач має найменшу відстань.

- Робота в офлайн режимі. Користувачі не завжди мають доступ до якісної та швидкої мережі інтернет. Навіть при наявності мобільного інтернету, його якість в середньому гірша, ніж якість домашньої чи офісної мережі інтернет. Якщо поставлено завдання, щоб користувач не втрачав зв'язок з додатком відразу, як тільки обірвався інтернет-зв'язок і міг далі користуватись додатком, то розробка саме мобільного додатку з режимом роботи, який не потребує доступу до мережі інтернет – це єдине можливе рішення;

- Ознака престижу. Розробка мобільного додатку компанією є ознакою того, що компанія дбає про своїх клієнтів та користувачів, даючи їм свободу вибору платформи користування та вищий рівень комфорту, наприклад, коли потрібно побудувати тривалий і якісний зв'язок з користувачами.

Недоліки мобільних додатків:

- Встановлення мобільних додатків. Якщо розмістити QR-коди на сайт і на мобільний додаток, то переходів на сайт буде завжди більше ніж встановлень мобільного додатку. Більшість користувачів після переходу за посиланням на додаток у магазин, не встановлюють його.

- Платформа. Кожний мобільний додаток створюється під певну мобільну операційну систему і може бути запущений тільки на цій операційній системі, на відміну від сайту, який може бути запущений на будь-якій платформі, або пристрої.

- Оновлення. Оновлення мобільного додатку в магазинах завжди займає деякий час на їх перевірку адміністрацією магазину на відміну від сайту. Після оновлення сайту користувачі образу можуть їм користуватись

- Вартість. Вартість розробки й підтримки мобільного додатку майже завжди вище, ніж розробка та підтримка роботи сайту.

- Маркетинг. Мобільні додатки просувати, значно, дорожче аніж сайт. Ціна одного нового користувача безпосередньо залежить від порогу його входження (наприклад вік, користування іншими додатків, країна проживання), чим він вищий - тим вища ціна.

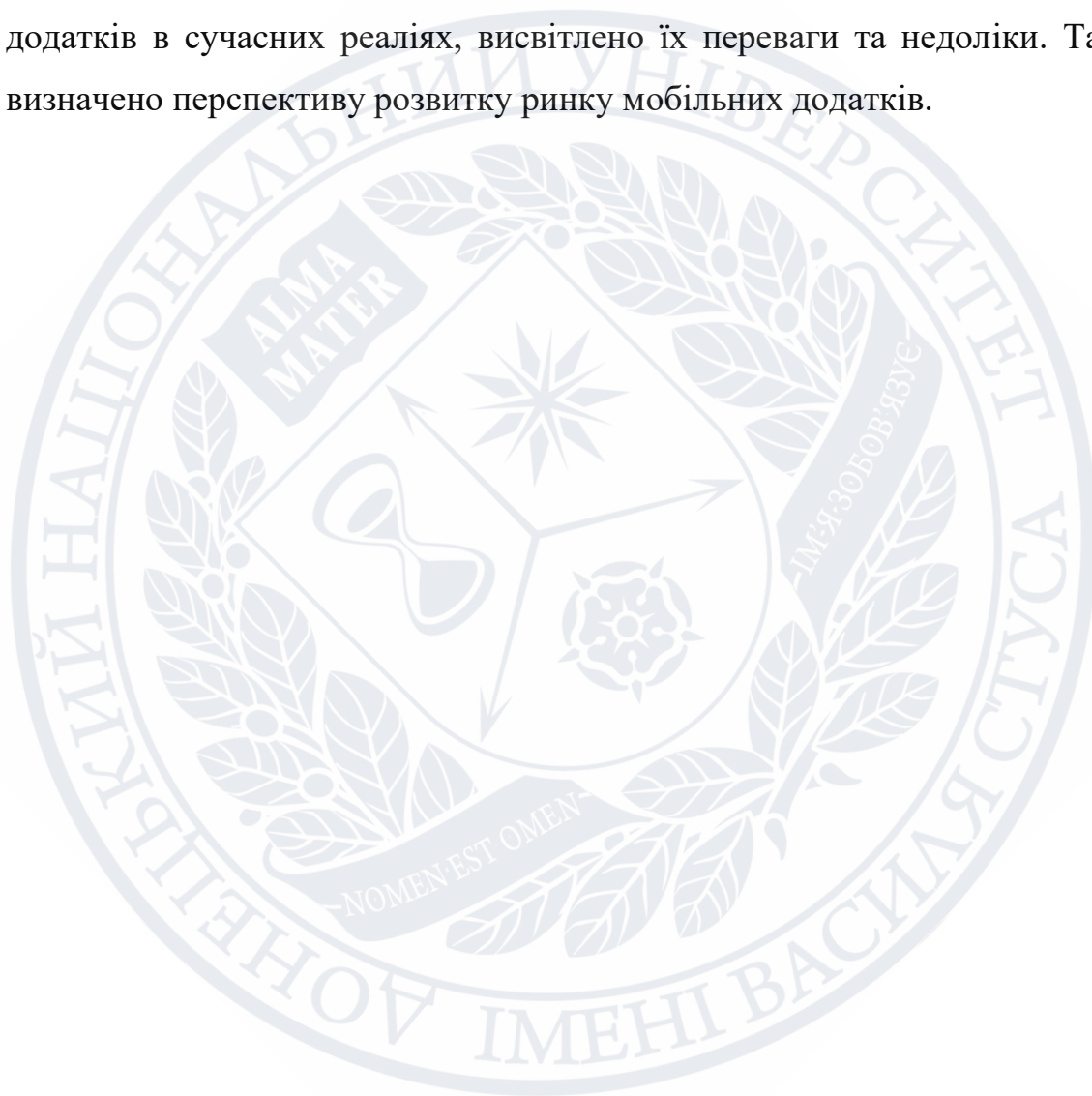
Спершу мобільні додатки були інструментами для оперування та контролю загальних потоків інформації, враховуючи календар, електронну пошту, контакти, інформацію про погоду та фондовий ринок. Так чи інакше, наявність інструментів та попит для розробників зумовили швидке поширення додатків і для інших категорій електронних пристроїв, які функціонують за допомогою настільних прикладних програм. Різноманіття та велика кількість мобільних додатків призвела до виникнення значної кількості пізнавальних ресурсів про них, а саме: блоги, журнали та спеціальні служби виявлення додатків в Інтернеті з відгуками, рекомендаціями та оглядами.

Мобільні додатки серед користувачів мобільних пристроїв стають все більш популярними. Відповідно до даних дослідження компанії App Annie за 2017 рік кількість завантажень додатків зросла на 60%, витрати споживачів зросли більш ніж удвічі. Час, витрачений на додаток кожним користувачем складає близько 43 дні на рік. Узагальнена кількість завантажень додатків у 2019 році зросла до 115 млрд: 84 млрд - в Google Play і 31 млрд - в App Store. Дослідники виявили, що

використання мобільних додатків чітко корелюється з їх наповненням та залежить від саме локації розміщення користувача і часу доби.

Висновок до розділу 1

В цьому розділі було розглянуто загальне поняття мобільних додатків та сучасних мобільних додатків. Визначено необхідність використання мобільних додатків в сучасних реаліях, висвітлено їх переваги та недоліки. Також було визначено перспективу розвитку ринку мобільних додатків.



РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ ТА ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ

Даний розділ присвячено постановці задачі дослідження, а також опису необхідних інструментів для її реалізації.

2.1 Постановка задачі

Необхідно розробити макет та прототип мобільного додатку. Розробити моделі даних, інтегрувати бізнес-логіку та реалізувати логіку спілкування додатку із сервером, застосовуючи мову програмування Dart та фреймворк Flutter.

Завданням даної роботи є розробка клієнтської частини мобільного додатку та реалізувати бізнес логіку навчальної системи «Get Simple». Мобільний додаток буде надавати можливість користувачам створювати курси, купувати або відвідувати безкоштовні курси, знаходити матеріал для саморозвитку і все це в комфортних умовах віддаленого користування.

Завдяки цьому додатку люди зможуть, не виходячи з дому, що на даний момент є актуальним, підвищувати кваліфікаційні навички, навчатися чомусь новому, а особи, що володіють знаннями або ж цікавим матеріалом, зможуть викладати свої курси для її поширення.

Старт користування платформою, як і на всіх інших. Користувач має зареєструватися на платформі, після чого він повинен обрати, який функціонал йому потрібен «Студент» або ж «Викладач». У цих 2 типів є свої відмінності.

Функціонал, що надається Студентові:

- Реєстрація.
- Перегляд усіх курсів на платформі.
- Перегляд обраних користувачем курсів.
- Перегляд вже завершених курсів.
- Пошук курсів.

- Отримання сповіщень.
- Відправка запитів на участь у різних курсах.
- Перегляд особистого профілю.
- Відкрити деталі курсу.
- Додавання курсу у архів.

Функціонал , що надається Викладачеві:

- Реєстрація.
- Створення курсів.
- Публікація курсів.
- Прийняття запитів від студентів на участь у курсі.
- Перегляд усіх курсів.
- Перегляд усіх своїх курсів.
- Отримання сповіщень.
- Перегляд особистого профілю.

2.2 Огляд існуючих аналогів

Coursera – Онлайн-платформа «Coursera» розпочала роботу в 2012 році, в цьому ж році в розділі «Освіта» рейтингу веб-сайтів журналу «Time» Coursera посіла перше місце. Лідируючу позицію майданчик забезпечив собі багато в чому завдяки динаміці розвитку: список організацій-партнерів постійно розширюється, кількість курсів, а з ним і студентів зростає. З Coursera співпрацюють не лише освітні установи США (Стенфорд, Принстон, Каліфорнійський, Колумбійський університети), університети окремих штатів (Теннессі, Нью-Йорк, Небраска), але і провідні культурні організації (наприклад, Музей природознавства США).

Udemy – Сайт MOOC-платформи «Udemy» має російськомовну версію, що робить навігацію по порталі максимально комфортною. Освітні проекти розбиті на шістнадцять категорій, серед яких є комп'ютерні, гуманітарні дисципліни, а також хобі та рукоділля, мистецтво і фотозйомка. Курси Udemy читають

викладачі, які мають великий практичний досвід у бізнесі, менеджменті, фінансах, тощо. Джеж Уелч, генеральний директор General Electric й топ-менеджером з колосальним управлінським досвідом, розробив один з найбільш популярних курсів.

Canvas Network – Проект Canvas Network відрізняється великою різноманітністю курсів, які проводять абсолютно різні за рівнем підготовки та напрямку діяльності люди: доктора наук, менеджери, письменники. Курси не мають єдиного підходу до викладу. Матеріал можуть пояснювати виключно в коротких відеолекціях, доповнювати можливістю обговорювати прослуханий на форумі з викладачем та іншими студентами, сертифікат по закінченні курсу можуть видавати чи ні.

Після аналізу аналогічних платформ було вирішено створити власну платформу із більш сучасним дизайном та схожим функціоналом для надання можливості у період COVID-19 здобувати освіту маючи лише телефон з інтернетом.

2.3 Огляд дизайнерських інструментів

Figma - це графічний онлайн редактор для сумісної роботи декількох користувачів. Її можна використовувати для створення прототипу сайту, інтерфейсу програми та обговорити правки з колегами в реальному часі.

Чому була обрана саме Figma:

1. Екосистема. Створюючи веб-сайт або мобільний додаток, дизайнерам часто доводиться завантажувати великі файли на хмарні сховища, пересилати їх поштою, тощо. Це забирає багато часу, крім того, файли можуть займати досить багато місця в пам'яті комп'ютера. Саме цю проблему вирішує Figma, через те що усі робочі файли знаходяться на власному хмарному сховищі, дизайнер може просто відправити посилання на файл замовнику або розробнику. Крім того, якщо на проекті змінюється фахівець, не виникне проблем з тим, де знаходяться вихідні дані.

2. Спільне редагування. Набір додатків Google Docs відмінно показали, що одночасна робота декількох людей над одними файлами полегшує комунікацію і покращує результат. Використовуючи Figma дизайнери, розробники, менеджери і замовники можуть одночасно ставити запитання, писати коментарі та редагувати дизайн.
3. Багатозадачність. Якщо потрібно працювати одночасно з декількома робочими областями то Sketch та Photoshop це не дуже оптимальний вибір, оскільки розробники досить часто скаржаться на їх продуктивність. Figma дозволяє дизайнерам та розробникам працювати з більш, ніж відкритими десятима проектами одночасно.
4. Прототипування. Переваги цього продукту найкраще розкриваються при розробці дизайну мобільних додатків. Якщо проект складається з більш, ніж 30 екранів, досить складно переходити між ними. Але завдяки Figma можна зібрати всі екрани в одному місці і керувати ними. Figma досить корисна і для розробників, вона дозволяє безпосередньо в програмі подивитися інформацію про компоненти, яка потрібні необхідно розробити [2].

2.4 Мова програмування Dart

Dart – це мова програмування від компанії Google, яка призначена в першу чергу для розробки веб-додатків (як на клієнтській так і на серверній частині) та для мобільних додатків. Це означає, що одну і ту ж програму написану на Dart можна компілювати під різні платформи - Windows, Android чи iOS [3]. Dart є об'єктно-орієнтованою мовою програмування. Всі значення, які використовуються у Dart, представляються як об'єкти [4]. Smalltalk, Java та JavaScript досить сильно впливали на розвиток Dart. Окрім цього синтаксис досить схожий на синтаксис C - подібних мов програмування.

2.5 Інструменти для розробки

Flutter – безкоштовний і відкритий набір засобів розробки мобільного користувацького інтерфейсу, створений компанією Google. Простіше кажучи, за допомогою Flutter можливо створити власне мобільний додаток з одним масивом коду. Це означає, що для створення двох додатків (IOS і Android) можна використовувати єдину мову програмування і одну базу коду[5].

Flutter націлений на дві важливі речі:

- SDK (Software Development Kit): набір інструментів, який допоможе у розробці додатків. Також він містить можливості для компіляції коду в нативному машинному коді (код для IOS і Android).
- Framework (Бібліотека призначеного для користувача інтерфейсу на основі віджетів): Колекція функціональних елементів призначеного для користувача інтерфейсу (кнопок, текстових вводів, повзунків і т.д.), які можна персоналізувати під особисті переваги.

Для розробки з Flutter використовується мова програмування під назвою Dart. Це також мова від Google, але значно поліпшена в останні роки [6]. Dart фокусується на розвитку верстки веб-сторінок; його можна з легкістю використовувати для створення мобільних і веб-додатків.

4 основні причини використовувати Flutter для MVP:

- Розробити мобільний додаток використовуючи Flutter більш дешево, оскільки не потрібно створювати і підтримувати два окремих мобільних додатки (під Android та iOS).
- Для створення MVP досить одного розробника.
- Це ефективно: неможливо помітити різницю між нативним додатком і додатком Flutter.
- Це красиво: можна легко використовувати віджети, що надаються Flutter, і персоналізувати їх для створення оригінального призначеного для користувача інтерфейсу для ваших клієнтів.

GetX – це надлегке і потужне рішення для Flutter. Воно поєднує в собі високопродуктивне управління станом, інтелектуальне впровадження залежностей, управління маршрутами швидким і практичним способом.

GetX має наступні базові принципи, які є пріоритетом для всіх ресурсів в бібліотеці[5]:

- GetX сфокусований на продуктивності і мінімальному споживанні ресурсів.
- GetX використовує простий і приємний синтаксис. Не має значення, що розробник хоче зробити, завжди є більш легкий спосіб з GetX. Це заощадить час розробки та забезпечить максимальну продуктивність, яку може забезпечити програму.
- GetX дозволяє повністю розділити уявлення, логіку уявлення, бізнес-логіку, впровадження залежностей і навігацію.
- GetX НЕ роздутий. Він має безліч функцій, які дозволяють почати програмувати, ні про що не турбуючись, але кожна з цих функцій знаходиться в окремих контейнерах і запускається тільки після використання. Якщо ви використовуєте тільки управління станом, то буде скомпільовано тільки управління станом. Якщо ви використовуєте маршрутизацію, то нічого з управління станом не буде скомпільовано. Таким чином кожне рішення GetX було спроектовано, щоб бути надлегким у використанні. Також в цьому є і заслуга Flutter, який вміє усувати невживані ресурси, як жоден інший фреймворк [7].

Ця архітектура має три головні стовпи :

- Керування станом.
- Керування маршрутом.
- Керування залежностями.

Саме завдяки останньому пункту у списку в додатку реалізовано впровадження залежностей або Dependency Injection

Dependency Injection – цей вислів вперше використано в статті Мартіна Фаулера *Inversion of Control Containers and the Dependency Injection Pattern*. Інверсія управління – концепція, що лежить в основі впровадження залежності. Це означає, що клас не повинен конфігурувати свої залежності статистично, а повинен бути налаштований іншим класом ззовні.

Clean Architecture – це підхід, при якому кодова база поділена на декілька рівнів з певними правилами залежності: внутрішній рівень не має нічого знати про зовнішні рівні і залежати від них. Завдяки цьому залежності вказуються всередині кожного рівня, щоб дозволяє прибрати залежність між рівнями (шарами) [11].

Clean Architecture, робить код:

- Незалежних від фреймворків.
- Тестованим.
- Незалежних від UI.
- Незалежних від Баз даних.
- Незалежним від будь-якого зовнішнього впливу.

Stacked Themes – що містить в собі набір віджетів та класів, які допомагають розробнику керувати декількома темами у Flutter додатку. Ця бібліотека містить базові класи, які спрощують керування темами.

Бібліотека використовує клас `Theme Manager` для керування всіма функціями теми. Перед запуском програми необхідно виконати ініціалізацію `Theme Manager`. Необхідно замінити основну функцію на «future» функцію й зробити її асинхронною щоб дочекатись результату від `Theme Manager`, що його успішно проініціалізовано.

Клас `Theme Builder` має властивість під назвою `themes` яка містить в собі об'єкти класу `Theme Data`. Зазвичай додатки не мають велику кількість тем, найчастіше мають світлу й темну, або ще нейтральну. Але `Theme Builder` зберігає список тем, що дає змогу зробити нескінчену кількість тем. Опис тем рекомендують зберігати в окремому файлі, оскільки теми мають досить багато

налаштувань, саме через це опис певної теми додатку може займати досить велику кількість строк коду.

Firebase – це платформа полегшення розробки різних видів додатків з великою кількістю функціоналу. Починалася вона як звичайний стартап, але на сьогодні це платформа яка використовується для розробки мобільних додатків. Головна перевага цієї платформи в тому, що вона дозволяє розробнику не відволікатися на створення серверної частини додатку. І це спрощує й прискорює створення мобільних додатків, надає розробникам можливість повністю зосередитися саме на розробці UX та UI компонентів, а саме на призначеному для користувачів інтерфейсу. Саме зв'язка Firebase з Flutter SDK дозволяє, наприклад, розробникам компанії AVADA MEDIA створювати швидкі та потужні мобільні додатки під Android і iOS, які дозволяють вирішувати найрізноманітніші задачі [14].

Платформа Firebase – це BaaS-рішень (Backend as a Service), яке дає розробникам велику кількість додаткового функціоналу й можливостей. Наприклад хмарний сервер, два види бази даних, хостинг сайту, авторизація в одній платформі. Якщо є необхідність синхронізувати дані між клієнтом і базою даних, то у розробників є можливість використовувати Firebase Realtime Database. Мобільний додаток підключається до бази даних завдяки технології WebSocket, яка відповідає за отримання даних протягом усього часу використання додатку.

Також Firebase є сховища для файлів користувачів. Firebase Storage надає досить надійне завантаження і зберігання файлів для мобільного додатка.

Крім цього платформа Firebase надає ще досить багато функцій, таких як аналітика використання додатку, відправка повідомлень та звіти про аварійне завершення роботи додатку, тощо. Надання таких функцій спрощує розробку й підтримку мобільного додатку [15].

Android Studio – «IDE» інтегроване середовище розробки, що використовується для розробки мобільних додатків під платформу Android. Ця IDE перебувала у вільному доступі починаючи з версії 0.1, яку було

опубліковано в травні 2013. Android Studio перейшла до стадію бета-тестування у червні 2014 року. У грудні 2014 року було випущено першу стабільну версію 1.0, тоді ж припинили підтримку плагіна Android Development Tools для IDE Eclipse [16].

Android Studio спроектовано на платформі IntelliJ IDEA, розробленою компанією JetBrains. Android Studio є офіційним засобом розробки Android додатків. Це середовище розробки є доступним і безкоштовним для використання на операційній системі Windows, macOS та Linux. У травні 2017 на конференції Google I/O, Google анонсували підтримку мови програмування Kotlin, яка на сьогодні використовується в Android Studio, як офіційна мова програмування для розробки мобільних додатків під платформу Android.

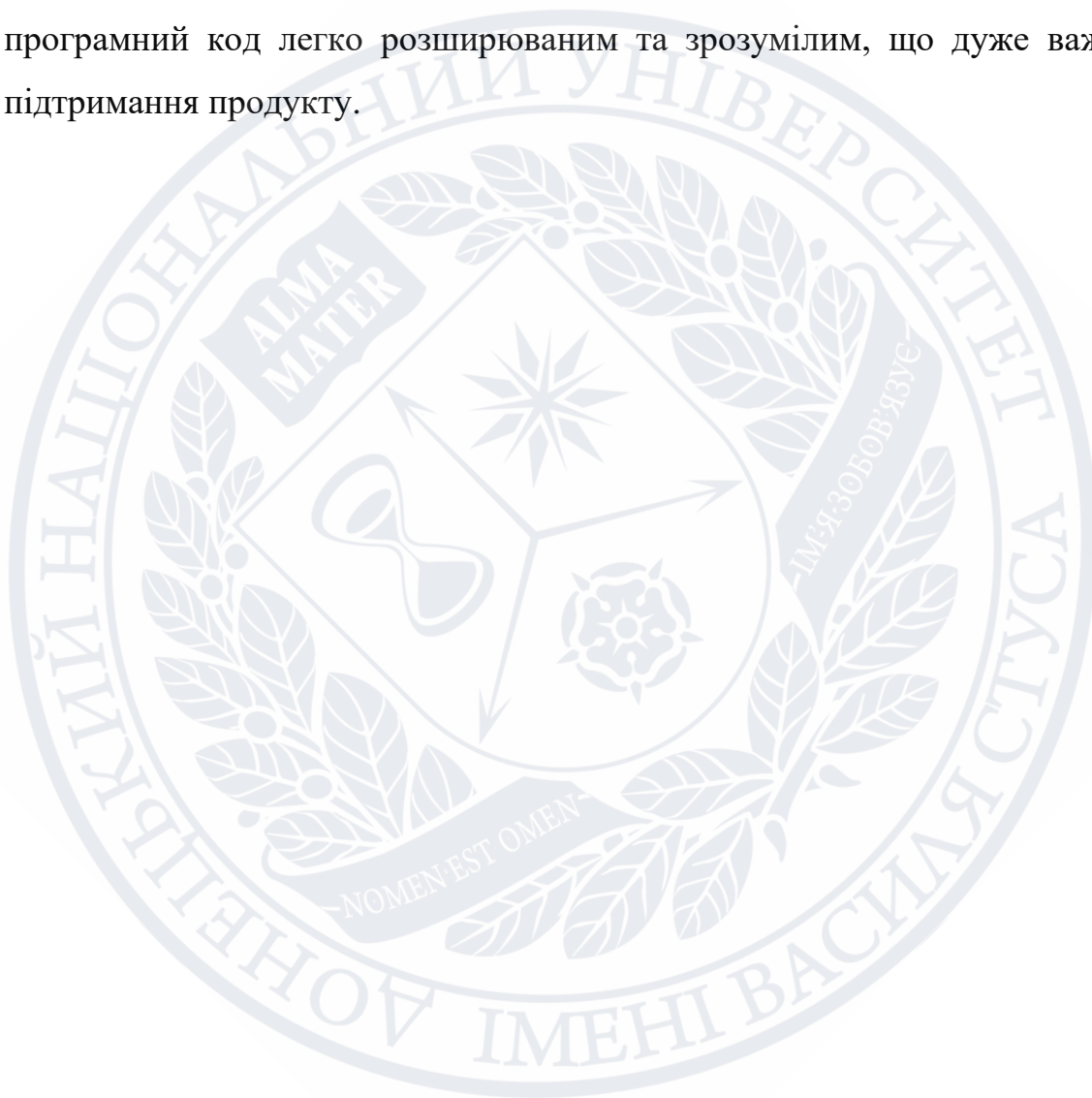
Робота Android Studio постійно покращується й додаються нові функції. На даний момент IDE Android Studio має в собі наступні функції:

- WYSIWYG. Розширений редактор макетів, який може працювати з UI елементами за допомогою Drag-and-Drop (перетаскування). Також доступна функція попереднього перегляду UI на різних розмірах та розширеннях екрану.
- Gradle. Система збірки додатку.
- Підтримка генерації Apk та App Bundle файлів.
- Форматування коду.
- Android Lint. Статичний аналізатор коду, який дозволяє знаходити проблемні й несумісні з різними версіями ділянки коду.
- ProGuard.
- Генерація й використання сертифікатів для підпису додатків.
- Шаблони використання UI компонентів.
- Розробка додатків для носимих пристроїв Android Wear.
- Розробка додатків для телевізорів на Android TV.
- Підтримка Google Cloud Platform та Firebase, яка включає в себе інтеграцію з сервісами Authentication, Firebase Cloud Messaging, Analytics, тощо.

- Вбудований Android Emulator завдяки якому можна збирати та тестувати проекти на різних версіях Android й на різних розмірах екрану

Висновок до розділу 2

В цьому розділі розглянуто всі інструменти, які були обрані для розробки. Обрання правильних інструментів значно спрощує розробку та робить програмний код легко розширюваним та зрозумілим, що дуже важливо для підтримання продукту.



РОЗДІЛ 3

РОЗРОБКА КЛІЄНТСЬКОЇ ЧАСТИНИ МОБІЛЬНОГО ДОДАТКУ

Даний розділ присвячено опису розроблюваного мобільного додатку.

3.1 Створення прототипу

Першим кроком у створенні мобільного додатку був прототип. Він потрібен для того, або розробник міг чітко розуміти як має працювати програма, зробити так, щоб усі кроки у додатку були інтуїтивно зрозумілі користувачеві[17].

Прототип було створено для двох різних функціоналів. Під час розробки прототипу було виділено та зображено спільний функціонал для студента та викладача:

- Реєстрація.
- Перегляд курсів.
- Перегляд особистого профілю.
- Отримання сповіщень.

Було виділено окремий функціонал для студента:

- Перегляд усіх курсів на платформі.
- Перегляд обраних користувачем курсів.
- Перегляд вже завершених курсів.
- Пошук курсів.
- Відправка запитів на участь у різних курсах.
- Відкрити деталі курсу.
- Додавання курсу у архів.

Також було виділено окремий функціонал для викладача:

- Створення курсів.
- Публікація курсів.
- Прийняття запитів від студентів на участь у курсі.

- Перегляд усіх курсів.
- Перегляд усіх своїх курсів.

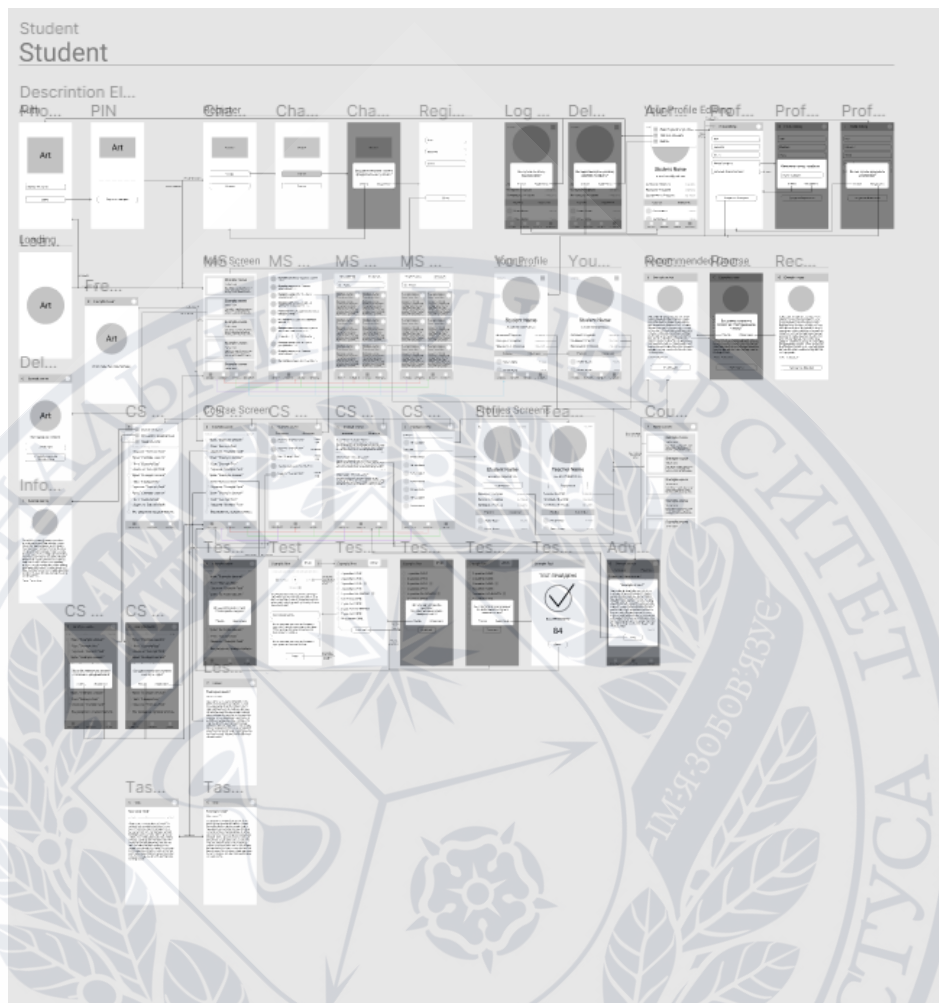


Рисунок 3.1 – Скріншот прототипу для Студента

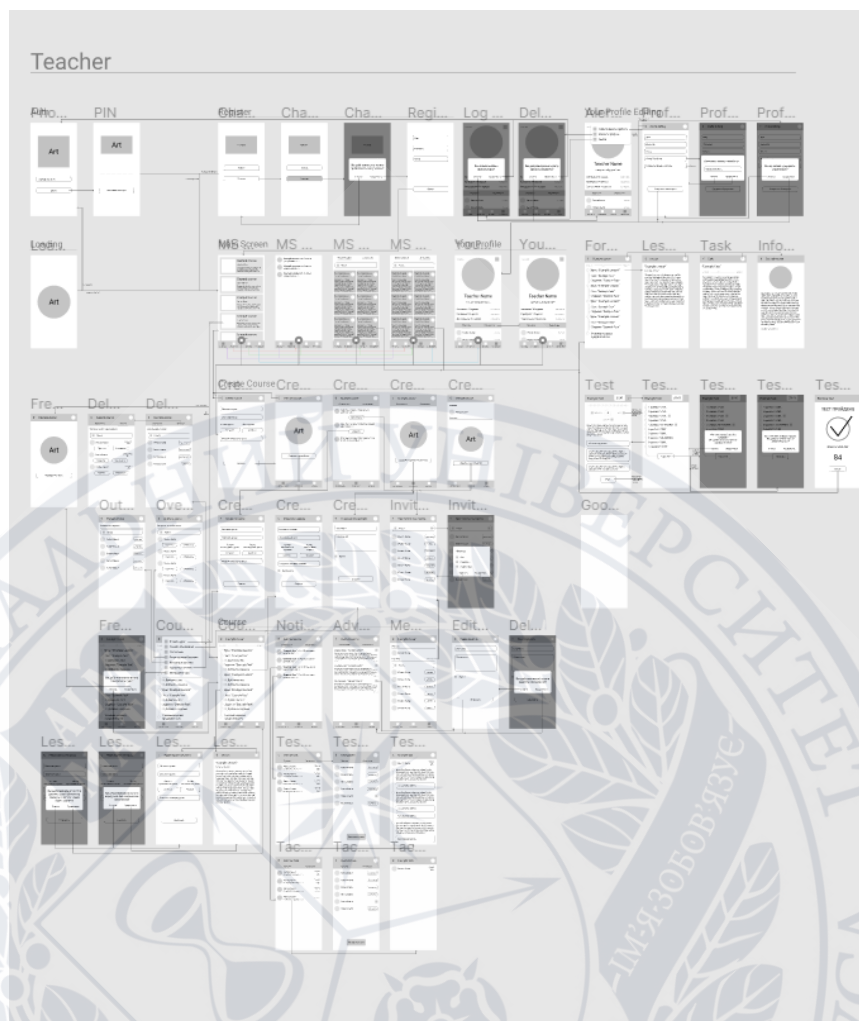


Рисунок 3.2 – Скріншот прототипу для Викладача

3.2 Створення дизайну проекту

При розробці UI дизайну мобільного додатку було взято до ваги ефект «естетика-юзабіліті» (aesthetic-usability effect). Ефект «естетика-юзабіліті» (aesthetic-usability effect) стосується схильності користувачів сприймати додаток з привабливий інтерфейс як більш корисний і зручний. Отже UI дизайн мобільного додатку має включати в себе красу, зручність та гнучкість [17].

Залежно від мети проекту та призначення інтерфейсу, було підібрано кольорову схему і оформлення. При розробці було взято до уваги те як кольори впливають на людей з точки зору психології. Також при розробці UI інтерфейсу колір можна використовувати деякі кольори, для того щоб зроби акцент на якість програми. Наприклад червоний колір має асоціацію з силою та пристрастю,

зелений – з лісом і прибутком, синій – з миром та спокоєм, фіолетовий – з творчістю і знаннями. Зокрема останнім трендом дизайну є розробка додатків, які могли б працювати на Android та iOS. Проте UI дизайн слід відокремити розробку дизайну під кожен платформу. У результаті це зменшить кількість помилок і некоректного функціонування.

Дві найпотужніші функції Figma – це компоненти та стилі. Вони дозволяють повторно використовувати об'єкти і атрибути інтерфейсу, щоб підтримувати дизайн в масштабі. Коли потрібно щось змінити, наприклад, колір посилання бренду або домашню іконку, можна внести зміну один раз - в вихідний основний компонент або стиль - і подивитися, як він оновиться в усіх ваших проектах [18].

Компоненти та стилі можна використовувати:

- В межах одного окремого файлу в безкоштовній версії Figma.
- Для різних файлів і проектів в пакеті Figma Professional.
- Між командами в пакеті Figma Organization.

Отже було винесено та описано усі стилі компонентів для світлої на темної теми, які використовуються у додатку.

При створенні, найменування та управлінні компонентами і стилями потрібно враховувати нескінченну кількість нюансів. Було розроблено максимально зрозумілу логіку користування додатком.

Згідно прототипу, було прийняте рішення розробляти дизайн проекту у двох темах – темна та світла. Розробка дизайну почалась з опису стилів та компонентів які будуть використовуватись під час дизайну і подальшої розробки:

- Типографіка для світлої на темної теми.
- Палітра кольорів.
- Іконки.

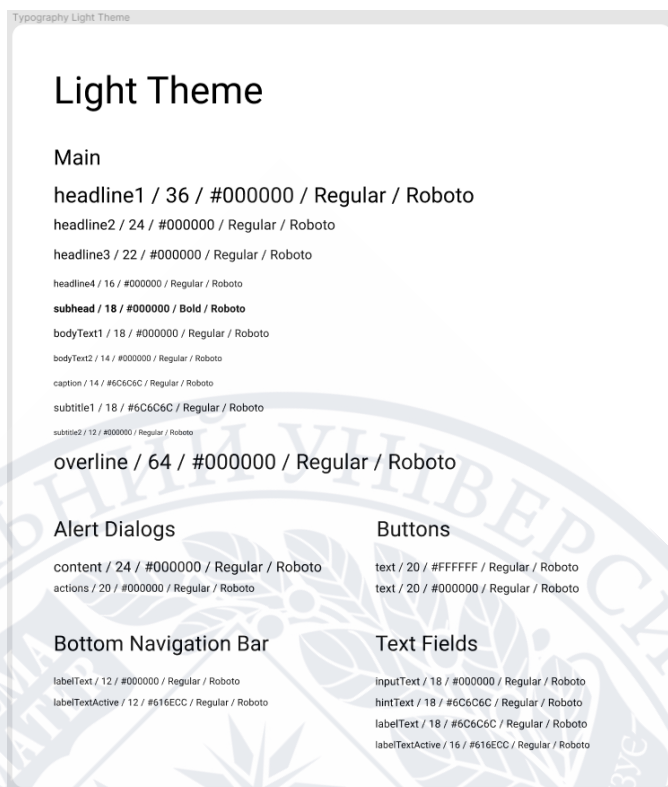


Рисунок 3.3 – Скріншот типографіка для світлої теми



Рисунок 3.4 – Скріншот типографіка для темної теми

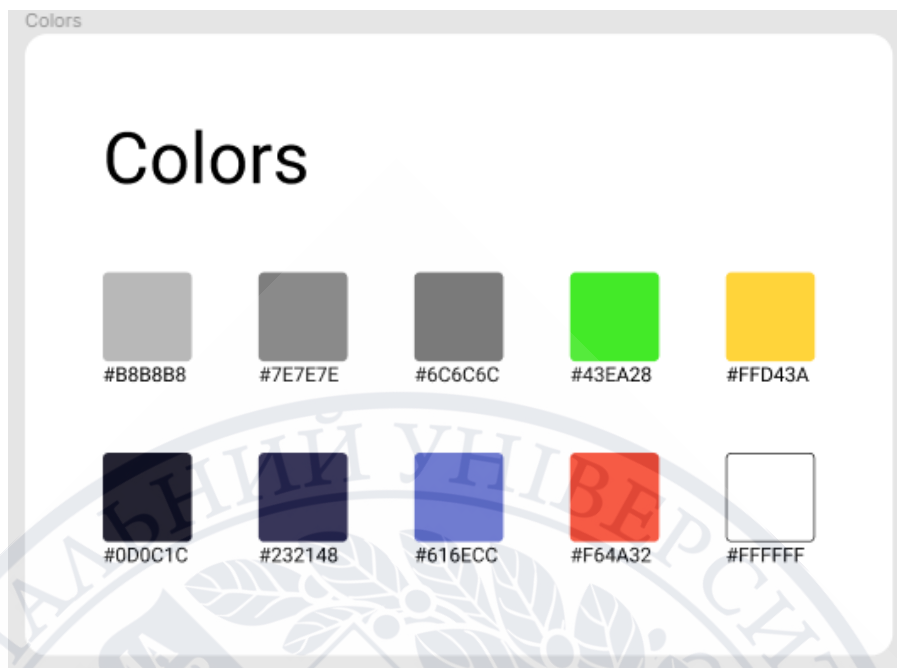


Рисунок 3.5 – Скріншот палітри кольорів

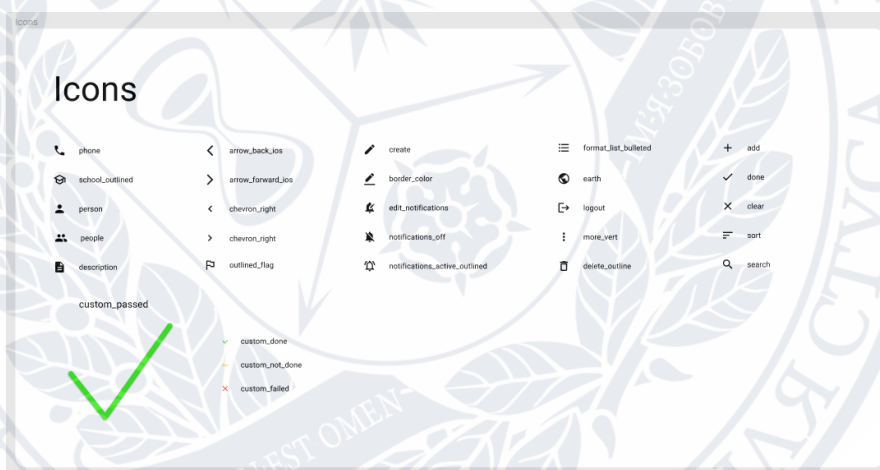


Рисунок 3.6 – Скріншот усіх іконок, що використовуються

Після опису компонентів та стилів розпочалася робота над розробкою дизайну UI інтерфейсу. Перед UI інтерфейсом стояло завдання забезпечити користувача коректною взаємодією з додатком, надання естетичного задоволення під час користування.

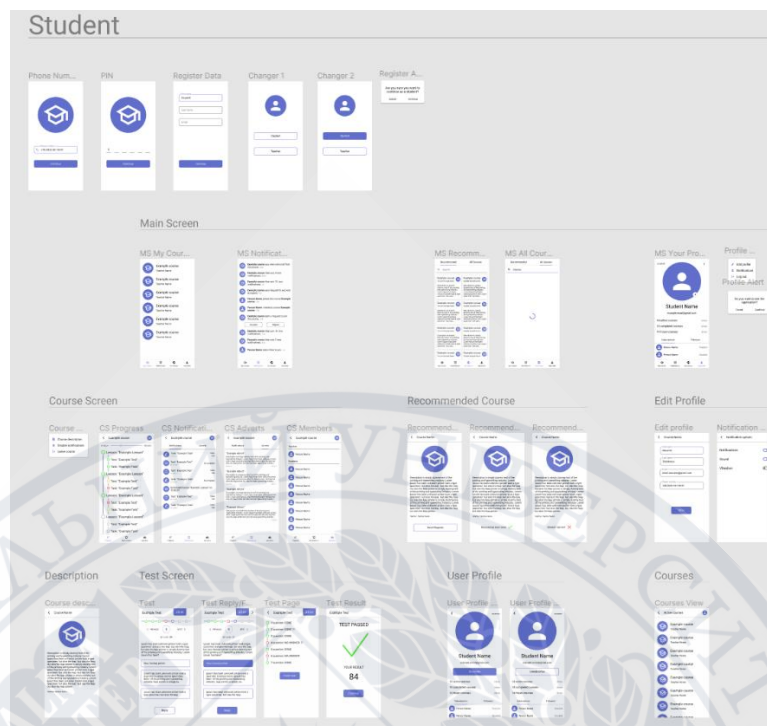


Рисунок 3.7 – Скріншот дизайну світлої теми для Студента



Рисунок 3.8 – Скріншот дизайну світлої теми для Викладача

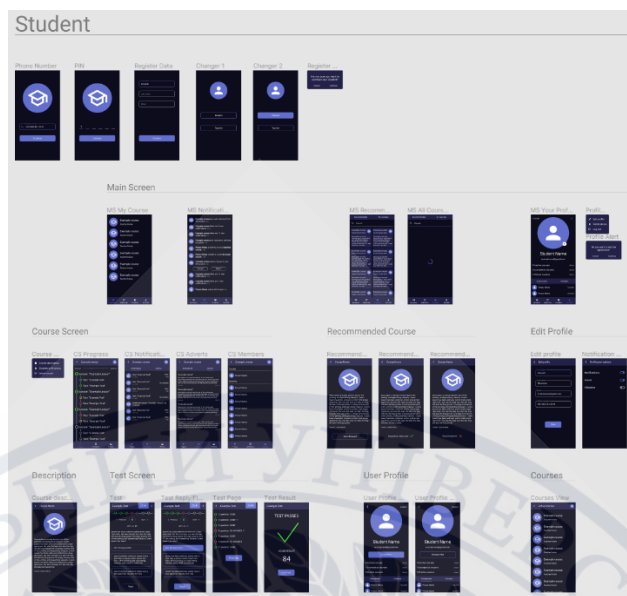


Рисунок 3.9 – Скріншот дизайну темної теми для Студента



Рисунок 3.10 – Скріншот дизайну темної теми для Викладача

3.3 Створення архітектури проекту

Архітектуру було обрано GetX з розмежуванням на модулі відповідальності як було описано в Clean Architecture.

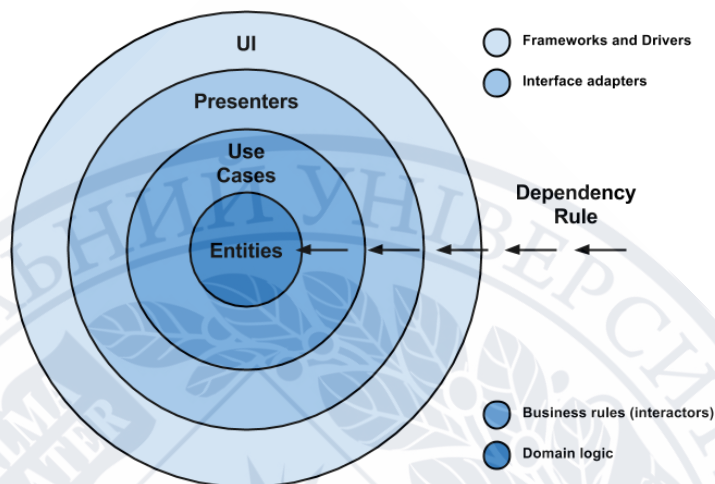


Рисунок 3.11- Структура правильних залежностей у додатку по Clean Architecture

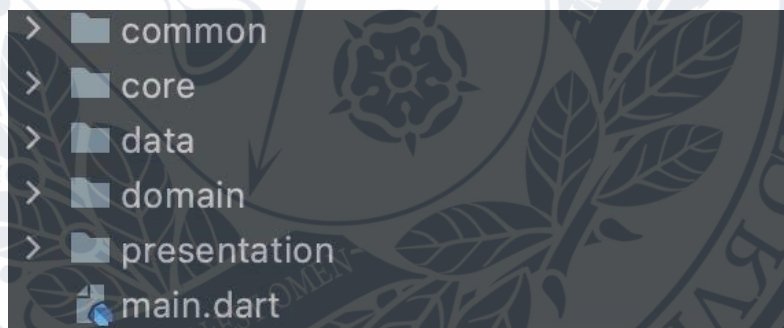


Рисунок 3.12- Структура папок у проекті

Папка common містить в собі функції, класи або якісь дані які не прив'язані до якогось певного шару програми. Папка core містить в собі усі базові класи, які використовуються у проекті. Папка data та domain відповідають за бізнес логік і використовують шаблон repository для доступ до даних.

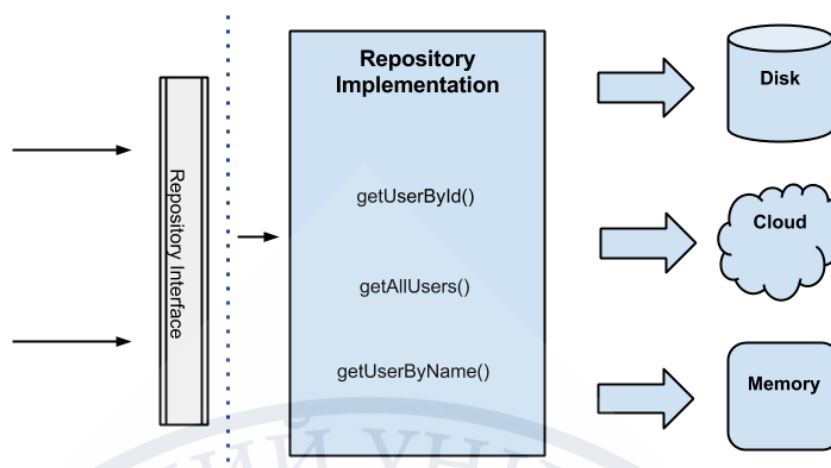


Рисунок 3.13- Структура роботи шаблону repository

Папка presentation містить в собі усі класи, які пов'язані з UI. Main.dart – файл запуску програми.

Отже додаток було розбито на модулі і ці модулі зроблено відносно незалежними один від одного.

3.4 Створення UI мобільного додатку

Flutter SDK використовує віджети для створення ієрархічної структури та формування структуру екрану. З цієї причина ця структура має вигляд дерева в якому перший віджет це його корінь. При цьому віджет може об'єднувати з собі декілька інших віджетів.

Формування «дерева віджетів» (Widget tree) необхідне лише для полегшення розуміння, тому що програмісти використовують віджети, проте Flutter немає «дерева елементів» (tree of elements). Кожен віджет відповідає за один елемент. Ці елементи пов'язані один за одним і формують дерево. Відповідно до цього кожен елемент це посиланням на що-небудь у дереві [19].

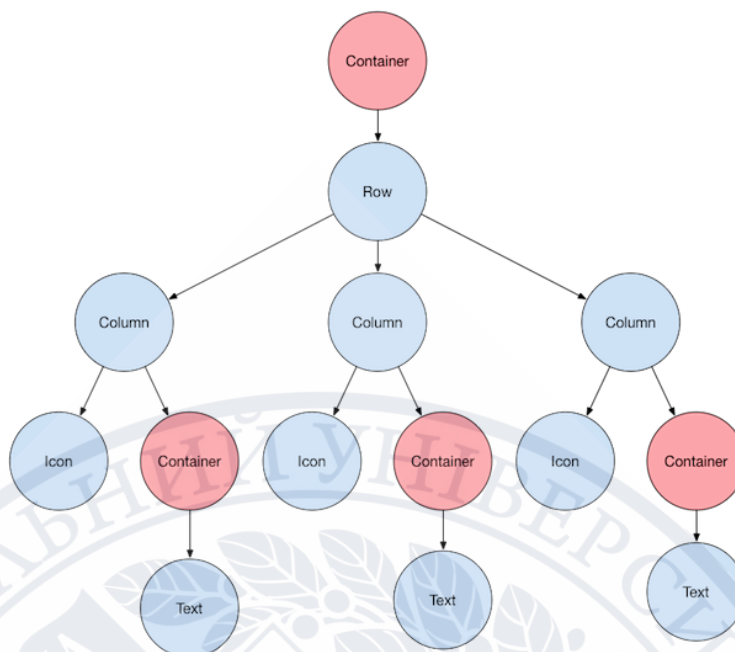


Рисунок 3.14 – структура ієрархії екрану у Flutter

Створивши архітектуру, яка поділена на незалежні модуль і отримавши знання стосовно формування «дерева віджетів» у Flutter було розроблено UI мобільного додатку у двох темах використовуючи готові «віджети» [20]:

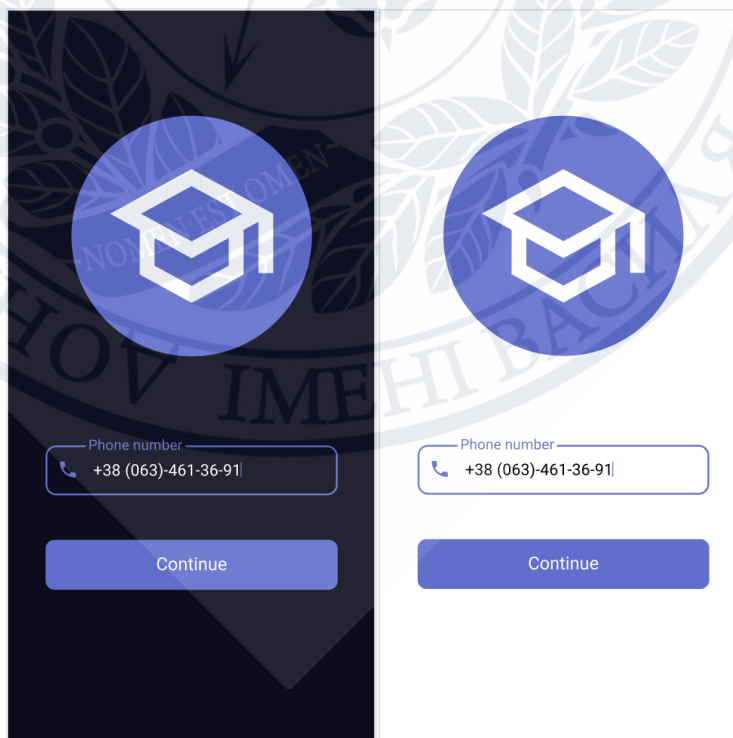


Рисунок 3.15- UI реєстрації у двох тема



Рисунок 3.16- UI вводу коду з смс у двох темах



Рисунок 3.17 – UI вводу додаткових даних у двох темах

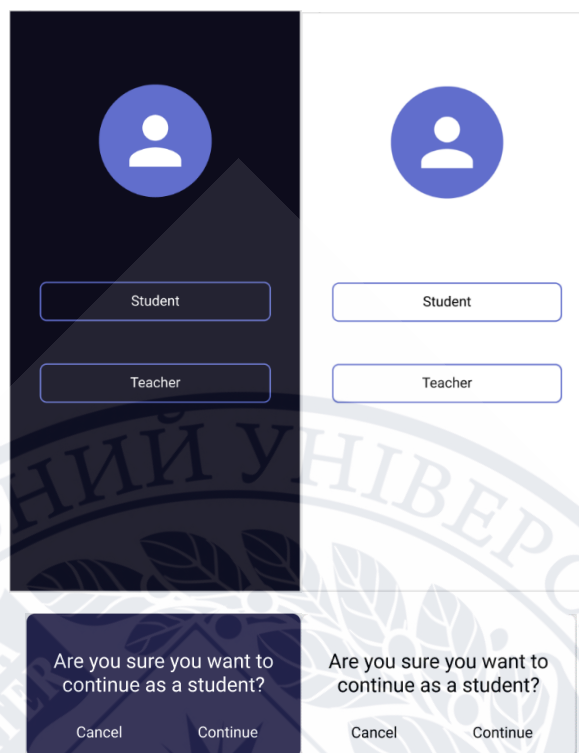


Рисунок 3.18- UI вибору типу аккаунта у двох темах



Рисунок 3.19- UI «Мої курси» у двох темах

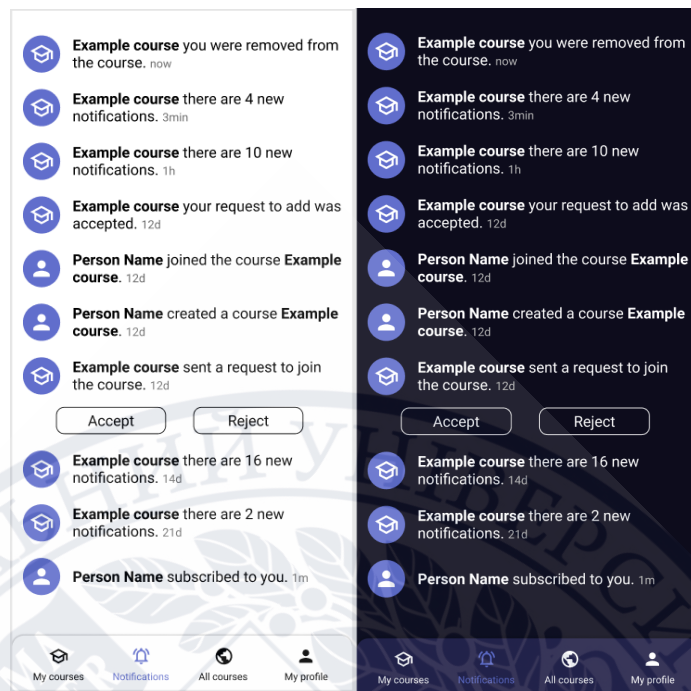


Рисунок 3.20- UI повідомлень у двох темах

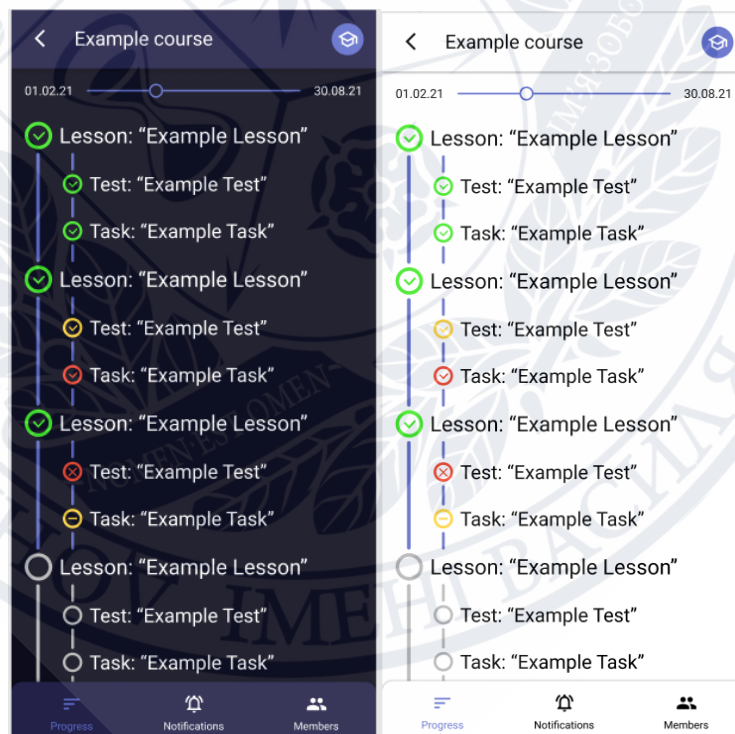


Рисунок 3.21 – UI прогресу учня за курсом у двох темах

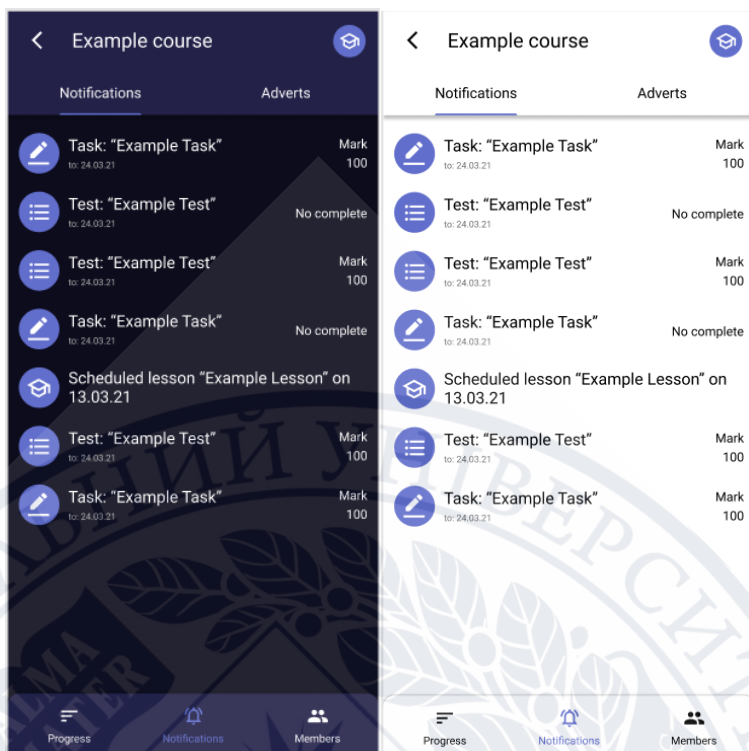


Рисунок 3.22- UI повідомлень за курсом у двох темах

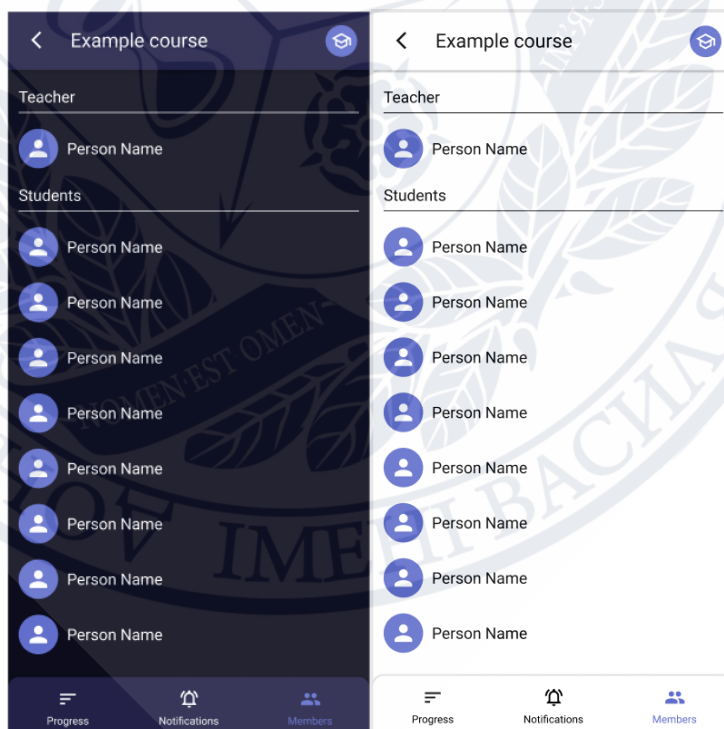


Рисунок 3.23- UI учасників курсу у двох темах

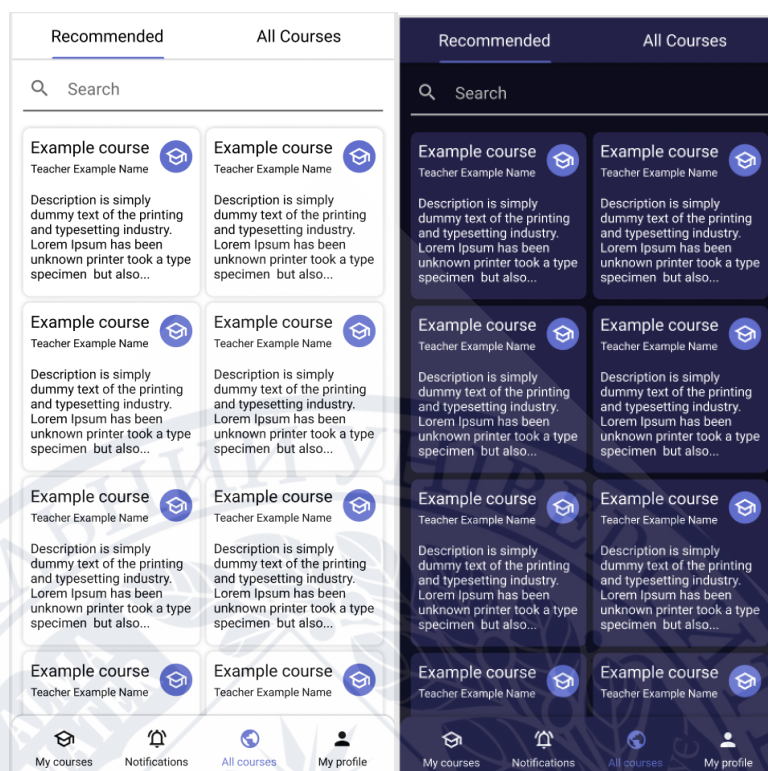


Рисунок 3.24- UI усіх та рекомендованих курсів у двох темах

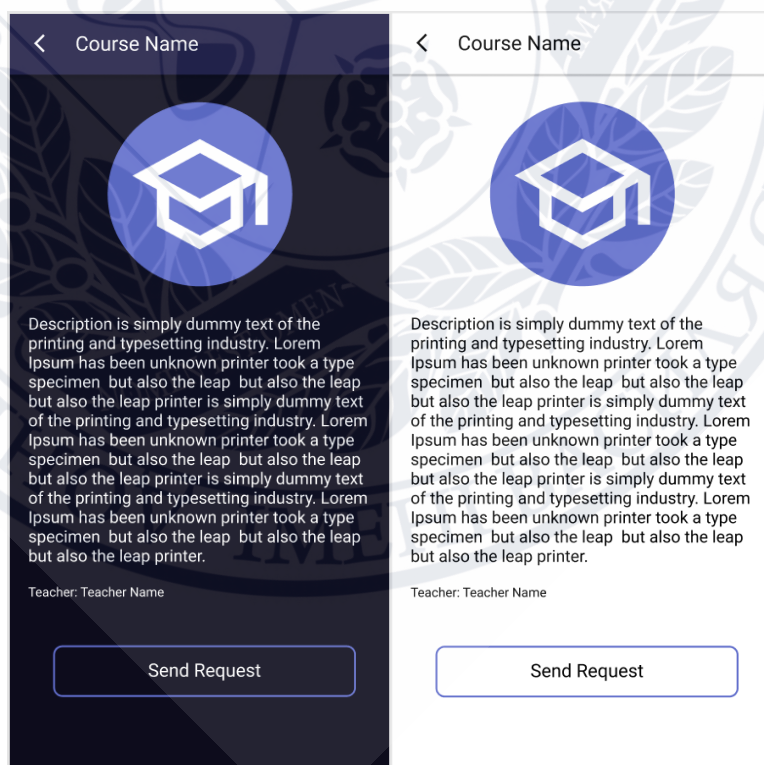


Рисунок 3.25- UI інформації про курс у двох темах

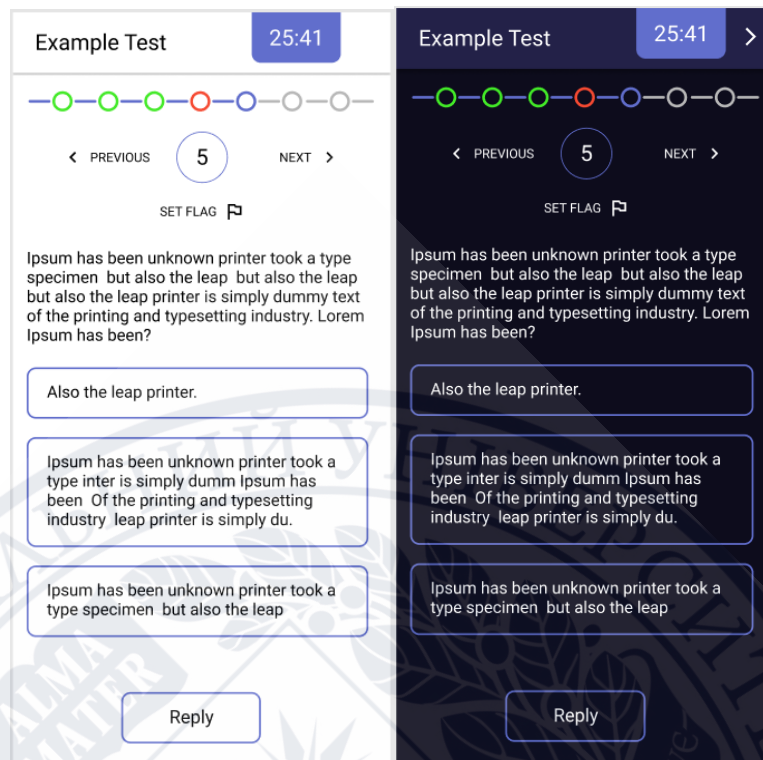


Рисунок 3.26- UI проходження тестування у двох темах

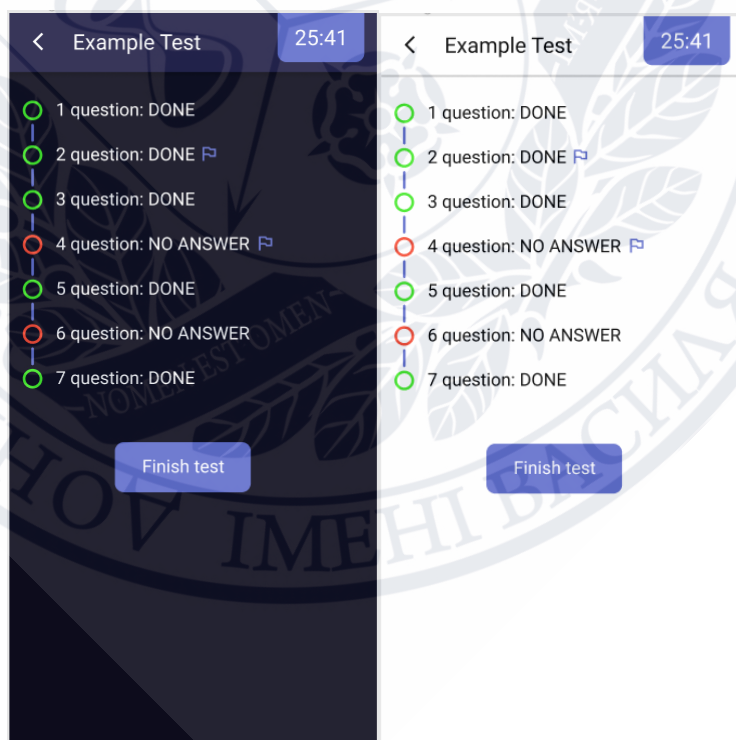


Рисунок 3.27- UI фінальний екран тестування у двох темах

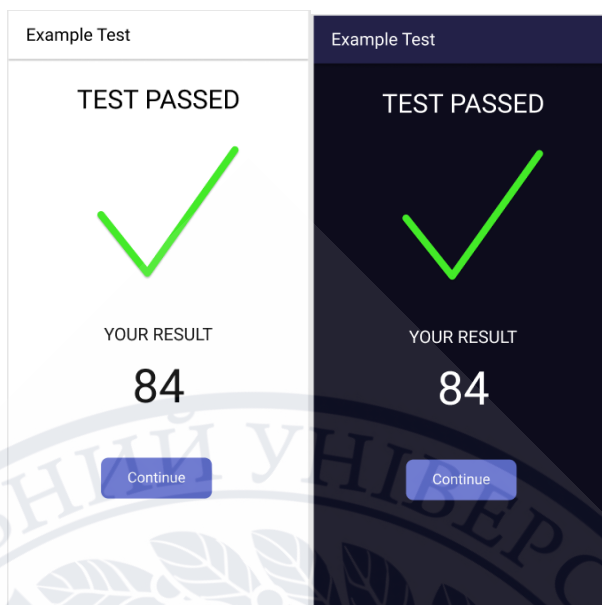


Рисунок 3.28- UI результатів тестування у двох темах

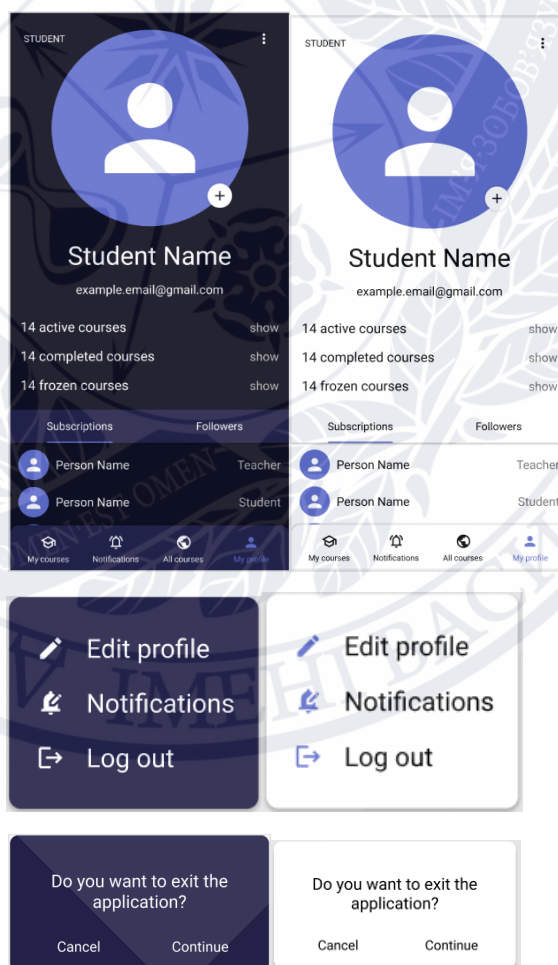


Рисунок 3.29 – UI профілю користувача у двох темах

Two side-by-side mobile app screens showing user profile editing forms. The left screen is titled "Course Name" and the right is "Edit profile". Both screens have a dark blue header with a back arrow. The forms contain fields for First name, Last name, Email, and Phone number, all with pre-filled values. A blue "Save" button is at the bottom of each screen.

Field	Value
First name	Alexandr
Last name	Zherebtsov
Email	email.example@gmail.com
Phone number	+38 (063)-461-36-91

Рисунок 3.30 – UI редагування профіля користувача у двох темах

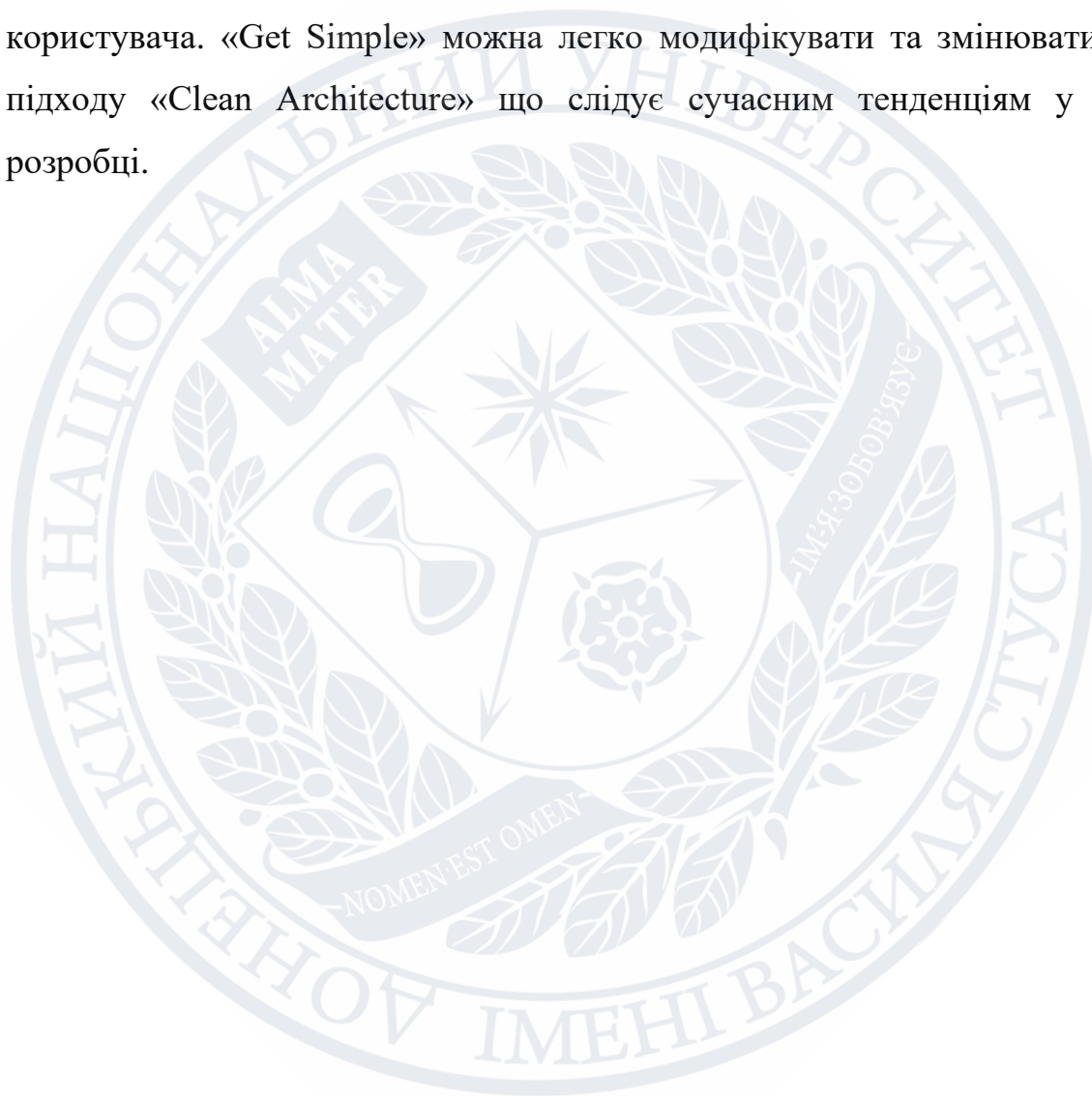
Two side-by-side mobile app screens showing notification options. The left screen is titled "Notification options" and has a dark blue background. The right screen is also titled "Notification options" and has a light gray background. Both screens show three toggle switches for Notifications, Sound, and Vibration, all of which are currently turned on.

Option	Status
Notifications	On
Sound	On
Vibration	On

Рисунок 3.31 – UI налаштування повідомлень у двох темах

Висновок до розділу 3

У результаті за допомогою Flutter SDK було розроблено мобільний додаток «Get Simple». Під час розробки було використано найзручніших і найновіші інструменти та технології, які доступні на сьогодні в pub.dev. В основі цього мобільного додатку використано багато сучасних рішень і бібліотек які доступні у pub.dev, додаток зроблено зрозумілим як для розробника так і для звичайного користувача. «Get Simple» можна легко модифікувати та змінювати завдяки підходу «Clean Architecture» що слідує сучасним тенденціям у мобільній розробці.



ВИСНОВКИ

Актуальність розробки мобільних додатків полягає в тому, що ринок мобільних пристроїв дуже стрімко зростає з кожним роком, відповідно до цього набирає популярності і розробка додатків під мобільні платформи Android та IOS.

Метою бакалаврської роботи було розробити мобільний додаток для онлайн-навчання.

Для того, щоб розпочати розробку мобільного додатку «Get Simple» було проаналізовано необхідність та актуальність розробки платформи для навчання.

Перед розробкою були досліджені та обрані інструменти, за допомогою яких було розроблено мобільний додаток. Мобільний додаток було розроблено для платформ Android та iOS, використовуючи Flutter SDK, як один з найперспективніших інструментів. В якості мови програмування було обрано мову Dart, яку була розроблена компанією Google. Її позиціонують як альтернативу до JavaScript.

Програма Figma була обрана для створення макету та прототипу майбутнього мобільного додатка. Під час розробки UX/UI-дизайну додатку використовувався так зване поняття «естетика-юзабіліті». Це дозволило підібрати графічні компоненти та кольори таким чином, щоб привертати увагу, а дизайн був доступний та інтуїтивно зрозумілий для користувачу.

Архітектура MVVM (Model – View – ViewModel) була використана у додатку «Get Simple». Для реалізації цієї архітектури у Flutter-додатку було використано бібліотеку Stacked.

Було розроблено функціонал додатку: реєстрація та авторизація користувача, можливість перегляду усіх та рекомендованих курсів, створення, редагування та видалення курсів, відправка та підтвердження запиту на проходження курсу, проходження тестування, система повідомлень, редагування профілю.

Виконавши поставлену задачу, була пророблена велика та об'ємна робота по створенню сучасної платформи для навчання. Вона представлена у вигляді мобільного додатку для платформ Android та iOS. Розроблено прототип та дизайн мобільного додатку. Реалізовано увесь запланований UI та усі заплановані функції мобільного додатку.



СПИСОК ЛІТЕРАТУРИ

1. Мобільні додатки у період пандемії COVID - 19. URL: <https://www.appannie.com/en/insights/market-data/mobile-app-usage-surged-40-during-covid-19-pandemic/>
2. 8 причин, щоб полюбити Figma. URL: <https://artjoker.ua/ru/blog/8-prichin-chtoby-polyubit-figma-tak-zhe-silno-kak-my/>
3. Введення в мову програмування Dart. URL: <https://metanit.com/dart/tutorial/1.1.php>
4. Dart documentation. URL: <https://dart.dev/guides>
5. Що таке Flutter. URL: <https://habr.com/ru/post/481326/>
6. Flutter - Beautiful native apps in record time. URL: <https://flutter.dev/>
7. GetX. URL: <https://github.com/jonataslaw/getx/blob/master/README.ru.md>
8. Dependency Injection. URL: <https://habr.com/ru/post/350068/>
9. Dependency Injection In Flutter. URL: <https://medium.com/flutter-community/dependency-injection-in-flutter-f19fb66a0740>
10. Dio. URL: <https://pub.dev/packages/dio>
11. Що таке Clean Architecture. URL: <https://proglib.io/p/clean-architecture-android-apps/>
12. Flutter Clean Architecture. URL: https://pub.dev/packages/flutter_clean_architecture
13. Stacked Themes. URL: https://pub.dev/packages/stacked_themes
14. Що таке Firebase. URL: <https://avada-media.ua/services/firebase/>
15. Firebase helps you build and run successful apps. URL: <https://firebase.google.com/>
16. Android Studio. URL: <https://developer.android.com/studio>
17. Guide to mobile App design. URL: <https://buildfire.com/mobile-app-design/>
18. Guide to components in Figma. URL: <https://help.figma.com/hc/en-us/articles/360038662654-Guide-to-Components-in-Figma>
19. Layouts in Flutter. URL: <https://flutter.dev/docs/development/ui/layout>

20. Material Components widgets. URL:

<https://flutter.dev/docs/development/ui/widgets/material>



Декларація щодо унікальності текстів роботи
та невикористання матеріалів інших авторів без посилань

Зорич Сергій Дмитрович

Прізвище, ім'я, по батькові

Інформаційних і прикладних технологій

Факультет

122 «Комп'ютерні науки»

Шифр і назва спеціальності

«Сучасні інформаційні технології та програмування»

Освітня програма

ДЕКЛАРАЦІЯ

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (бакалаврська) робота на тему: «Розробка клієнтської частини для навчальної системи» є написаною мною особисто.

Одночасно заявляю, що ця робота:

- не передавалась іншим особам і подається до захисту вперше;
- не порушує авторських та суміжних прав, закріплених статтями 21-25 Закону України «Про авторське право та суміжні права»;
- не отримувались іншими особами, а також дані та інформація не отримувались у недозволений спосіб.

Я усвідомлюю, що у разі порушення цього порядку моя кваліфікаційна (бакалаврська) робота буде відхилена без права її захисту, або під час захисту за неї буде поставлена оцінка «незадовільно».

дата

підпис