

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Мобільний додаток для вивчення 3ds Max»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проектування»

Виконавець роботи: студент групи ІТ-72 Пархоменко Олександр Вячеславович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2021 р.

Науковий керівник

(підпис)

к.т.н., доц., Баранова І.В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

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

Студент _____

(підпис)

Суми-2021

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
« ____ » _____ 2021 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Пархоменко Олександр Вячеславович

1 Тема роботи Мобільний додаток для вивчення 3ds Max

керівник роботи Баранова Ірина Володимирівна, к.т.н., доцент

затверджені наказом по університету від « 14 » квітня 2021 р. № 0181-VI

2 Строк подання студентом роботи «1» червня 2021 р.

3 Вхідні дані до роботи _____ технічне завдання на розробку мобільного додатку _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, функціональне і структурне проектування додатку, практична реалізація мобільного додатку

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність роботи, постановка задачі, дослідження аналогів, функціональні вимоги до мобільного додатку, засоби реалізації, моделювання, практична реалізація, тестування, демонстрація роботи

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.02.2021

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	01.02.2021 – 10.02.2021	
2	Огляд існуючих аналогів	10.02.2021 – 20.02.2021	
3	Постановка задачі	20.02.2021 – 28.02.2021	
4	Вибір технологій реалізації	28.02.2021 – 10.03.2021	
5	Структурно-функціональне моделювання	10.03.2021 – 30.03.2021	
6	Розробка інтерфейсу мобільного додатку	30.03.2021 – 05.04.2021	
7	Розробка всіх розділів мобільного додатку	05.04.2021 – 10.04.2021	
8	Наповнення мобільного додатку контентом	10.04.2021 – 25.04.2021	
9	Тестування додатку	20.04.2021 – 25.04.2021	
10	Оформлення документації	25.04.2021 – 10.05.2021	
11	Виправлення знайдених помилок, підготовка до захисту	10.05.2021 – 30.05.2021	

Студент

(підпис)

Пархоменко О.В.

Керівник роботи

(підпис)

к.т.н., доц. Баранова І.В.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Мобільний додаток для вивчення 3ds Max».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 24 найменувань, 3 додатків. Загальний обсяг роботи – 94 сторінки, в тому числі 43 сторінки основного тексту, 3 сторінки списку джерел та 48 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці мобільного додатку для вивчення 3ds Max.

В першому розділі було проаналізовано предметну область, актуальність роботи та аналоги. Також було обрано засоби для реалізації проекту.

В другому розділі було виконано моделювання роботи мобільного додатку, за допомогою методології IDEF та UML.

В третьому розділі, описано процес розробки мобільного додатку. Описано її структуру, процес розробки інтерфейсу, та спосіб навігації у додатку. Також, наведено основні етапи програмної реалізації. Проведено тестування та демонстрацію роботи.

Результатом виконаної роботи є готовий мобільний додаток для вивчення 3ds Max

Ключові слова: мобільний додаток, Android, вивчення, 3ds Max, 3D моделювання, Xamarin, Xamarin.Forms, XAML, C#, Visual Studio.

ЗМІСТ

Вступ.....	6
1 Аналіз предметної області	8
1.1 Огляд останніх досліджень і публікацій	8
1.2 Аналіз програмних продуктів - аналогів.....	10
1.3 Постановка задачі.....	15
2 Проектування програмного додатку	17
2.1 Структурно-функціональне моделювання	17
2.2 Діаграма варіантів використання	20
3 Розробка мобільного додатку	22
3.1 Структура додатку.....	22
3.2 Розробка інтерфейсу.....	24
3.3 Навігація в додатку.....	25
3.4 Кодування проекту	27
3.5 Тестування додатку	32
3.6 Демонстрація роботи.....	34
Висновки.....	42
Список використаних джерел.....	43
Додаток А Технічне завдання.....	46
Додаток Б Планування робіт	54
Додаток В Код основних модулів додатку	65

ВСТУП

У наш час неможливо уявити значиму сферу, в котрій під час конструювання не використовують об'ємний графічний вигляд будь-то звичайної моделі, або великої сцени, з використанням етапів моделювання, текстурування, анімації та ін. Саме спираючись на цей тривимірний вигляд, на стадії проектування, є можливість усунути велику кількість проблем, які могли б виникнути на наступних етапах відтворення цього "макету". Великий вплив візуалізація знайшла у багатьох галузях, але найбільшого поширення отримала у промисловості, медицині та у сфері розваг (кінематограф, комп'ютерні ігри та анімація).

У сфері промисловості головними користувачами є інженери, будівельники та інші спеціалісти. Для даної діяльності використовують програми твердотільного моделювання, наприклад SolidWorks. У цій сфері головна увага випадає на креслення, графіки та можливість використання формул для обчислення, тому рідко є можливість побачити вигляд об'єкту, максимально направленим на реалістичність.

У сфері медицини розвиток іде у двох основних напрямках: томографія та конструюванні протезів. Сучасне тривимірне сканування знаходить утворені дефекти у частинах органів та тканин. Без використання даної технології було б неможливо цього побачити, так як при рентгені вони будуть приховані. За результатами томографії можна створити протез чи імплант, котрі будуть підходити індивідуально для кожного пацієнта.

У сфері розваг тривимірне проектування використовуються у створенні як "одухотворених" персонажів, так і "неживих" об'єктів. Модель створюється за допомогою полігонів, відкритих та закритих ребер і точок, які створюють полігон. Саме завдяки полігонам створюється об'ємний вигляд об'єкту, а більш

детальним він виглядає за рахунок збільшення кількості тих самих полігонів. Така практика зазвичай використовується на головних об'єктах у сцені, а менш значимі мають їх меншу кількість. Це лише перший етап створення цієї конструкції. Наступним кроком створюються текстури для кожного об'єкту та налаштовується світло, для заломлення і віддзеркалення його. Після чого налаштовується рендер, для відтворення моделі та сцени.

У цій темі потрібна дуже велика практика, тому було вирішено створити мобільний додаток, який буде допоміжним провідником у сфері тривимірного моделювання у програмі 3ds Max.

Метою дипломного проекту є розроблення мобільного додатку, який буде допоміжним у вивченні теоретичного та практичного матеріалу тривимірного моделювання у програмі 3ds Max. В додатку кожен користувач зможе вивчити теоретичну інформацію теми з переліку, виконати практичне завдання та перевірити свої знання за допомогою тестів у додатку.

Для досягнення мети, сформовано перелік основних задач:

- Проаналізувати стан питання розробки мобільних додатків для навчання.
- проаналізувати схожі мобільні додатки.
- Обрати програму для реалізації з достатнім для створення функціоналом.
- Розробити інтерфейсу додатку та реалізувати його функціонал.

Результатом дипломної роботи буде продукт, за допомогою якого користувач зможе переглядати теоретичний матеріал, виконувати практичні завдання та перевіряти свої знання за допомогою тестів у створеному мобільному додатку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Мобільні додатки, призначені для освіти, відносяться до списку найефективніших способів вивчення матеріалу [1]. Основна причина в тому, що вони не просто дають інформацію для вивчення предмету навчання в процесі його освоєння, а також пропонують велику кількість методик освоєння та подальшого використання.

У всьому світі схожі проекти просуваються та набирають велику популярність. За допомогою них користувачі різного віку, галузі навчання та професії мають змогу роздобути потрібну для них інформацію як з допоміжного джерела для вивчення матеріалу, так і подальшого його закріплення за допомогою практик та тестів [2].

Мобільні додатки для вивчення будуть цікаві наступним категоріям слухачів:

- учням, студентам та абітурієнтам – вони допомагають набути потрібний матеріал при вивченні предмета та підготуватися до іспитів;
- фахівцям цієї сфери – набути деякі знання та підтримувати кваліфікаційний рівень;
- навчальним закладам – допоможуть встановити матеріал для вивчення учнями або студентами;
- іншим зацікавленим особам, які вирішили освоїти новий матеріал для подальшого становлення фахівцями у цій сфері.

Система освіти у вищих навчальних закладах гарно поєднуються з сучасними інформаційними технологіями. Мобільні додатки надають гарну

можливість створити навчання більш інформативним та інтерактивним. Схожа форма навчання привертає велику аудиторію можливою взаємодією з нею.

Мобільні додатки для навчання дають змогу змінити систему навчання до більш комфортної для студентів у наш час та вивести її на новий рівень. Цей вигляд вивчення поставленого матеріалу для студентів є ефективним за наступними ознаками [3].

- Економія часу та сил – схожі додатки поєднують великий обсяг теоретичної і практичної інформації у смартфоні користувача. Цей вигляд навчання замінює багато статей, книг та ін., матеріал яких розкиданий по багатьом джерелам, в одному мобільному додатку.
- Інтерактивність – перевагою такого навчання є можливість інтерактивної взаємодії з користувачем. Так, при розгляді однієї з теоретичних або практичних тем, він має можливість наглядати за процесом роботи у сфері навчання.
- Мобільність – у користувача таких додатків є можливість навчатися у будь-якому місті та будь-який час, як при відпочинку, так і у дорозі.

Мобільні навчальні програми повинні відповідати наступним вимогам, а саме [4]:

Компактність – інформаційний контент повинен бути стиснутими за обсягом, через імовірність втрати зв'язку у даному середовищі.

Високий рівень мікроергономіки – великий розмір зображення, та якість звуку на мобільному екрані. Невеликий розмір файлу, для зменшення часу його завантаження.

Доступність – навчальний додаток у смартфоні, завантажується для подальшого користування незалежно від місця, у котрому знаходиться майбутній користувач. За рахунок радіусу дії мобільних мереж, можливість навчання у мобільних додатках здійснюється з будь-якого місця та зручного для користувача часу.

Доступ по необхідності – мобільні додатки надають доступ при появі такої необхідності. Тим самим, надають всю необхідну інформацію у час її потреби.

Мобільні додатки для навчання виконують роль помічника студента та частково замінюють викладача у таких моментах, як [1]:

- отримання додаткової інформації та повторення вивченого матеріалу.
- збільшує швидкість отримання інформації.
- змінює процес навчання до більш комфортного за рахунок спрощення сприйняття матеріалу.

Навчання за допомогою мобільних додатків сприяє спрощенню навчання за рахунок комфорту, доступності та взаємодії між студентами та навчальним закладом.

1.2 Аналіз програмних продуктів - аналогів

Зараз є можливість побачити дуже велику кількість довідників та інших мобільних додатків, які допоможуть користувачу покращити навички у тій чи іншій сфері. Наприклад, великої популярності досягли програми, котрі були націлені на сферу медицини, зокрема анатомію. Наразі є можливість побачити велику кількість мобільних додатків, які допоможуть користувачу отримувати знання та навички у цій сфері. Було б доцільно відмітити декілька таких додатків та їхні переваги.

TED – мобільний додаток з переліком лекцій ораторів TED [5]. За допомогою цього додатку, користувач зможе знайти лекцію на цікаву йому тему.

Codeacademy – мобільний додаток, призначений для поліпшення навичок у програмуванні [6]. У переліку цього додатку є як лекції, так і практичні завдання. Додаток сприяє поліпшенню знань для початкових користувачів. Додаток є допоміжним, для вивчення програмування, який вважається одним із

най популярніших. Привертає користувачів він, за рахунок своєї простоти та інтерактивності, де процес навчання проходить у ігровому виді. Заняття у цьому додатку розподілені по групам. Процес проходження завдань закінчується оцінкою за виконання, а кожен наступний рівень стає більш складним.

Visual Anatomy Free – розробники даного мобільного додатку бачать своєю цільовою аудиторією студентів та професорів з анатомії. Цей мобільний додаток має в наявності повну анатомію людського тіла [7]. Переглянути анатомію можна за сортованим списком, який поділений на різні групи. Також, окрім текстової інформації у додатку є ряд аудіо та відео роликів на різні теми по анатомії та фізіології. Головною особливістю даного додатку є так званий режим “тест”, за допомогою якого користувач має можливість перевірити свої знання.

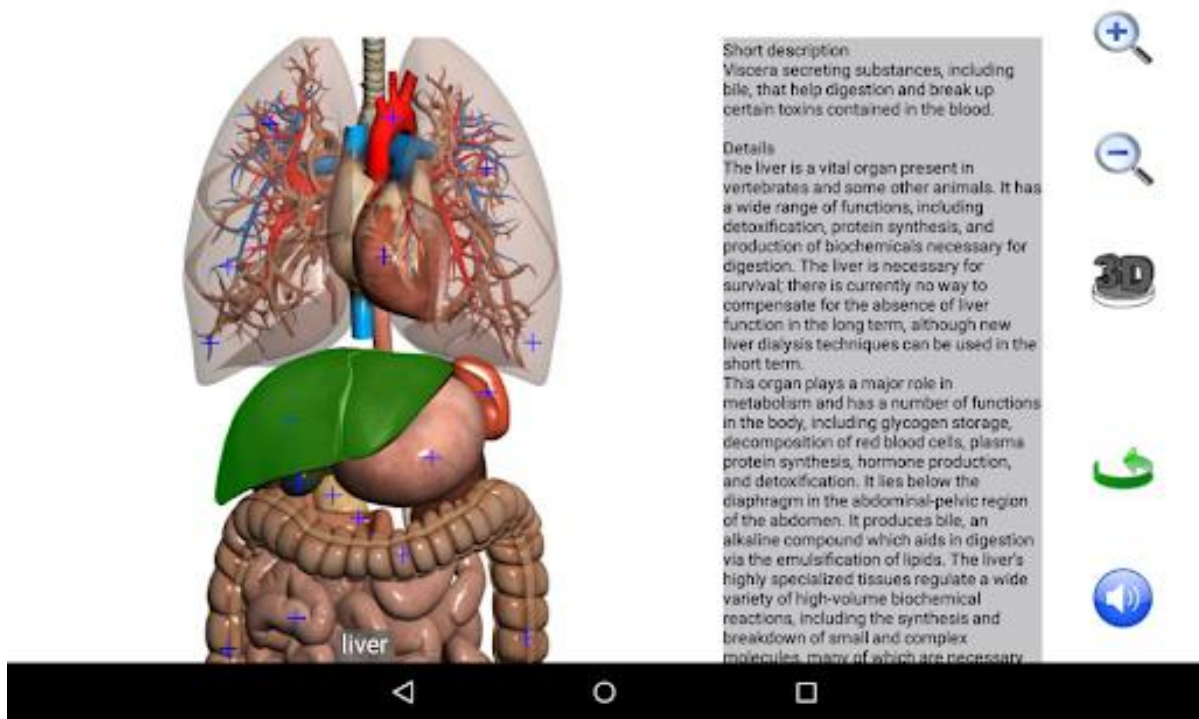


Рис. 1.1 – Мобільний додаток Visual Anatomy Free

Essential Skeleton – у цьому мобільному додатку акцент був сформований на тривимірні моделі скелета людини, з котрою користувач може всіляко

маніпулювати [8]. Наприклад: збільшувати та зменшувати його, повертати на 360 градусів, керувати окремо виділеними кістками, сховати, чи навпаки залишити лише одну або групу кісток на екрані. Окрім цього, у додатку є опис усіх кісток.

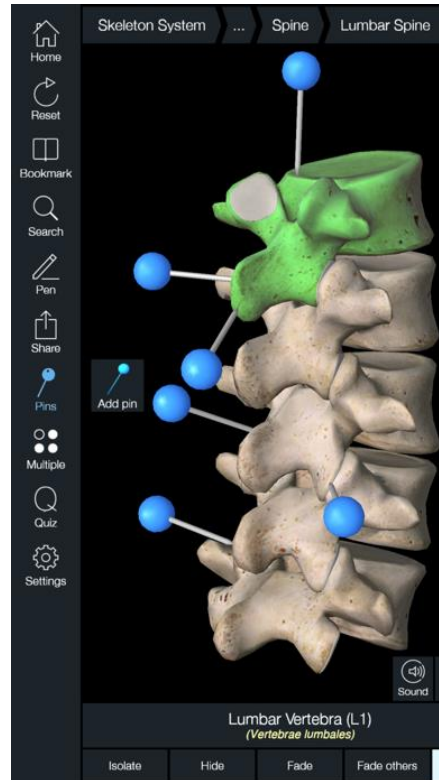


Рис. 1.2 – Мобільний додаток Essential Skeleton

Серед мобільних додатків для навчання був знайдений лише один продукт під назвою "Autodesk 3DS Max Tutorial" [9]. Він має багато недоліків, та був створений лише у вигляді ряду відеороликів, які демонструють лише одну можливість, як побудувати модель, накласти текстуру та інше, текстова інформація з детальним описом можливостей програми та різними можливостями моделювання – відсутня, що лише додає складності у освоєні в даній сфері.

Окрім цього, в даному додатку взагалі відсутнє сортування, всі розділи сортовані у незрозумілому порядку. На рис. 1.3 представлено вікно з розділами у програмі. Як бачимо, першим у списку є розділ "Bake to Texture", який вивчається

лише після освоєння інтерфейсу програми, вміння моделювання різними видами та з початковими навиками розставлення та налаштування світла та камери. У даному прикладі, як продемонстровано у цьому додатку, він лише викликає ряд питань у користувача початкового рівня та відштовхне від цієї сфери своїм не розумінням програми.

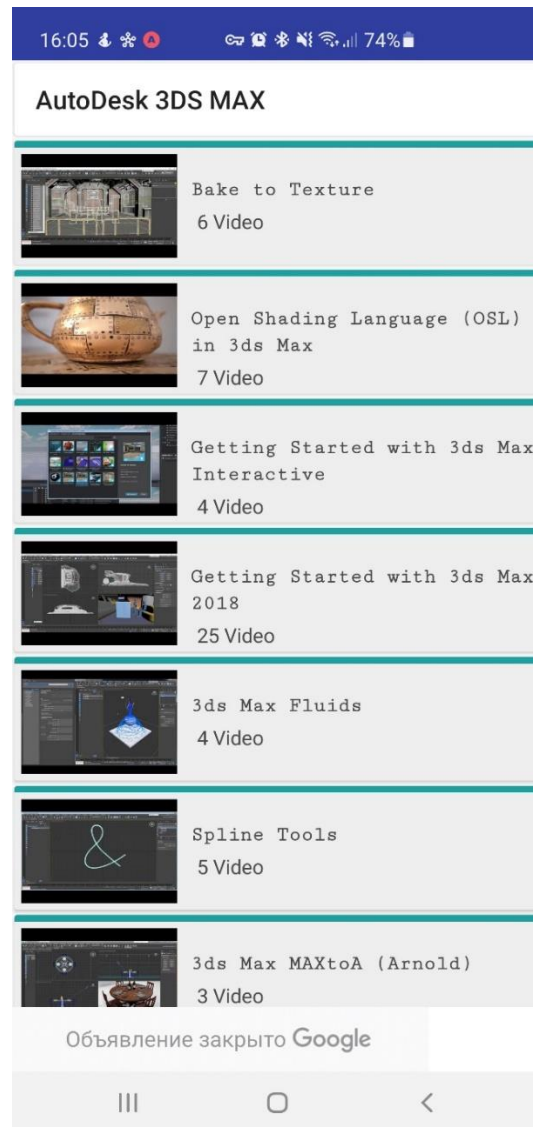


Рис. 1.3 – Розділи у мобільному додатку Autodesk 3DS Max Tutorial

Також, можна зазначити, що відеоролики записані англійською мовою (рис.1.4), що викликає труднощі у людей, які погано володіють англійською мовою.



Рис. 1.4 – Зміст розділу у мобільному додатку Autodesk 3DS Max Tutorial

Переглянувши аналогічні мобільні додатки, можна помітити, що дуже важко знайти мобільний додаток, який буде виконувати роль довідника у тривимірному моделюванні, особливо, коли мова заходить про програму 3ds Max, тому ця тема є актуальною для її подальшої розробки.

1.3 Постановка задачі

Мета роботи - створити мобільний додаток для вивчення 3ds Max, який допоможе користувачу отримати навички роботи з цією програмою, за рахунок теоретичного та практичного матеріалу, а також перевірити рівень вивченого матеріалу за рахунок тестування.

Для реалізації поставленої мети необхідно вирішити такі задачі, як:

- провести аналіз предметної області, дослідити існуючі аналоги, методи рішення проблеми, функціонал, недоліки та переваги, на його основі сформулювати постановку задачі;
- дослідити програмне забезпечення, яке може використовуватися для вирішення існуючої проблеми та обрати програмні засоби для реалізації поставленої мети роботи;
- провести структурно-функціональний аналіз проєкту, побудувати контекстну діаграму та діаграми варіантів використання;
- розробити структуру додатку, графічний інтерфейс користувача;
- наповнити сторінки контентом для вивчення 3ds Max, створити тести для перевірки вивчення матеріалу;
- реалізувати комфортну навігацію у додатку та основний функціонал.

Мобільний додаток повинен виконувати наступні функції:

- Можливість вільного перегляду теоретичного та практичного матеріалу;
- Комфортний перехід між середніми та наступними розділами, під час перегляду теоретичних та практичних розділів;
- Можливість перевірки знань та навичок за допомогою тестів;
- Скачування .docx, .pdf, .max файлів, які знаходяться у практичному розділі.

Програмний продукт буде створюватися за допомогою програмного середовища розробки Visual Studio на платформі Xamarin [10]. Алгоритм програми реалізується на програмній мові C# [11], інтерфейс створюється за допомогою мови розмітки - XAML [12].

Головною перевагою створення мобільних додатків є їхня гнучкість у створенні самого додатку та у його використанні, при одній необхідній умові, а саме у доступі до інтернету.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ

2.1 Структурно-функціональне моделювання

Для опису системи було використано методологію IDEF0. Дана методологія описує побудову ієрархічної системи діаграм, де спочатку описується цілісна система, потім проводиться декомпозиція. При декомпозиції кожна система розбивається на підсистеми, які описуються окремо.

Зобразимо контекстну діаграму користування мобільним додатком рис. 2.1. Наступним кроком виконаємо декомпозицію.

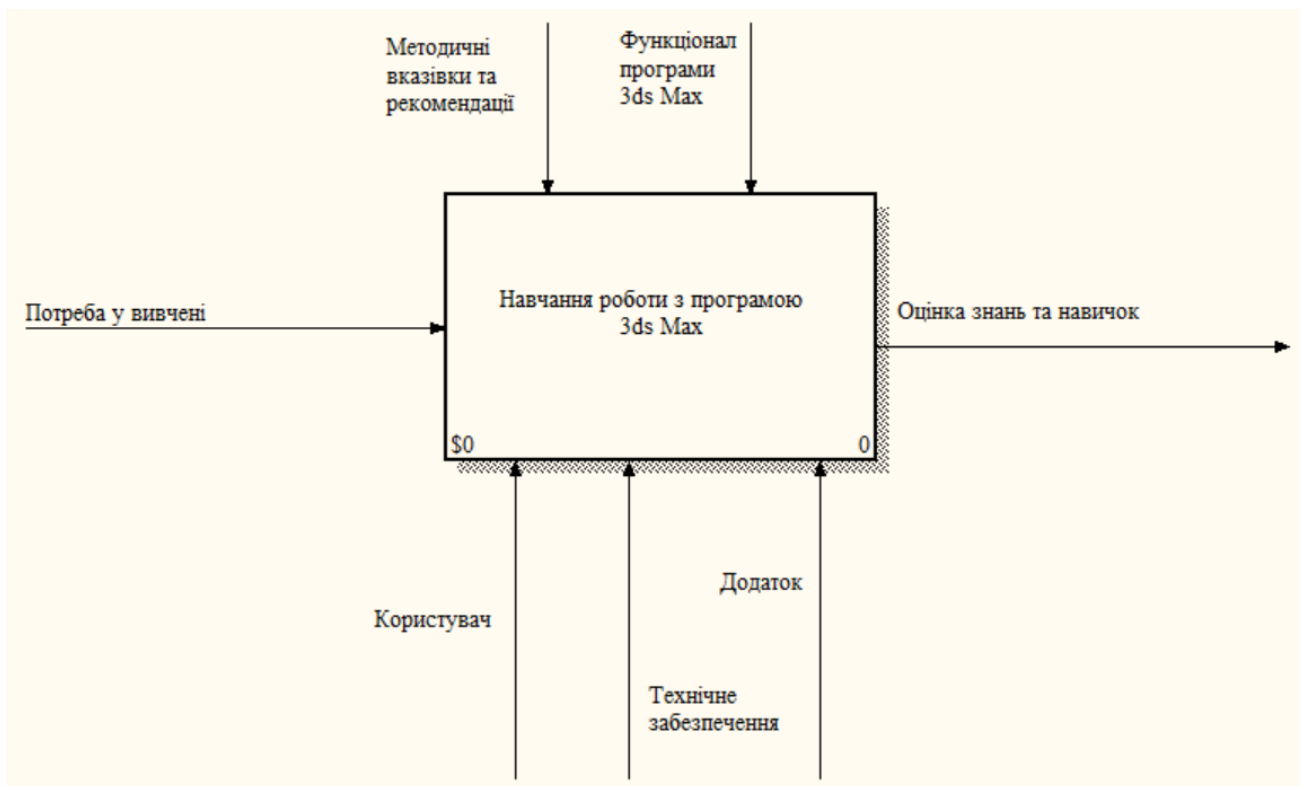


Рис. 2.1 – Контекстна діаграма

На рис. 2.2 можемо побачити виконану декомпозицію процесу функціонування. Під час декомпозиції виділено 3 функції користування додатком: “Вивчення теоретичного матеріалу”, “Вивчення практичного матеріалу”, “Тестування”.

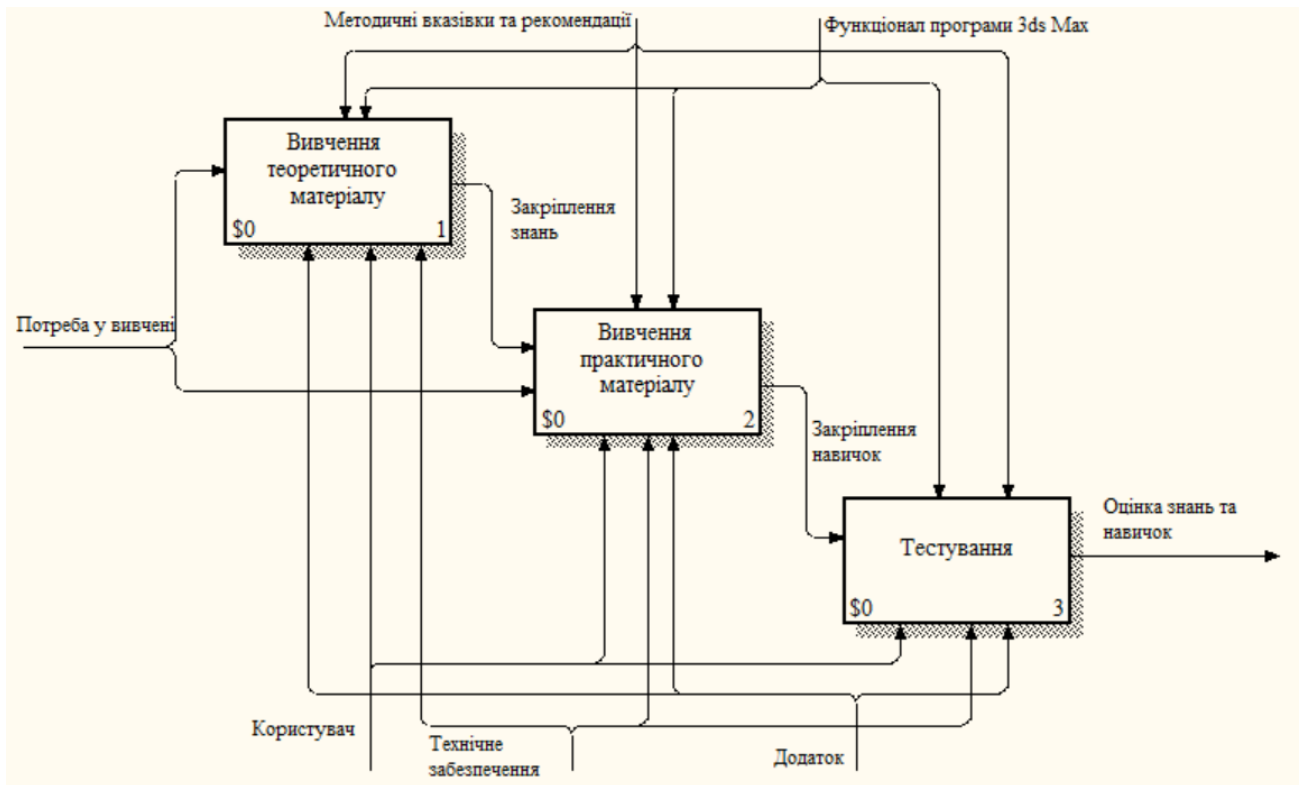


Рис. 2.2 – Декомпозиція процесу функціонування

Декомпозиція блоку “Вивчення теоретичного матеріалу” передбачає дві наступні функції: “Вибір розділу” та “Перегляд теоретичного матеріалу” (рис. 2.3).

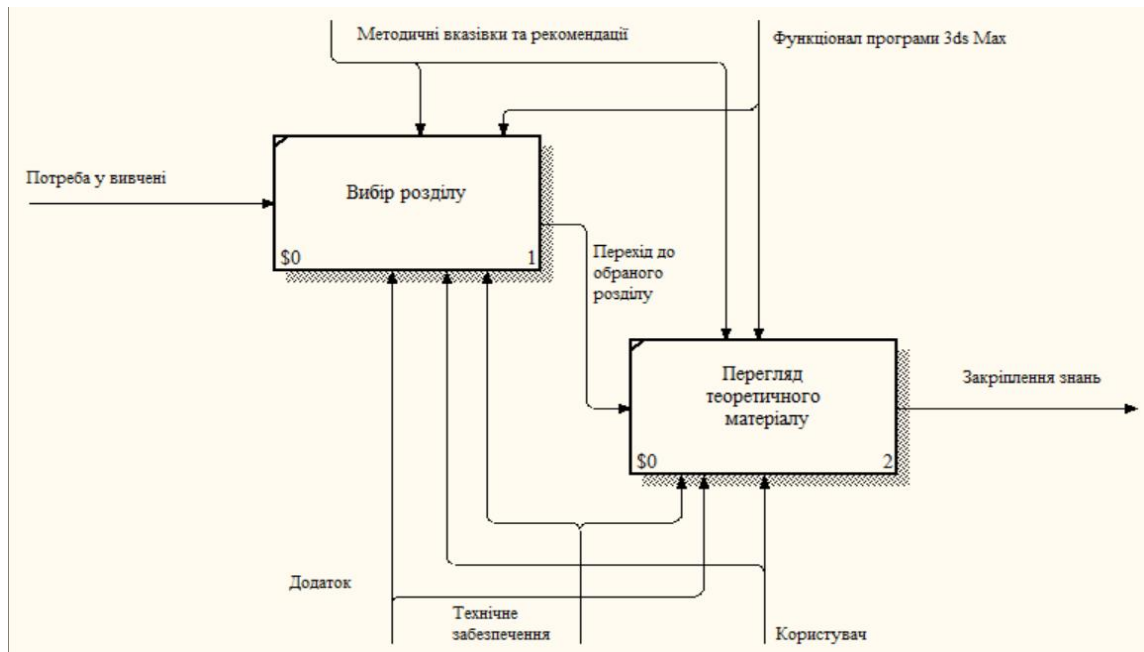


Рис. 2.3 – Декомпозиція блоку “Вивчення теоретичного матеріалу”

Аналогічним способом зробимо наступну декомпозицію блоку “Вивчення практичного матеріалу”, наведену на рисунку 2.4.

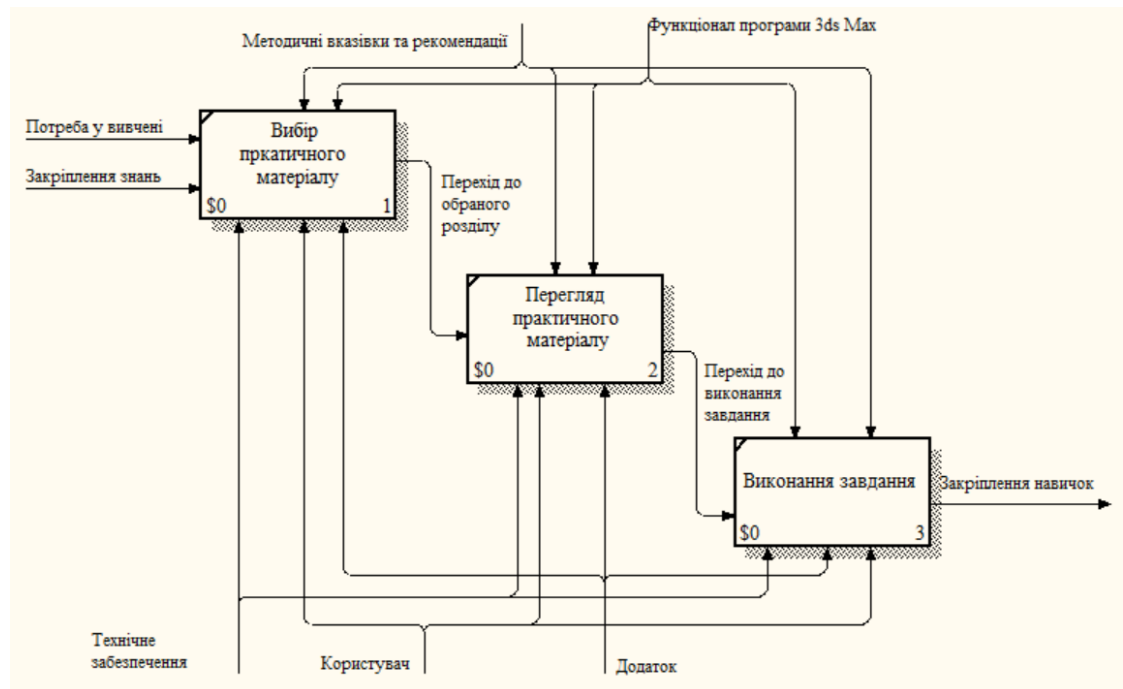


Рис. 2.4 – Декомпозиція блоку “Вивчення практичного матеріалу”

При декомпозиції блок «Тестування» розкладемо на три функції: “Вибір тесту”, “Проходження тесту” та “Отримання результату”.

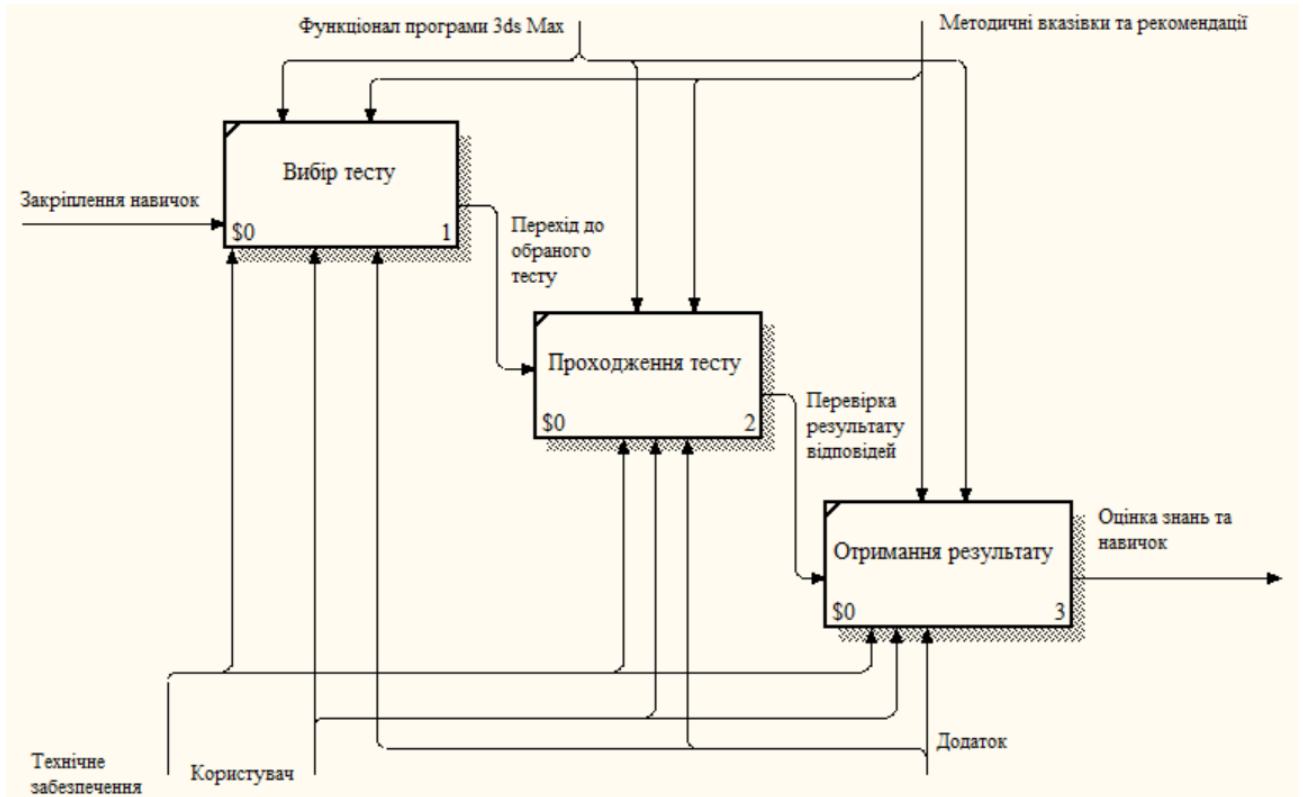


Рис. 2.5 – Декомпозиція блоку «Тестування»

2.2 Діаграма варіантів використання

Наступним етапом створимо діаграму варіантів використання. Ця діаграма призначена для відображення зв'язку взаємодії між акторами та варіантами використання. Даний зв'язок відображає, ким виконується подія.

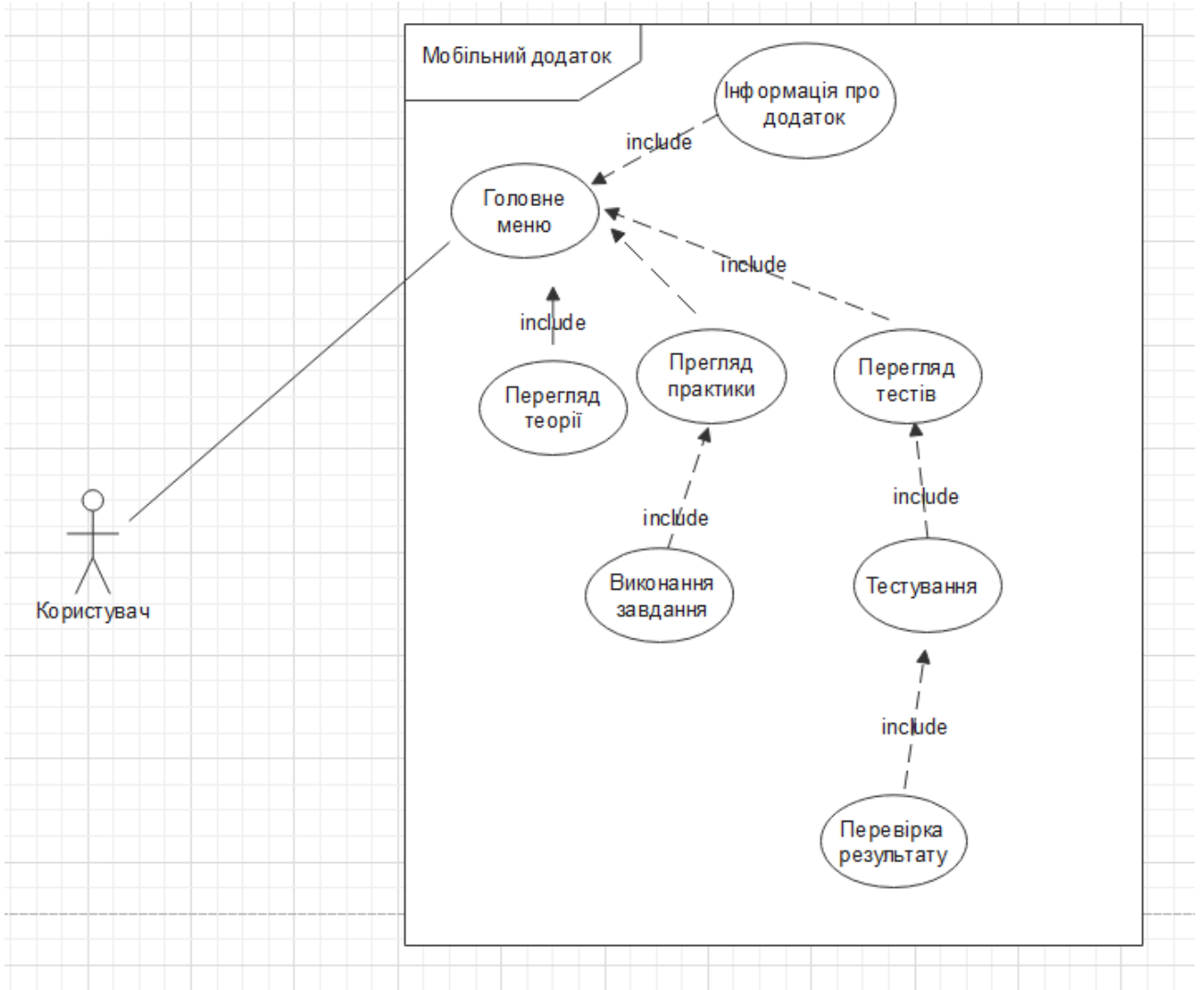


Рис. 2.6 – Діаграма варіантів використання

3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

Під час розробки мобільного додатку велику увагу було приділено реалізації чіткого й комфортного для користування функціоналу.

Щоб спростити освоєння в інтерфейсі – він був створений максимально простим для сприйняття та комфортним у користуванні.

При створенні інформаційного контенту було відібрано актуальну та найважливішу інформацію.

У додатку також передбачено практичну частину – розміщення матеріалів для виконання, де користувач має змогу завантажити .pdf файл та виконати практичне завдання.

Користувачам надається можливість проходити тестування з вибором одного правильного варіанту відповіді та переходу до наступного питання або пропуску поточного, та можливості повернутися в будь-який час, доки сторінка відкрита.

Фінальним кроком проходження тесту є відображення оцінки для користувача. За бажанням, він зможе повернутися до проходження тесту в наступний раз.

3.1 Структура додатку

Схематично структура додатку представлена на рисунку 3.1. Додаток містить три розділи – теоретичну, практичну частини і тести.

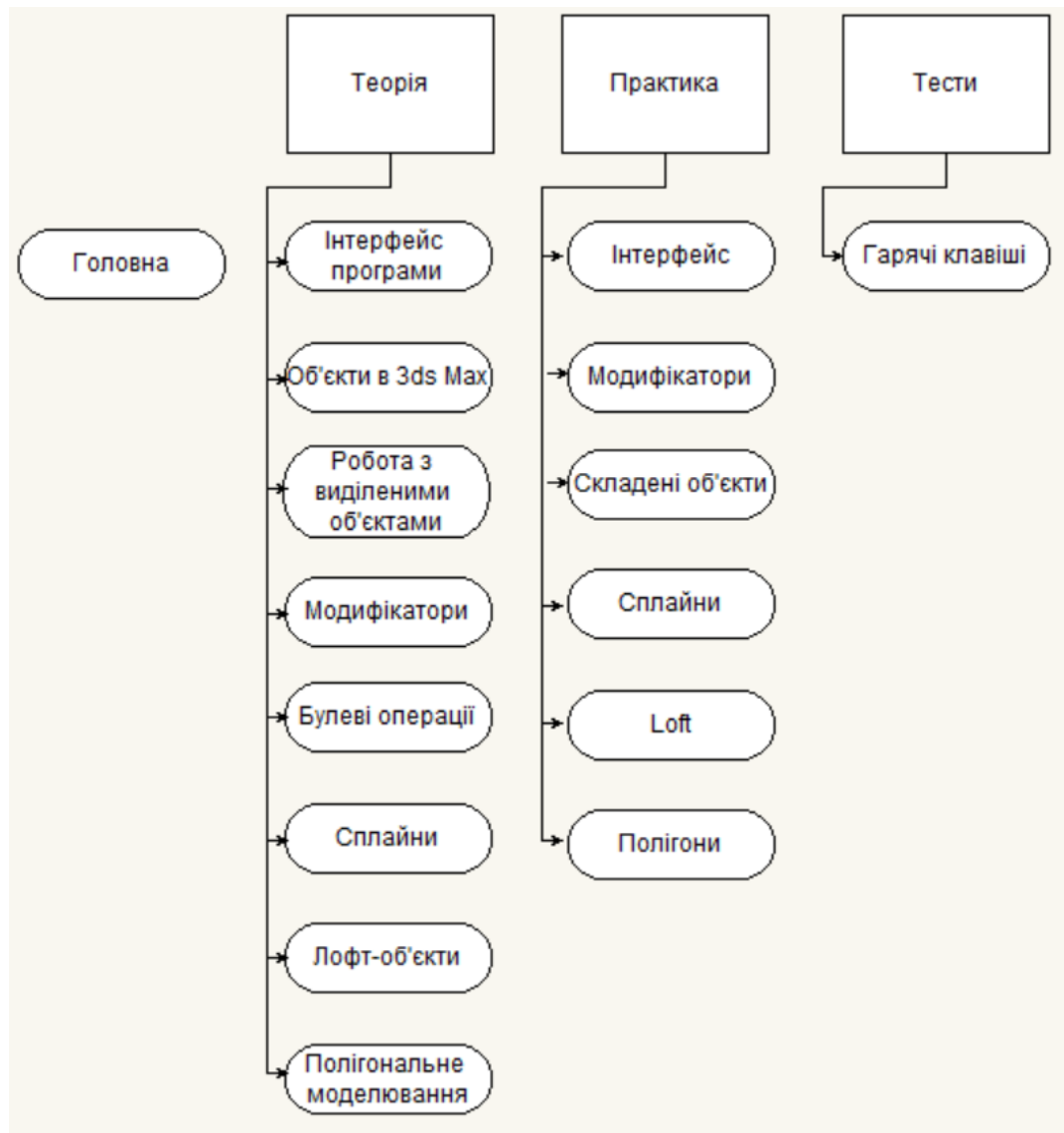


Рис. 3.1 – Схематична структура додатку

В теоретичній частині відображається найбільш важлива інформація для початку моделювання в програмі 3ds Max.

Практична частина дозволяє виконати завдання та закріпити отримані теоретичні знання.

Тест дозволяє перевірити набуті знання з використання гарячих клавіш у програмі.

3.2 Розробка інтерфейсу

Під час розроблення мобільного додатку був спроектований і сам інтерфейс сторінок. При проектуванні відібрані основні кольори, які матимуть пріоритет у користувальницькому інтерфейсі, та побудовано прототип для кожного виду сторінок (рис. 3.2).

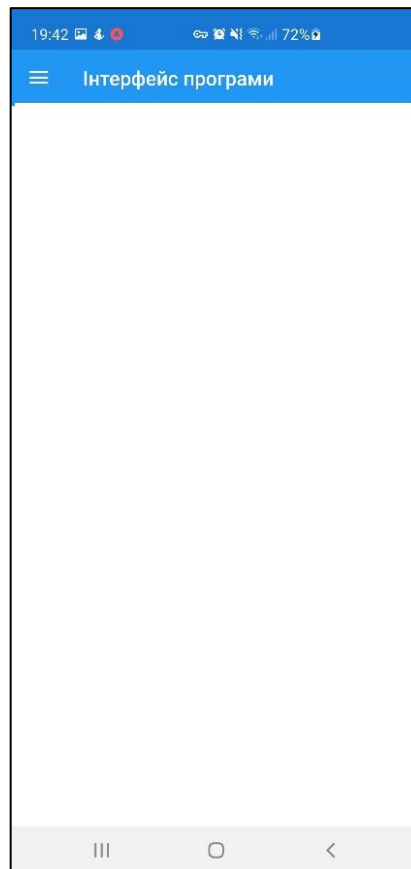


Рис. 3.2 – Прототип вигляду сторінок з інформаційним контентом

Інтерфейс мобільного додатку створений за допомогою мови розмітки XAML, та в окремих випадках, при програмній зміні, сформований написанням коду на C#.

Саме відображення відбувається за рахунок відкриття нових сторінок, що дозволяє відтворити інформацію лише при її необхідності, тобто тоді, коли користувачу потрібна та або інша інформація.

Головна сторінка відображає логотип програми 3ds Max в середині екрану та найменування додатку зверху.

На сторінках з теоретичною частиною відображаються блоки текстової частини, яка ілюструється зображеннями, що візуально доповнюють описану на сторінці інформацію.

Практична частина представляє собою сторінки з назвою практичного завдання з кнопкою для завантаження файлу знизу.

Тестовий блок відображає користувачу список з назвами тестів, які потрібно виконати (наразі представлено один тест). На сторінці тесту відображається перелік питань тесту та варіантами відповіді, які реалізовані за допомогою `RadioButton`.

3.3 Навігація в додатку

Навігація у мобільному додатку реалізована за допомогою оболонки похідного класу “`Shell`”, який відображає спливаюче вікно з лівого боку. Даний клас був використаний для поліпшення переходу між сторінками для користувача. На рисунку 3.3 приведений скріншот з програмного середовища розробки з частиною коду та візуалізацією на емуляторі.

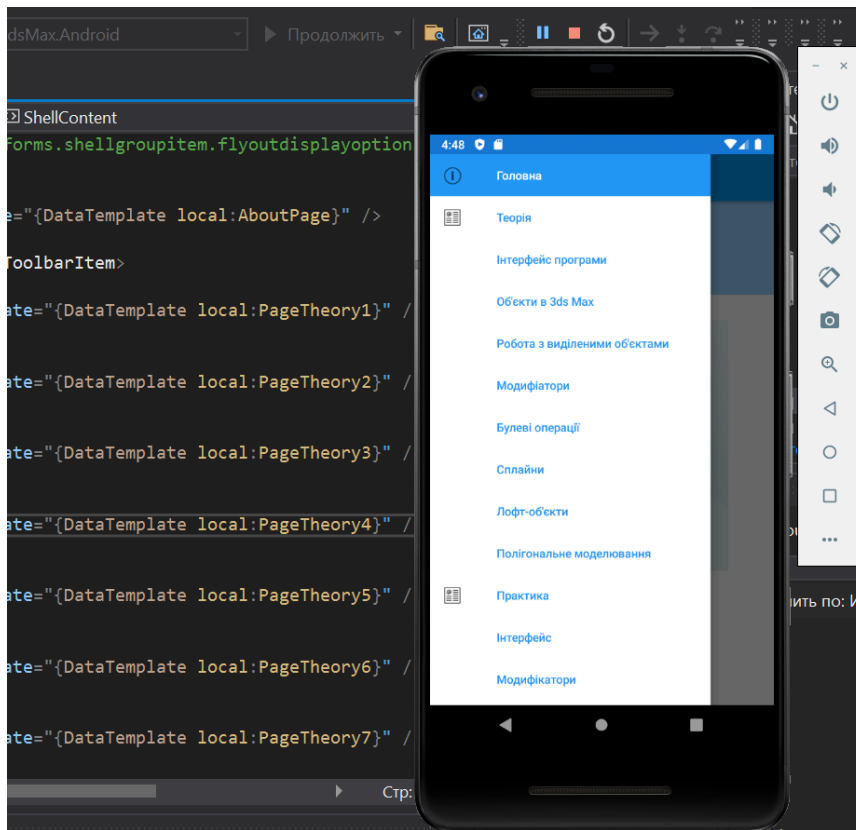


Рис. 3.3 – Навігація у додатку

Фрагмент коду програмної реалізації переходу між сторінками наведений на рисунку 3.4.

```

<FlyoutItem Title="Головна" Icon="icon_about.png">
  <ShellContent Route="AboutPage" ContentTemplate="{DataTemplate local:AboutPage}" />
</FlyoutItem>
<ToolbarItem Text="Теорія" Icon="icon_feed.png"></ToolbarItem>
<FlyoutItem Title="Інтерфейс програми">
  <ShellContent Route="PageTheory1" ContentTemplate="{DataTemplate local:PageTheory1}" />
</FlyoutItem>
<FlyoutItem Title="Об'єкти в 3ds Max">
  <ShellContent Route="PageTheory2" ContentTemplate="{DataTemplate local:PageTheory2}" />
</FlyoutItem>
<FlyoutItem Title="Робота з виділеними об'єктами">
  <ShellContent Route="PageTheory3" ContentTemplate="{DataTemplate local:PageTheory3}" />
</FlyoutItem>
<FlyoutItem Title="Модифікатори">
  <ShellContent Route="PageTheory4" ContentTemplate="{DataTemplate local:PageTheory4}" />
</FlyoutItem>
<FlyoutItem Title="Булеві операції">
  <ShellContent Route="PageTheory5" ContentTemplate="{DataTemplate local:PageTheory5}" />

```

Рис. 3.4 – Програмна реалізація переходу між сторінками

У боковому вікні розміщено назви розділів та назви сторінок з контентом. Під час натискання на потрібний користувачу розділ відбувається перехід на сторінку (рис. 3.5).

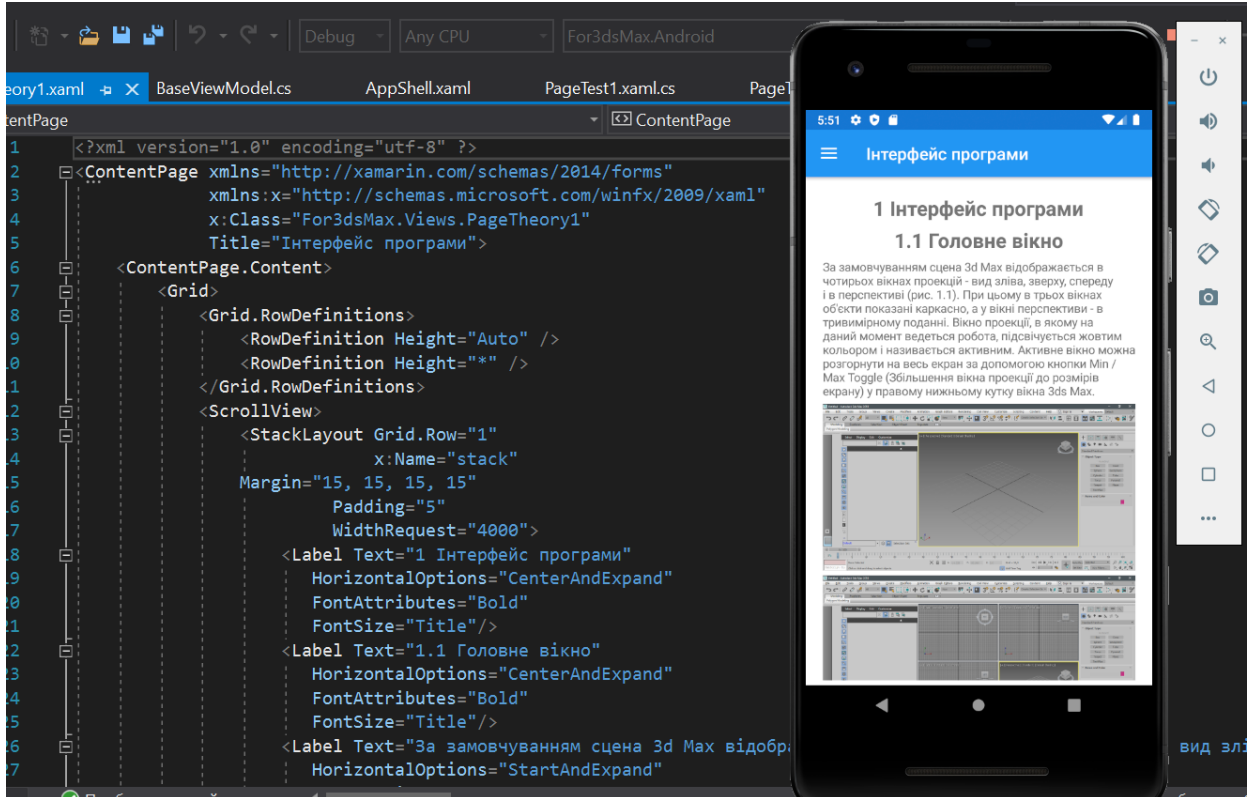


Рис. 3.5 – Перехід до сторінки “Інтерфейс програми” за допомогою бокового вікна

3.4 Кодування проекту

Програмна частина мобільного додатка створена за допомогою мови програмування C#, який пов'язаний з мовою розмітки XAML, та надає можливість відокремити програмну частину та реалізацію інтерфейсу. Даний підхід був обраний за того, що за допомогою XAML набагато простіше відтворити користувацький інтерфейс, чим за допомогою мови C#.

Першим кроком створено проект “Мобільний додаток (Xamarin.Forms)”, та обрано шаблон “Спливаючий елемент”, який у майбутньому будемо використовувати для навігації у проекті.

Далі, у маніфесті Android вказано номер версії, мінімальну версію Android, тобто 5.0 (рівень API 21) та цільову, тобто максимальну – 11.0 (рівень API 30).

Наступним кроком для створення практичної частини було розроблено елементи сторінок вмісту C# PagePracticeViewModel.cs, якому є дочірнім класом BaseViewModel.cs.

В останньому створено ім’я типу класу Comand та метод OpenAsync() для роботи з файлами, точніше для відкриття файлів за посиланням у вікні браузера. Даний клас відповідає практичному контенту та відповідає завантаженню файлів в сторінках з практичним матеріалом.

Код програмної реалізації для PagePracticeViewModel1 – PagePracticeViewModel6 представлено на рисунку 3.6.

```

9 namespace For3dsMax.ViewModels
10 {
11     public class PagePractice1ViewModel : BaseViewModel
12     {
13         public PagePractice1ViewModel()
14         {
15             Title = "About";
16             OpenWebCommand = new Command(async () => await Browser.OpenAsync("https://mix.sumdu.edu.ua/textbooks/30
17         }
18         public ICommand OpenWebCommand { get; }
19     }
20 }

```

Рис. 3.6 – Реалізація за допомогою C# класів PagePractice1 – PagePractice6

Ці класи є дочірніми до автоматично створеного середовищем розробки Microsoft Visual Studio у проектах типу Xamarin.Forms класу BaseViewModel, який був використаний для роботи з файлами. На рисунку 3.7 демонструється структура класу, назва якого була записана останньою.

```

12 public class BaseViewModel : INotifyPropertyChanged
13 {
14     public IDataStore<Item> DataStore => DependencyService.Get<IDataStore<Item>>();
15
16     bool isBusy = false;
17     public bool IsBusy{...}
22
23     string title = string.Empty;
24     public string Title{...}
29
30     protected bool SetProperty<T>(ref T backingStore, T value,
31         [CallerMemberName] string propertyName = "",
32         Action onChanged = null){...}
42
43     #region INotifyPropertyChanged
44     public event PropertyChangedEventHandler PropertyChanged;
45     protected void OnPropertyChanged([CallerMemberName] string propertyName = "")
46     {

```

Рис. 3.7 – Структура класу BaseViewModel

Далі створені ще елементи сторінок представлення XAML та C# PagePractice1 – PagePractice6, на які описано посилання для відкриття сторінок у класі AppShell, який також був створений автоматично, та представляє собою роботу з спливаючим меню.

У рядку коду 5 (рис. 3.8) підключено для звертання просторове ім'я ViewModels. В ньому знаходяться класи, використані при роботі з даною сторінкою.

У рядку коду 17 наведено звертання до змінної, яка викликає роботу з файлом та його завантаженням (рис 3.8).

```

2  <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4      x:Class="For3dsMax.Views.PagePractice1"
5      xmlns:vm1="clr-namespace:For3dsMax.ViewModels"
6      Title="Інтерфейс">
7      <ContentPage.BindingContext>
8          <vm1:PagePractice1ViewModel />
9      </ContentPage.BindingContext>
10     <ContentPage.Content>
11         <StackLayout Margin="15,15,15,15">
12             <Label Text="ОСНОВИ СТВОРЕННЯ ТА РОБОТИ З ПРОСТИМИ ОБ'ЄКТАМИ"
13                 HorizontalOptions="CenterAndExpand"
14                 FontAttributes="Italic"
15                 FontSize="Subtitle"/>
16             <Button Text="Відкрити файл"
17                 Command="{Binding OpenWebCommand}"
18                 BackgroundColor="{StaticResource Primary}"
19                 TextColor="White" />
20         </StackLayout>

```

Рис. 3.8 – Демонстрація коду PagePractice1.xaml

Реалізація теоретичних сторінок виглядає таким чином. У просторовому імені Views були створені елементи представлення сторінок XAML та C# PageTheory1 – PageTheory8. Для відтворення інформації були заповнені сторінки представлення PageTheory.xaml. На рисунку 3.9 наведено частину коду відтворення сторінок.

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4      x:Class="For3dsMax.Views.PageTheory1"
5      Title="Інтерфейс програми">
6      <ContentPage.Content>
7          <Grid>
8              <Grid.RowDefinitions>
9                  <RowDefinition Height="Auto" />
10                 <RowDefinition Height="*" />
11             </Grid.RowDefinitions>
12             <ScrollView>
13                 <StackLayout Grid.Row="1"
14                     x:Name="stack"
15                     Margin="15, 15, 15, 15"
16                     Padding="5"
17                     WidthRequest="4000">
18                     <Label Text="1 Інтерфейс програми"
19                         HorizontalOptions="CenterAndExpand"
20                         FontAttributes="Bold"

```

Рис. 3.9 – Демонстрація коду PageTheory1.xaml

Наступним кроком було створено у просторовому імені Views елементи представлення сторінок XAML та C#. PageTest1. На сторінці PageTest1.xaml описана візуальна частина сторінки, де представлені:

- блоки з текстовою інформацією,
- місце для зображення, яке за замовченням є порожнім,
- RadioButton'и, які імітують систему проходження тестів з вибором лише одного вірного результату серед однієї RadioButton.GroupName,
- Кнопка, створена для натискання, щоб отримати результат проходження тесту (рис 3. 10).

```

2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3   xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4   x:Class="For3dsMax.Views.PageTest1"
5   Title="Інтерфейс">
6   <ContentPage.Content>
7     <ScrollView>
8       <StackLayout Padding="15,15,15,15" BackgroundColor="FloralWhite">
9         <Label Padding="0,15,0,0" Text="1. Виберіть вірну відповідь" FontSize="Body" FontAttributes="Bold" FontFam
10        <Label Padding="0,0,0,0" x:Name="lb1" Text="In this label write some text..." FontSize="Body" FontAttribut
11        <ImageButton BackgroundColor="FloralWhite" VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img1"
12        <Label Padding="0,0,0,0"/>
13        <RadioButton x:Name="rb11" GroupName="RBGroup1" Content="" ></RadioButton>
14        <RadioButton x:Name="rb12" GroupName="RBGroup1" Content="" ></RadioButton>
15        <RadioButton x:Name="rb13" GroupName="RBGroup1" Content="" ></RadioButton>
16        <RadioButton x:Name="rb14" GroupName="RBGroup1" Content="" ></RadioButton>
90        <Label Padding="0,15,0,0" Text="10. Виберіть вірну відповідь" FontSize="Body" FontAttributes="Bold" Fo
91        <Label Padding="0,0,0,0" x:Name="lb10" Text="In this label write some text..." FontSize="Body" FontAtt
92        <ImageButton BackgroundColor="FloralWhite" VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="In
93        <Label Padding="0,0,0,0"/>
94        <RadioButton x:Name="rb101" GroupName="RBGroup10" Content="" ></RadioButton>
95        <RadioButton x:Name="rb102" GroupName="RBGroup10" Content="" ></RadioButton>
96        <RadioButton x:Name="rb103" GroupName="RBGroup10" Content="" ></RadioButton>
97        <RadioButton x:Name="rb104" GroupName="RBGroup10" Content="" ></RadioButton>
98
99        <Button x:Name="btn1" Text="Check Result"></Button>
100      </StackLayout>
101    </ScrollView>
102  </ContentPage.Content>

```

Рис. 3.10 – Демонстрація коду PageTest11.xaml

Програмна реалізація проходження тесту представлена у PageTest1.xaml.cs.

Для відтворення випадкового розкиду питань було створено два методи RandomTasks() та RandomTexts. Перший створює запис у масив індивідуального номеру розмір з кількістю питань у тестах, а другий створює випадковий порядок

відповіді на питання у `RadioButton.Content`, тобто у текстовий підпис варіанту вибору.

Наразі вся інформація, що стосується тестових питань, знаходиться у двомірному масиві, де під другим (елементом масиву під номером 1) номером стовбця знаходиться вірна відповідь. Цей спосіб був вибраний для можливості перевірки вірної відповіді з поєднанням випадкового представлення у `RadioButton`.

Для можливості маніпулювання з кнопкою та варіантами відповідей, були створені події для кожного з них, де у подіях натискання варіантів відповіді перевіряється вірність відповіді під час зміни обраного `RadioButton`.

Сам підрахунок кількості правильних відповідей відбувається у події натискання на кнопку “Button” (рис. 3.11).

```

30 private void CreateEvents()...
74
75 private void CheckedIsTorF(Boolean TorF)...
127
128 private void InitializationText(Label label, string[,] Array,int count1, int[] count2, int i)..
129 {
130     label.Text = Convert.ToString(Array[count1, count2[i]]);
131 }
132 private void InitializationSource(ImageButton imagebutton, string[,] Array, int count1, int[] count2, int i)
133 {
134     imagebutton.Source = Convert.ToString(Array[count1, count2[i]]);
135 }
136
137 private void RandomTasks()...
187
188 private void RandomTexts(RadioButton rb1, RadioButton rb2, RadioButton rb3, RadioButton rb4, int Count, string[,]
228

```

Рис. 3.11 – Структура класу `PageTest1.xaml.cs`

3.5 Тестування додатку

Після розроблення мобільного додатку потрібно провести тестування всіх можливостей взаємодій користувача з додатком. Тестування будемо проводити на двох телефонах: Samsung Galaxy S10+ та Samsung Galaxy J7.

Результати тестування наведено в таблиці 3.1.

Таблиця 3.1 – Тестування мобільного додатку

ТС_ID	Сценарій тесту	Акт	Подія	Статус
TСI.1	Перевірка проходження тесту	1. Вибір тесту 2. Обирання відповіді з переліку доступних 3. Всі питання повинні бути чітко видними 4. Вибір відповіді у завданні повинен бути лише один	Під час обрання вірних відповідей, їх кількість буде підрахована. Користувачу буде виведена інформація з кількістю правильних відповідей у відсотках	+
TСI.2	Перевірка результату тесту	1. Натиснення на кнопку для завершення проходження тесту	Відкривається вікно з назвою тесту та правильною кількістю відповідей у відсотках	+
TСI.3	Вибір сторінки	1. Натиснути на кнопку у верхній частині лівого екрану 2. Провести пальцем з лівої частини екрану до центра 3. Обрати потрібну сторінку з переліку	Після обрання сторінки, вона повинна відкритися на екрані мобільного пристрою користувача	+
TСI.4	Завантаження файлу з практичною частиною	1. Натиснути на кнопку у верхній частині лівого екрану 2. Провести пальцем з лівої частини екрану до центра 3. Обрати потрібну сторінку з практичним матеріалом у переліку 4. Натиснути на кнопку для відкриття її у браузері 5. Завантажити на мобільний пристрій	Після відкриття практичного матеріалу, користувач матиме змогу перейти у браузер та завантажити файл на свій мобільний пристрій	+
TСI.5	Відкриття всіх розділів	1. Натиснути на кнопку у верхній частині лівого екрану 2. Провести пальцем з лівої частини екрану до центра	Відкривається спливаюче меню зі списком всіх доступних сторінок для користувача	+

Проведене тестування показало, що додаток працює коректно, помилки відсутні.

3.6 Демонстрація роботи

При відкритті додатку користувачу відображається головне меню із зображенням логотипу програми 3ds Max в середині екрану (рис 3.12).

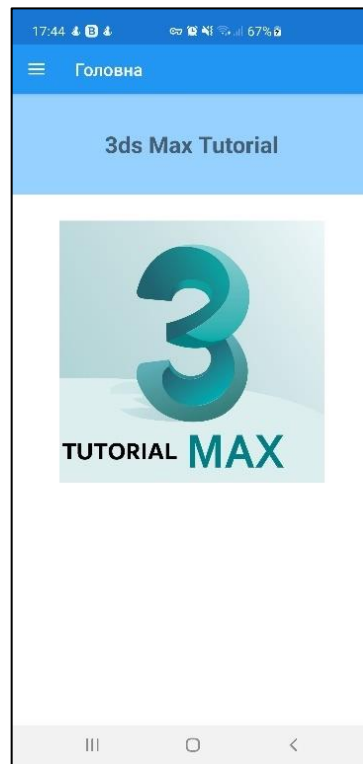


Рис. 3.12 – Демонстрація вигляду головної сторінки

Для того, щоб перейти до сторінок з інформативним контентом мобільного додатку, потрібно протягнути пальцем з лівого боку екрану до центра, таким чином користувач може побачити спливаюче вікно із переліком наявних розділів (рис 3.13).

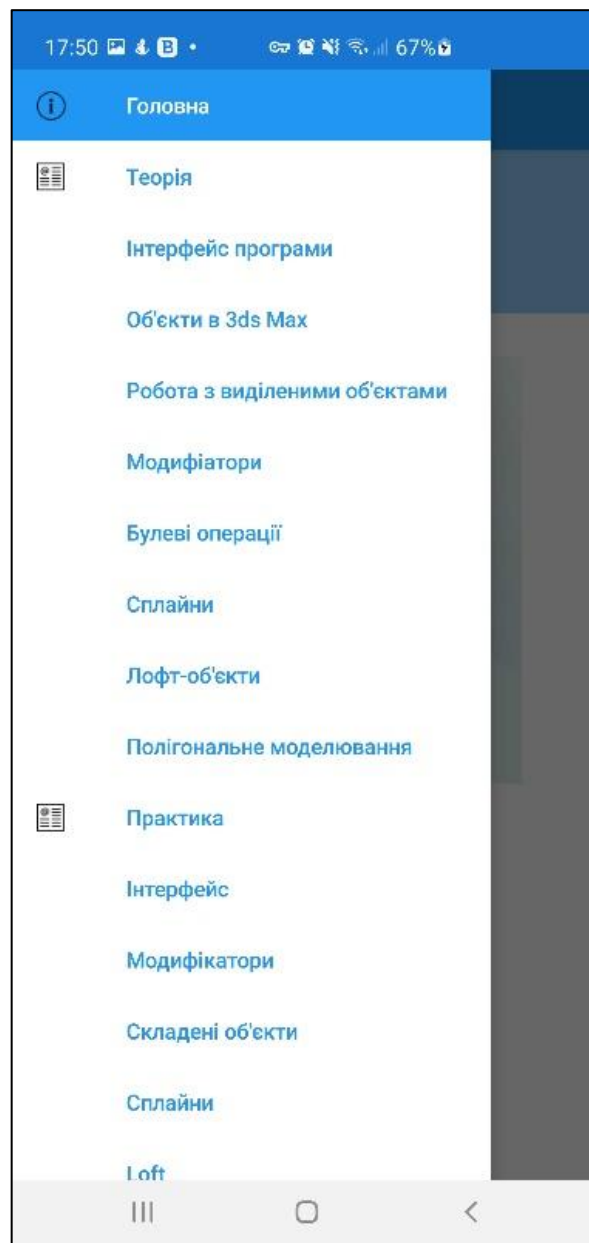


Рис. 3.13 – Демонстрація вигляду спливаючого меню

Наступним кроком потрібно обрати необхідну сторінку для перегляду її змісту. Наприклад, виберемо з першого розділу “Теорія” – сторінку “Модифікатори” (рис. 3.14).

Для перегляду інформації на кожній із сторінок, потрібно прогортати її до низу (рис. 3.14).

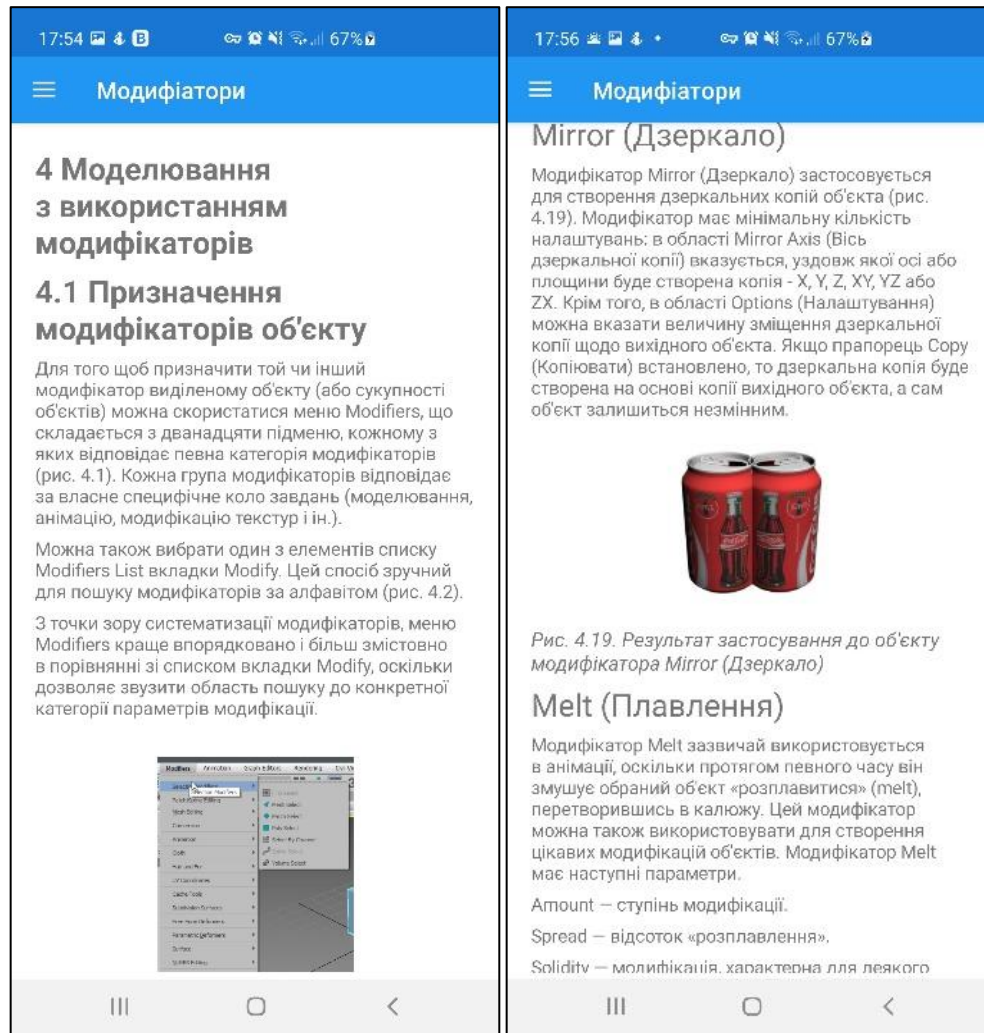


Рис. 3.14 – Демонстрація сторінки з теоретичним контентом “Модифікатори”

Коли користувач захоче змінити сторінку на іншу, йому потрібно таким же чином, як і раніше, протягнути пальцем з лівого боку екрану до центра або натиснути на позначку меню у верхній частині лівого екрану, що зображена як “≡”.

При відкритті спливаючого меню можемо побачити розділ, на якому користувач зараз знаходиться – він відрізняється заднім фоном тексту синього кольору (рис. 3.15).

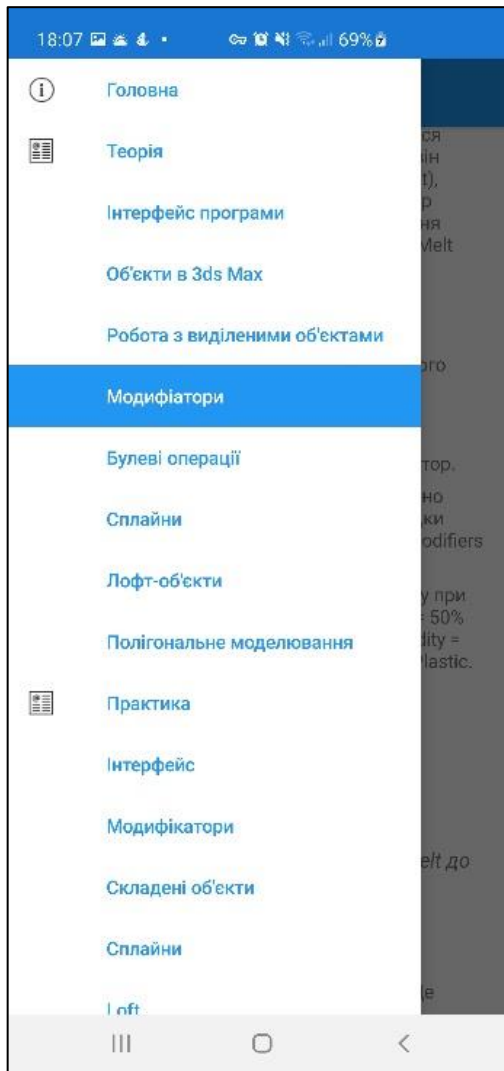


Рис. 3.15 – Демонстрація вигляду спливаючого меню з відкритою сторінкою “Модифікатори” з теоретичного розділу

Виберемо, наприклад, в наступному розділі “Практика” – сторінку “Сплайни”. При відкриті сторінки з розділу “Практика” користувач має змогу побачити повну назву практичного матеріалу та кнопку для завантаження файлу – “Відкрити файл” (рис. 3.16).

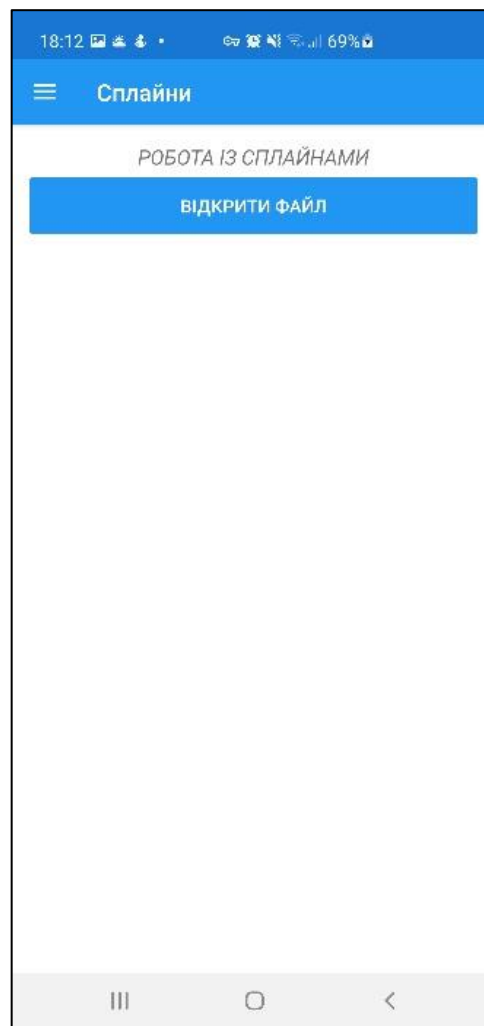


Рис. 3.16 – Демонстрація вигляду сторінки з практичним матеріалом “Слайни”

Щоб завантажити файл, потрібно натиснути на кнопку, після чого цей файл буде завантажений у браузері. За необхідності, можливо потрібно буде підтвердити завантаження у самому браузері. Далі обраний файл завантажиться на мобільному пристрої (рис. 3.17).

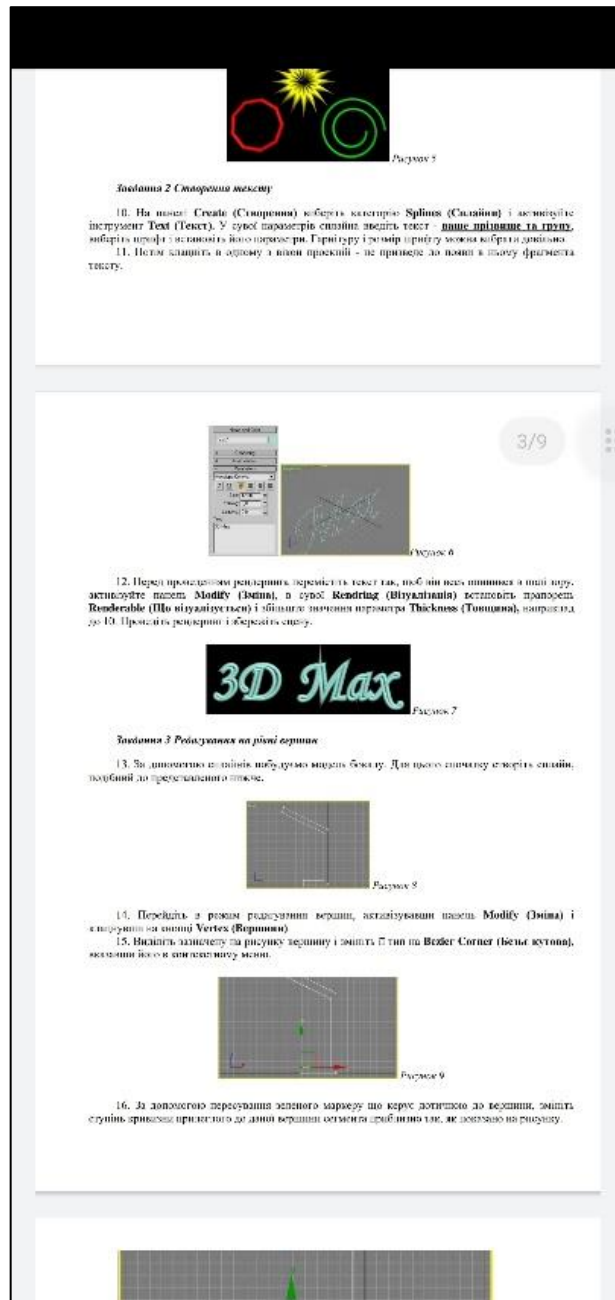


Рис. 3.17 – Завантажений файл за допомогою кнопки “Відкрити файл” на мобільному пристрої

Тепер перейдемо в наступний розділ “Тести” та виберемо тест “Гарячі клавіші”. Користувачу демонструється сторінка з рядом тестових питань, на котрі користувач може відповісти однією лише відповіддю з чотирьох (рис. 3.18).

Після проходження тестової сторінки у її кінці користувач може натиснути на кнопку “Перевірити результат”, тоді користувачу відобразиться спливаюче вікно з отриманим результатом (рис. 3.19).

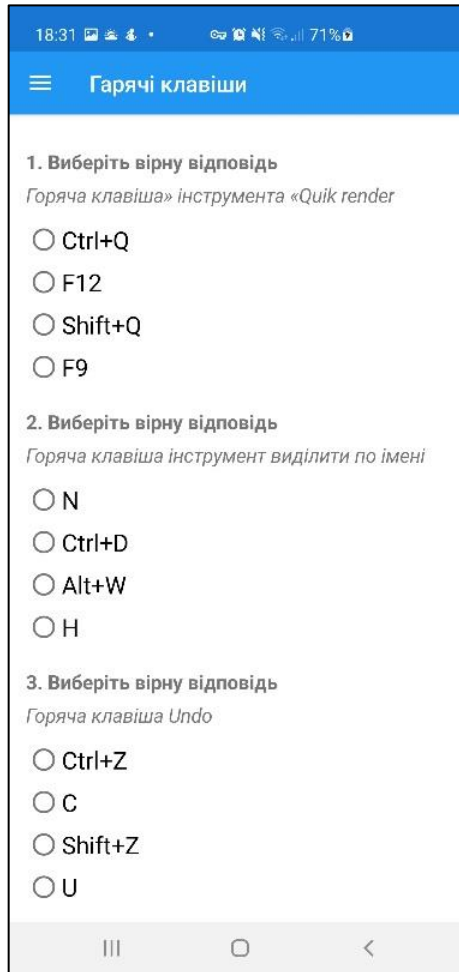


Рис. 3.18 – Демонстрація сторінки з тестовим матеріалом “Гарячі клавіші”

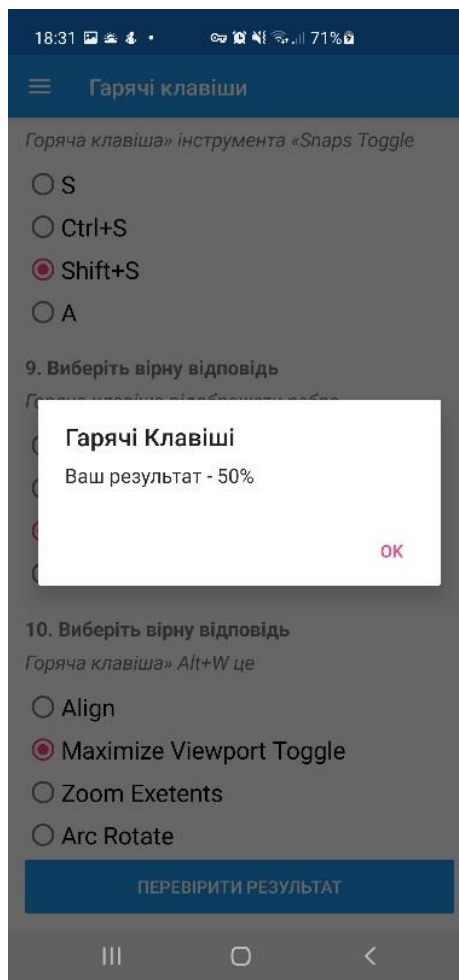


Рис. 3.19 – Демонстрація події при натисканні кнопки “Перевірити результат”

Таким чином, можна зробити висновок, що поставлена мета роботи досягнута – розроблено мобільний додаток, в якому реалізовано необхідний функціонал.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи був проведений аналіз предметної області, виокремлено категорії та вимоги до мобільних додатків для навчання. Також проведений огляд продуктів-аналогів, були оцінені всі їх недоліки та переваги.

В результаті сформульовано постановку задачі, виділено 4 основні задачі для досягнення цілі.

Сформульовано технічне завдання на виконання роботи та проведено планування робіт, аналіз ризиків, розроблено календарний план виконання робіт.

Проведено структурно-функціональний аналіз проєкту, побудовано контекстну діаграму IDEF0 та проведено її декомпозицію.

Розроблено структуру додатку, інтерфейс головної сторінки, сторінок теоретичного та практичного матеріалів, тестів.

Було розроблено функціонал перегляду теоретичних сторінок з великим обсягом інформативного контенту, можливість завантаження файлів з браузера для проходження практичної частини, тестову частину, де у користувача є можливість перевірити свої знання за допомогою проходження тестів.

В результаті розроблено програмний продукт – мобільний додаток, який є допоміжним засобом при вивченні програми 3ds Max.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мобильное приложение для образования [Электронный ресурс] – URL: <https://wezom.com.ua/blog/prilozheniya-dlya-obrazovaniya>.
2. Мобильное обучение. Обучение на мобильных устройствах: прошлое, настоящее и будущее [Электронный ресурс] – URL: <http://appttractor.ru/mLearning/>
3. Рекомендации ЮНЕСКО по политике в области мобильного обучения [Электронный ресурс] – URL: <http://iite.unesco.org/pics/publications/ru/files/3214738.pdf>
4. Белоус В. Мобильные учебные приложения в современном образовании. // Освітнологічний дискурс, 2018, № 1–2 (20–21). – С. 353–362. ISSN Online: 2312–5829.
5. Мобільний додаток TED. Архив полезных и вдохновляющих лекций. [Электронный ресурс] – URL: <https://www.wonderzine.com/wonderzine/life/life/237303-ted>.
6. Codecademy – приложение, которое поможет вам начать программировать [Электронный ресурс] – URL: <https://lifelhacker.ru/codecademy/>
7. Visual Anatomy Free. [Электронный ресурс] – URL: <https://www.edumobapp.com/product.html>.
8. Essential Skeleton 4. [Электронный ресурс] – URL: <https://apps.apple.com/us/app/essential-skeleton-4/id623811668>
9. 3ds Max Tutorials. [Электронный ресурс] – URL: <https://play.google.com/store/apps/details?id=admin.a3dsmaxtutorial&hl=ru&gl=US>
10. What is Xamarin? [Электронный ресурс] – URL: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>

11. C# documentation [Электронный ресурс] – URL: <https://docs.microsoft.com/en-us/dotnet/csharp/>
12. XAML overview (WPF .NET) [Электронный ресурс] – URL: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-5.0>.
13. Visual Anatomy. App review by Debbie Gorrell, Common Sense Media [Электронный ресурс] – URL: <https://www.commonsensemedia.org/app-reviews/visual-anatomy/>
14. РД IDEF 0 – 2000: Методология функционального моделирования IDEF0 Разработан Центром CALS-технологий «Прикладная Логистика». [Электронный ресурс] – URL: <https://nsu.ru/smk/files/idef.pdf>.
15. Марка Д., МакГоуен К. Методология структурного анализа и проектирования SADT. – Пер. с англ. – М.: Метатехнология, 1993. – 243 с.
16. A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition. – Project Management Institute. – 2018, 726 с.
17. Paul Hersey, Kenneth H. Blanchard, Dewey E. John. Management of Organizational Behavior: Utilizing Human Resources. – Prentice Hall, 1996, 360с.
18. Morrison M. Why SMART Objectives don't work [Электронный ресурс] – URL: <https://rapidbi.com/why-smart-objectives-dont-work/>
19. Morrison M. History of SMART Objectives [Электронный ресурс] – URL: <https://rapidbi.com/history-of-smart-objectives/>
20. Винеvская А. В. Использование потенциала информационных технологий в создании мобильной образовательной среды. //Концепт. – 2012. №9. [Электронный ресурс] – URL: <https://cyberleninka.ru/article/n/ispolzovanie-potentsiala-informatsionnyh-tehnologiy-v-sozdanii-mobilnoy-obrazovatelnoy-sredy/viewer>.
21. Голицына И. Н., Половникова Н. Л. Мобильное обучение как новая технология в образовании. [Электронный ресурс] – URL:

<https://cyberleninka.ru/article/n/mobilnoe-obuchenie-kak-novaya-tehnologiya-v-obrazovanii/viewer>.

22. Beerbohm M., Mohmmmed M. Xamarin: Xamarin for beginners , Building Your First Mobile App with C# .NET and Xamarin – 3rd Edition. Independently Published, 2020. 116 p.
23. Snider E. Mastering Xamarin.Forms: App architecture techniques for building multi-platform, native mobile apps with Xamarin.Forms 4. – 3rd Edition. – Pact Publishing, 2019. – 185 p.
24. Bilgin C. Hands-On Mobile Development with .NET Core: Build cross-platform mobile applications with Xamarin, Visual Studio 2019, and .NET Core 3. – Pact Publishing, 2019. – 504 p.

ДОДАТОК А
ТЕХНІЧНЕ ЗАВДАННЯ

на розробку інформаційної системи
«Мобільний додаток для вивчення 3ds Max»

ПОГОДЖЕНО:

Доцент кафедри комп'ютерних наук

_____ Баранова І.В.

Студент групи ІТ-72

_____ Пархоменко О.В.

1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ

1.1 Призначення мобільного додатку

Додаток повинен надати користувачу інформацію, яка має допомогти йому почати працювати з програмою “3ds Max”, та закріпити навички роботи в ній.

1.2 Мета створення мобільного додатку

Полегшити процес вивчення матеріалу, призначеного для роботи з програмою “3ds Max” у одному мобільному додатку, для зручності поглиблення у цій сфері.

1.3 Цільова аудиторія

Цільовою аудиторією мобільного додатку являються люди, які просуваються у сфері візуалізації. Так як у додатку велика сукупність інформації на різні теми та з різною складністю 3d моделювання у програмі “3ds Max”, даний додаток підійде як користувачам початкового рівня, так і поглибленого.

2 ВИМОГО ДО МОБІЛЬНОГО ДОДАТКУ

2.1 Вимоги до мобільного додатку в цілому

2.1.1 Вимоги до структури й функціонування мобільного додатку

Мобільний додаток має бути доступним в магазині додатків для вільного скачування. Кожен розділ повинен бути індивідуальним і містити чітку інформацію про відповідну тему. Вивчення кожного розділу закріплюється проходженням тесту, для самоперевірки. Для зручності навігації великими за обсягом темами, позиція, на якій зупинився користувач, буде зберігатися.

2.1.2 Вимоги до збереження інформації

Уся інформація надана у мобільному додатку буде зберігатися у XAML коді на сторінці, яка була створена під конкретну тему.

2.1.3 Вимоги до розмежування доступу

Створений мобільний додаток повинен знаходитися у вільному доступі для

завантаження.

У додатку передбачений лише один рівень доступу.

Кожен користувач має можливість переглянути всю інформацію, яка є у мобільному додатку.

2.2 Структура мобільного додатку

2.2.1 Загальна інформація про структуру мобільного додатку

Структура мобільного додатку являє набір сторінок, які відображені у “Переліку”, і має наступний вигляд:

Головна – на сторінці зображені логотип додатку і список, у якому описуються теми, які є посиланням на наступні сторінки.

Про додаток – на даній сторінці відображена основна інформація про мобільний додаток.

Розділ – при переході на будь-який розділ, на сторінці буде відображена інформація стосовно конкретного розділу.

2.2.2 Навігація

Навігація по сторінкам додатку реалізована наступним чином. Для переходу на конкретний розділ потрібно обрати відповідний пункт у спеціально відведеній частині “Перелік” на головній сторінці. Щоб перейти на іншу сторінку, потрібно натиснути на кнопку “Назад” або “Вперед”. Для повернення на головну сторінку, створена кнопка “Головна”.

2.2.3 Наповнення мобільного додатку (контент)

Для створення контенту, будемо додавати нові сторінки у Xamarin.Forms.

Редагування та додавання контенту також буде проводитися у Xamarin.Forms.

Весь контент, який стосується інформації, як користуватися 3ds Max, повинен бути взятий з документації від розробника, відповідної літератури, власних навиків.

2.2.4 Дизайн та структура додатку

Стиль мобільного додатку повинен бути приємним візуально, та не викликати складнощів під час використання.

Основна інформація повинна бути чітко описана у текстовому вигляді, та у деяких моментах підкріплюватися відповідним зображенням з програми 3ds Max.

Головна сторінка має наступне схематичне зображення:

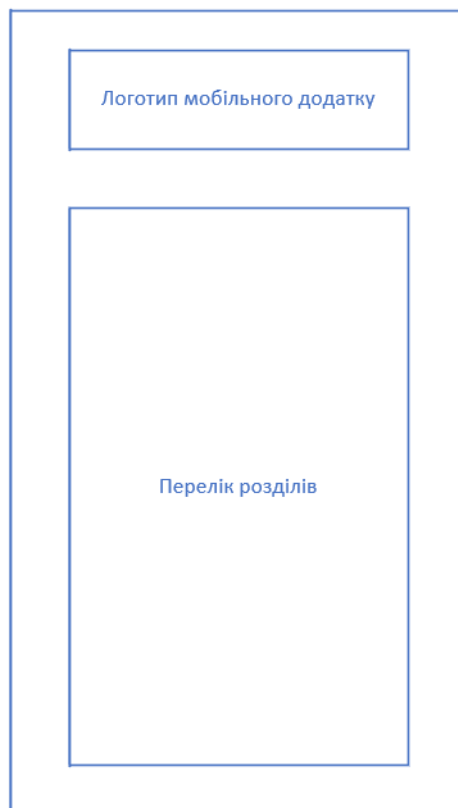


Рисунок 1 – Схема головної сторінки



Рисунок 2 – Схема сторінок розділів

2.2.5 Система навігації (карта мобільного додатку)

Карта мобільного додатку зображена на рисунку 3.



Рисунок 3 – Карта мобільного додатку

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці 1.

Таблиця 1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Можливість швидкого переходу між темами	Клієнт
UN-02	Перегляд інформації у вільному доступі	Клієнт
UN-03	Детальний опис кожного розділу	Клієнт
UN-04	Підкріплення інформації за допомогою скріншотів з програми 3ds Max	Клієнт
UN-05	Великий обсяг інформації	Клієнт
UN-06	Різний обсяг інформації за поглибленістю та складністю	Клієнт

2.3.2 Функціональні вимоги

На основі потреб користувача були визначені такі функціональні вимоги:

- Детальний опис кожного розділу та підкріплення скріншотами з програми 3ds Max.
- Можливість швидкого переходу між темами.
- Доповнення інформації новим контентом.
- Створення контенту різного виду складності та поглибленості.

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці 2.

Умовні позначення в таблиці 2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

Таблиця 2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Створення переліку розділів	M	Надає можливість переходити між розділами
SR-02	Створення дій для швидкого переходу між розділами	S	Надає можливість швидкого переходу між розділами
SR-03	Модуль зворотного зв'язку	C	Надає можливість зворотного зв'язку
SR-04	Сортування розділів за типом	M	Спрощує навігацію у додатку

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до лінгвістичного забезпечення

Мобільний додаток має бути виконаний українською мовою.

3 СКЛАД І ЗМІСТ РОБІТ ІЗ СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ

Докладний опис етапів роботи зі створення мобільного додатку наведено в таблиці 3.

Таблиця 3 – Етапи створення мобільного додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Постановка цілей необхідних для досягнення певного результату	4 день
2	Складання технічного завдання	3 дні
3	Підготовка прототипу	3 дні
4	Створення макету дизайну мобільного додатку	3 дні
5	Створення контенту для розділів	5 днів
6	Підготовка тестів для користувачів	3 дні
7	Створення візуальної частини	5 днів
8	Створення додатку	10 днів
9	Створення розділів	7 днів
10	Додавання контенту	5 днів
11	Перевірка працездатності	1 день
12	Налагодження додатку	2 дні
	Загальна тривалість робіт	48 днів

4 ВИМОГИ ДО СКЛАДУ Й ЗМІСТУ РОБІТ ІЗ ВВЕДЕННЯ МОБІЛЬНОГО ДОДАТКУ В ЕКСПЛУАТАЦІЮ

Для того, щоб мобільним додатком могли користуватися, додамо його до магазину додатків по типу Google Play. Щоб уникнути складності під час доробки та додавання нового матеріалу, потрібно дотримуватися вимоги, які були зазначені у ТЗ.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

Деталізація мети проекту методом SMART. Продуктом дипломного проекту є мобільний додаток "Довідник для 3ds Max".

Результати деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити мобільний додаток "Довідник для 3ds Max", який буде допоміжним інструментом для вивчення програми 3ds Max.
Measurable (вимірювана)	Результатом роботи проекту є оцінка замовника.
Achievable (досяжна)	Реалізація здійснюється за допомогою платформи Xamarin у програмному середовищі Visual Studio
Relevant (реалістична)	У наявності є всі необхідні технічні та програмні засоби. Розробники достатньо кваліфіковані для виконання поставлених задач.
Time-framed (обмежена у часі)	Ціль має часове обмеження. Робота повинна бути виконана у терміни, що були оговорені замовником проекту. Проект повинен бути виконаний згідно з календарним планом.

Планування змісту структури робіт. Основним інструментом для планування змісту структури робіт служить WBS діаграма – графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов'язані з продуктом проекту. Побудуємо структуру WBS, у якій детально опишемо

роботи, які потрібно виконати на кожному етапі створення проекту. Виконаємо декомпозицію робіт для даного проекту. Діаграма WBS зображена на рис. Б.1.

Планування структури організації, для впровадження готового проекту (OBS). Після побудови WBS розробимо організаційну структуру виконавців OBS. Організаційна структура проекту стосується тільки внутрішньої організаційної структури проекту і не стосується відносин проектних груп чи учасників з батьківськими організаціями. Діаграма OBS зображена на рис. Б.2. Список виконавців, що функціонують в проекті знаходиться в табл. Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Пархоменко О.В.	Виконує розробку основного функціоналу проекту та інтерфейс користувача
Проектувальник	Пархоменко О.В.	Проектує 3D моделі та елементи тривимірної графіки, розробляє дизайн програми.
Тестувальник	Пархоменко О.В.	Відповідає за тестування функціоналу та дизайну додатку, перевірку моделі на адекватність.
Косультант проекту	Баранова І.В.	Формує завдання на розробку проекту.
Менеджер проекту	Баранова І.В.	Відповідає за виконання термінів, розподіл ресурсів та завдань між учасниками. Виконує збір та аналіз даних.

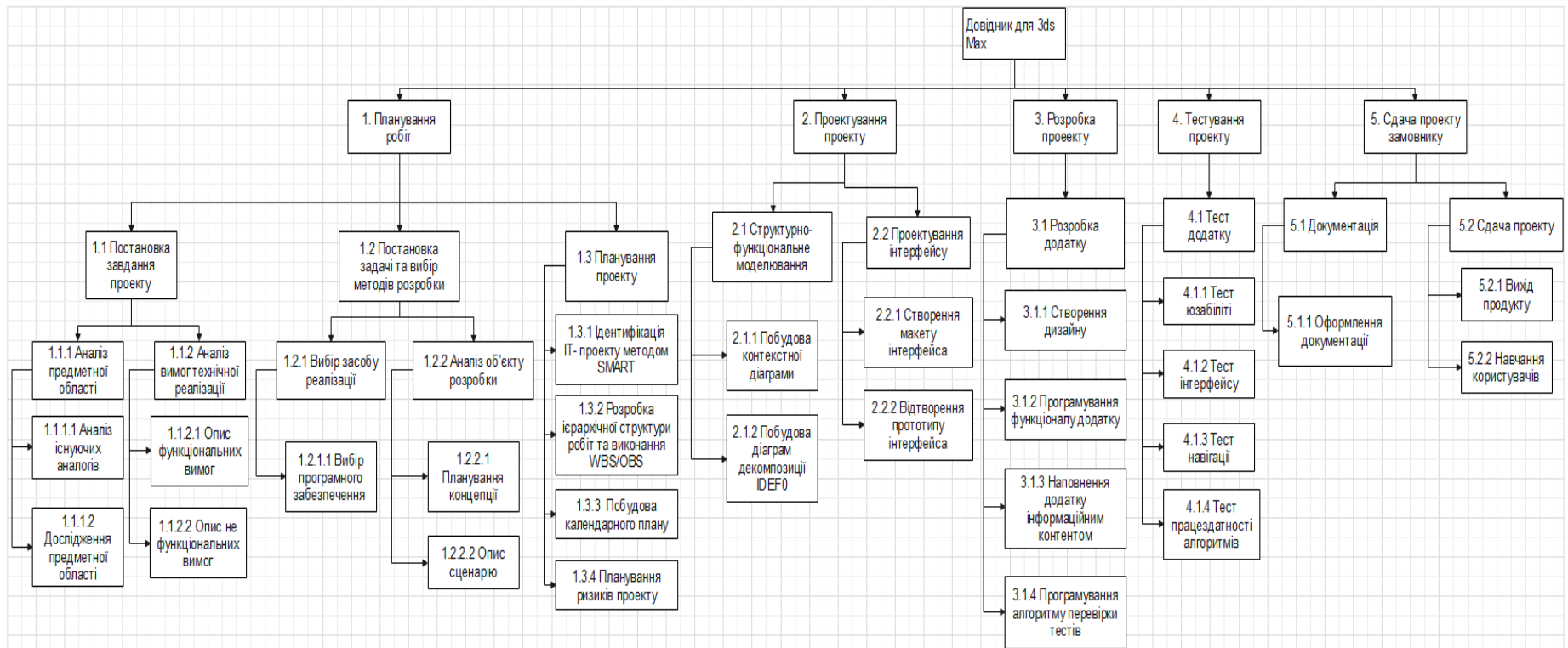


Рисунок Б.1 – WBS. Структура робіт проекту

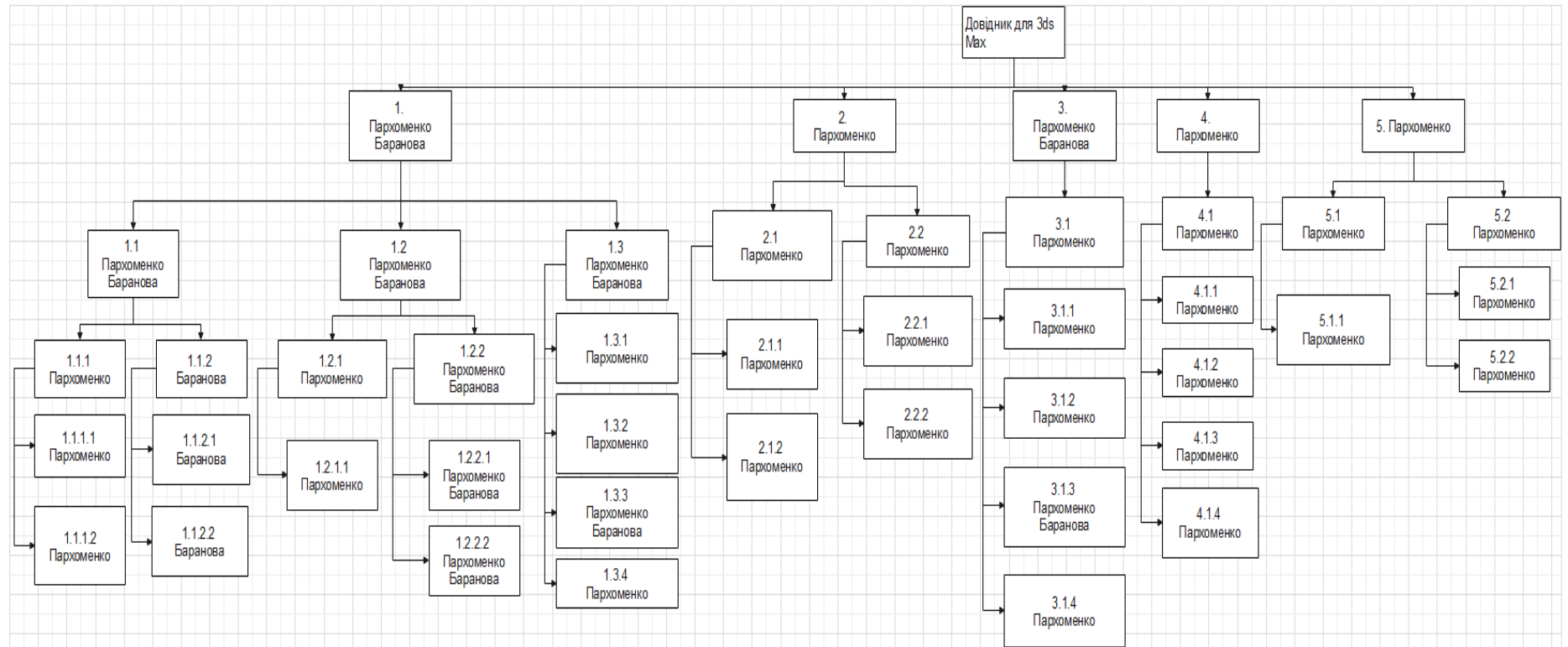


Рисунок Б.2 – Організаційна структура проекту (OBS)

Діаграма Ганта. Далі побудуємо календарний план виконання дипломного проекту. Найпоширеніший формат графіка в будь-якій галузі – діаграма Ганта. Цей графік дозволяє менеджерам проекту і всій команді розробників візуалізувати графіки часу і взаємозв'язок між окремими завданнями та етапами роботи над проектом. Тривалість виконання робіт зазначена в днях, але фактична тривалість виконання робіт приблизно дорівнює 2–3 години на день. Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, з урахуванням вихідних та святкових днів, побудовано календарний графік. Діаграма Ганта та список робіт діаграми Ганта зображені на рис. Б.3–Б.6.



Рисунок Б.3 – Діаграма Ганта

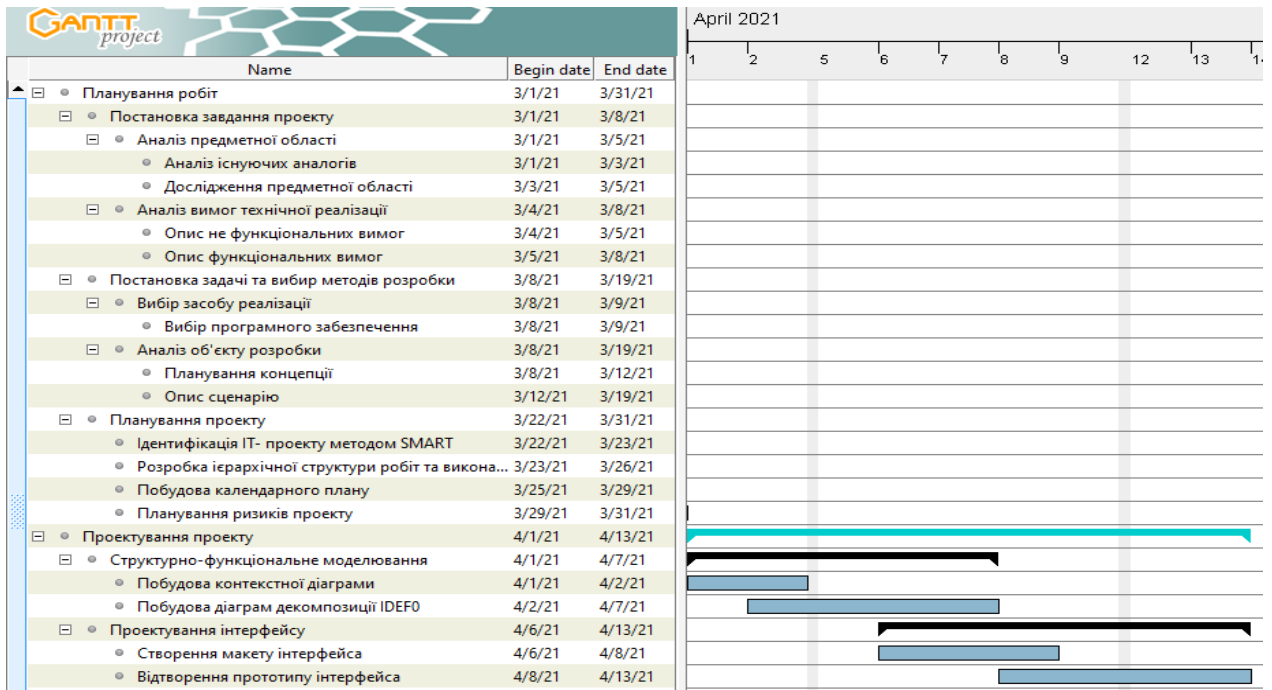


Рисунок Б.4 – Продовження діаграми Ганта

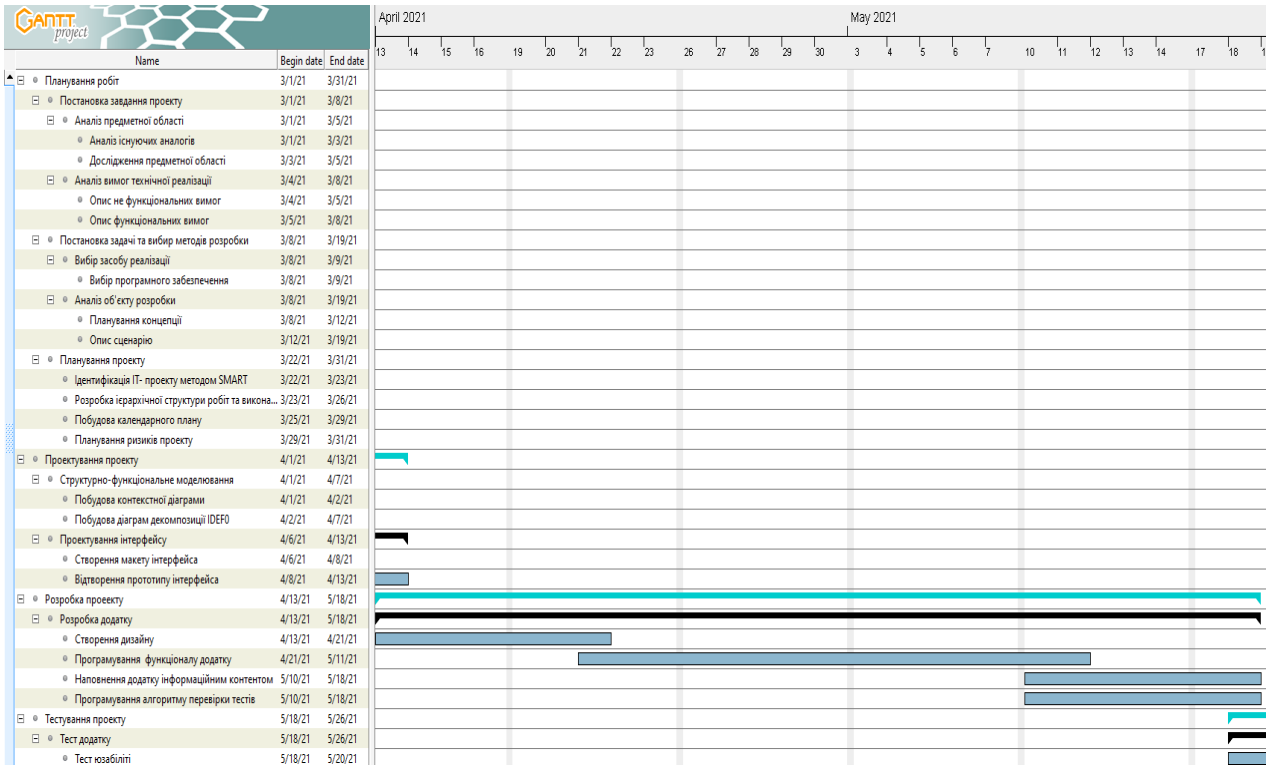


Рисунок Б.5 – Продовження діаграми Ганта

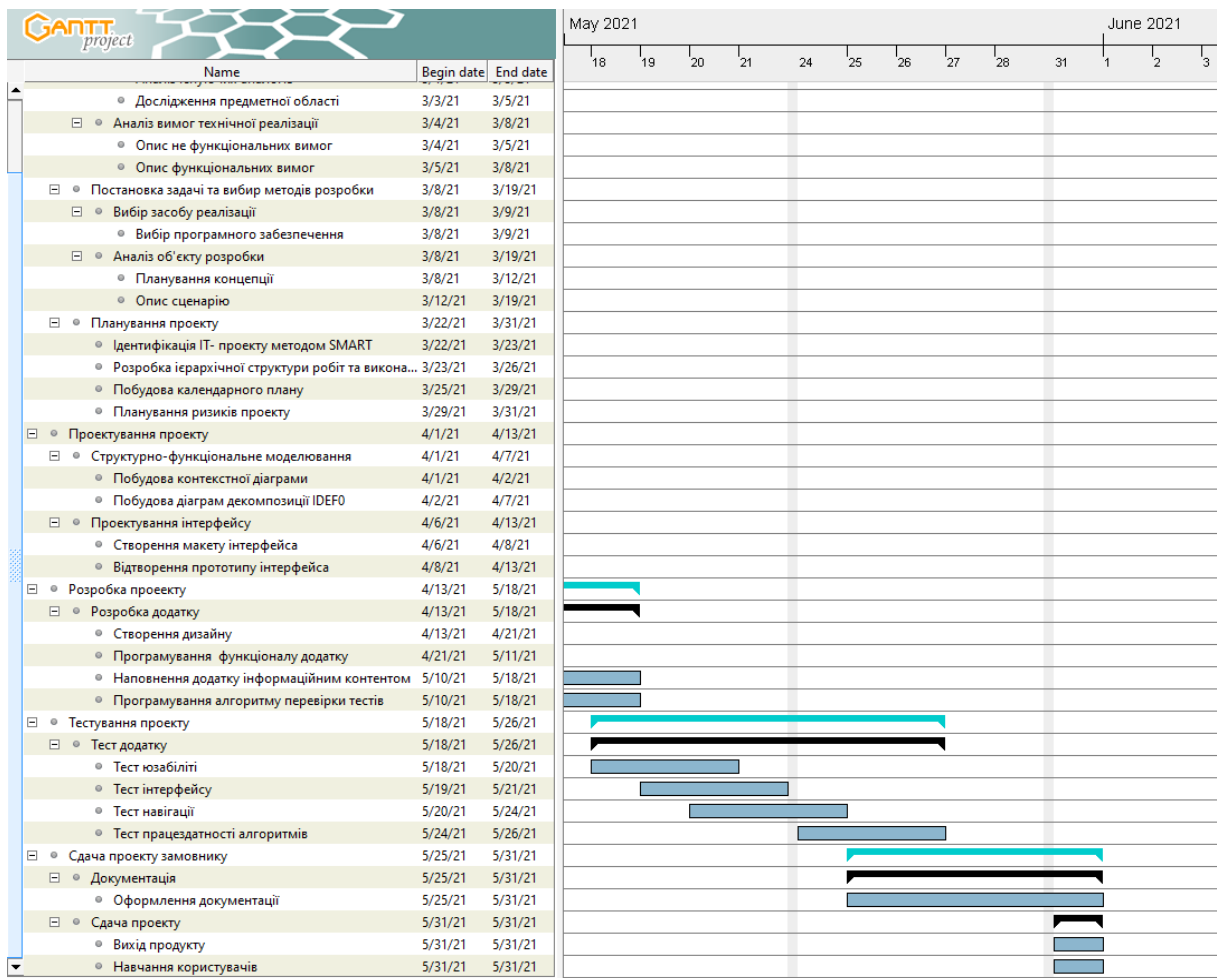


Рисунок Б.6 – Продовження діаграми Ганта

Аналіз ризиків. Виконаємо якісну і кількісну оцінку ризиків роботи. При якісній оцінці визначимо ризики, що потребують швидкого реагування. Така оцінка визначить ступінь важливості ризику і дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде виконана для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання проекту. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків. У табл. Б.5 знаходиться класифікація ризиків за показниками ймовірності виникнення ризику та величини втрат.

Далі виконаємо планування реагування на ризики – це розробка методів і технологій зниження негативного впливу ризиків на проект. Визначимо ефективність розробки реагування на проект, визначимо чи будуть наслідки впливу ризику на проект позитивними або негативним. Оцінюємо ризики за показниками, що знаходяться в табл. Б.3. На основі оцінки будемо матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рис. Б.7.

Таблиця Б.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Ймовірність виникнення	3		RS_7	RS_4
	2		RS_2, RS_6, RS_9	
	1			RS_1, RS_3, RS_5, RS_8, RS_10
		1	2	3
	Вплив ризику			

Рисунок Б.7 – Матриця ймовірності виникнення ризиків та впливу ризику

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. Б.4.

Таблиця Б.4 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	1R2	9
2	Виправдані	3R4	1, 2, 3, 5, 6, 8
3	Недопустимі	6R9	4, 7

Таблиця Б.5 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Поява альтернативного продукту	Низька	Високий	3	1. Провести попереднє дослідження альтернативних продуктів. 2. Вибрати унікальну стратегію створення проекту.	Прийняття	
RS_2	Відкритий	Низький кваліфікаційний рівень співробітників	Середній	Середній	4	1. Правильний розподіл задач, для кожної персони 2. Підвищення навиків персоналу	Прийняття	Враховуючи брак часу, покращення навиків буде проходити вивченням потрібного матеріалу у вільному доступі
RS_3	Відкритий	Помилка у графічному інтерфейсі	Низька	Високий	3	1. Побудова чіткого макету, для подальшого створення інтерфейсу 2. Поетапне відтворення інтерфейсу зі створеного прототипу	Попередження	При виявленні проблеми, відходити на попередні етапи та виправляти виявлені проблеми
RS_4	Відкритий	Неякісна працездатність алгоритму	Висока	Високий	9	1. Поетапна перевірка результату на наявність проблем.	Пом'якшення	Змінити підхід до створення
RS_5	Відкритий	Зміна вимог проекту	Низька	Високий	3	1. Обговорення чіткої концепції проекту. 2. Постійна перевірка плану проекту	Пом'якшення	Переоцінка проекту та плану при кожній зміні вимог

Продовження таблиці Б.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_6	Відкритий	Постійні зміни у ТЗ	Середня	Середній	4	1. Виділити основні етапи проекту 2. Чіткий опис вимог для проекту	Перенос	Внести всі зміни та поправки. Обговорити всі етапи повторно.
RS_7	Відкритий	Неправильний розподіл часу	Середня	Високий	6	1. Чітко відвести потрібну кількість часу, з урахуванням навиків персоналу, важливості етапу та складністю виконання	Пом'якшення	Переоцінити кількість потрібного часу повторно, зменшити відведений час на менш важливих етапах
RS_8	Відкритий	Помилка при оцінюванні масштабів проекту	Низька	Високий	3	1. Провести детальний аналіз проекту. Збільшити увагу на розподілі етапів та встановленні відведеного часу. 2. Проаналізувати масштаби проекту на основі додаткових джерел.	Пом'якшення	Переоцінка масштабів проекту. Перебудова стратегії реалізації проекту.
RS_9	Відкритий	Нечітка поставлена задача	Низький	Середній	2	1. Детально оцінювати кожен етап проекту. 2. Детальний опис кожного етапу	Попередження	Повернутися до попереднього етапу, та перепризначити постановку задачі
RS_10	Відкритий	Вибір не ефективної технології розробки	Низька	Високий	3	1. Детально оцінити всі переваги та недоліки кожного з технологій розробки. Обрати най ефективнішу	Попередження	Переоцінити всі технології повторно, змінити існуючу

ДОДАТОК В

КОД ОСНОВНИХ МОДУЛІВ ДОДАТКУ

BaseViewModel.cs

```

using For3dsMax.Models;
using For3dsMax.Services;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Runtime.CompilerServices;
using Xamarin.Forms;

namespace For3dsMax.ViewModels
{
    public class BaseViewModel : INotifyPropertyChanged
    {
        public IDataStore<Item> DataStore => DependencyService.Get<IDataStore<Item>>();

        bool isBusy = false;
        public bool IsBusy
        {
            get { return isBusy; }
            set { SetProperty(ref isBusy, value); }
        }

        string title = string.Empty;
        public string Title
        {
            get { return title; }
            set { SetProperty(ref title, value); }
        }

        protected bool SetProperty<T>(ref T backingStore, T value,
            [CallerMemberName] string propertyName = "",
            Action onChanged = null)
        {
            if (EqualityComparer<T>.Default.Equals(backingStore, value))
                return false;

            backingStore = value;
            onChanged?.Invoke();
            OnPropertyChanged(propertyName);
            return true;
        }

        #region INotifyPropertyChanged
        public event PropertyChangedEventHandler PropertyChanged;
        protected void OnPropertyChanged([CallerMemberName] string propertyName = "")
        {
            var changed = PropertyChanged;
            if (changed == null)
                return;
        }
    }
}

```

```

        changed.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }
    #endregion
}
}

```

PagePractice1ViewModel.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Input;
using Xamarin.Essentials;
using Xamarin.Forms;

namespace For3dsMax.ViewModels
{
    public class PagePractice1ViewModel : BaseViewModel
    {
        public PagePractice1ViewModel()
        {
            Title = "About";
            OpenWebCommand = new Command(async () => await
Browser.OpenAsync("https://mix.sumdu.edu.ua/textbooks/30251/1138936/%D0%9B%D0%B0%D0%B1_%D1%8
0%D0%BE%D0%B1_01__%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%94%D0%92%D0%92_2021.p
df"));
        }
        public ICommand OpenWebCommand { get; }
    }
}

```

AboutPage.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="For3dsMax.Views.AboutPage"
    Title="Головна">

    <ContentPage.Resources>
        <ResourceDictionary>
            <Color x:Key="Accent">#96d1ff</Color>
        </ResourceDictionary>
    </ContentPage.Resources>
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <StackLayout BackgroundColor="{StaticResource Accent}"
VerticalOptions="FillAndExpand" HorizontalOptions="Fill">
            <StackLayout Orientation="Horizontal" HorizontalOptions="Center"
VerticalOptions="Center">
                <ContentView Padding="0,40,0,40" VerticalOptions="FillAndExpand">
                    <Label Text="3ds Max Tutorial"
                        FontSize="Title"
                        FontAttributes="Bold"/>
                </ContentView>
            </StackLayout>
        </StackLayout>
    </Grid>

```

```

        <ScrollView Grid.Row="1">
            <StackLayout Orientation="Vertical" Padding="30,24,30,24" Spacing="10">
                <Image Source="logo.png" VerticalOptions="Center" HeightRequest="300" />
            </StackLayout>
        </ScrollView>
    </Grid>
</ContentPage>

```

AboutPage.xaml.cs

```

using System;
using System.ComponentModel;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace For3dsMax.Views
{
    public partial class AboutPage : ContentPage
    {
        public AboutPage()
        {
            InitializeComponent();
        }
    }
}

```

PagePractice1.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="For3dsMax.Views.PagePractice1"
    xmlns:vm1="clr-namespace:For3dsMax.ViewModels"
    Title="Інтерфейс">
    <ContentPage.BindingContext>
        <vm1:PagePractice1ViewModel />
    </ContentPage.BindingContext>
    <ContentPage.Content>
        <StackLayout Margin="15,15,15,15">
            <Label Text="ОСНОВИ СТВОРЕННЯ ТА РОБОТИ З ПРОСТИМИ ОБ'ЄКТАМИ"
                HorizontalOptions="CenterAndExpand"
                FontAttributes="Italic"
                FontSize="Subtitle"/>
            <Button Text="Відкрити файл"
                Command="{Binding OpenWebCommand}"
                BackgroundColor="{StaticResource Primary}"
                TextColor="White" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

PagePractice1.xaml.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;

using Xamarin.Forms;

```

```

using Xamarin.Forms.Xaml;

namespace For3dsMax.Views
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PagePractice1 : ContentPage
    {
        public PagePractice1()
        {
            InitializeComponent();
        }
    }
}

```

PageTest1.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="For3dsMax.Views.PageTest2"
    Title="Гарячі клавіші">
    <ContentPage.Content>
        <ScrollView>
            <StackLayout Padding="15,15,15,15">
                <Label Padding="0,15,0,0" Text="1. Виберіть вірну відповідь" FontSize="Body"
                    FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
                <Label x:Name="lb1" Text="In this label write some text..." FontSize="Body"
                    FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
                <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img1"
                    Source=""></ImageButton>
                <RadioButton x:Name="rb11" GroupName="RBGroup1" Content="" ></RadioButton>
                <RadioButton x:Name="rb12" GroupName="RBGroup1" Content="" ></RadioButton>
                <RadioButton x:Name="rb13" GroupName="RBGroup1" Content="" ></RadioButton>
                <RadioButton x:Name="rb14" GroupName="RBGroup1" Content="" ></RadioButton>

                <Label Padding="0,15,0,0" Text="2. Виберіть вірну відповідь" FontSize="Body"
                    FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
                <Label x:Name="lb2" Text="In this label write some text..." FontSize="Body"
                    FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
                <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img2"
                    Source=""></ImageButton>
                <RadioButton x:Name="rb21" GroupName="RBGroup2" Content="" ></RadioButton>
                <RadioButton x:Name="rb22" GroupName="RBGroup2" Content="" ></RadioButton>
                <RadioButton x:Name="rb23" GroupName="RBGroup2" Content="" ></RadioButton>
                <RadioButton x:Name="rb24" GroupName="RBGroup2" Content="" ></RadioButton>

                <Label Padding="0,15,0,0" Text="3. Виберіть вірну відповідь" FontSize="Body"
                    FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
                <Label x:Name="lb3" Text="In this label write some text..." FontSize="Body"
                    FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
                <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img3"
                    Source=""></ImageButton>
                <RadioButton x:Name="rb31" GroupName="RBGroup3" Content="" ></RadioButton>
                <RadioButton x:Name="rb32" GroupName="RBGroup3" Content="" ></RadioButton>
                <RadioButton x:Name="rb33" GroupName="RBGroup3" Content="" ></RadioButton>
                <RadioButton x:Name="rb34" GroupName="RBGroup3" Content="" ></RadioButton>

                <Label Padding="0,15,0,0" Text="4. Виберіть вірну відповідь" FontSize="Body"
                    FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
                <Label x:Name="lb4" Text="In this label write some text..." FontSize="Body"
                    FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
            </StackLayout>
        </ScrollView>
    </ContentPage.Content>
</ContentPage>

```

```

        <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img4"
Source=""></ImageButton>
        <RadioButton x:Name="rb41" GroupName="RBGroup4" Content="" ></RadioButton>
        <RadioButton x:Name="rb42" GroupName="RBGroup4" Content="" ></RadioButton>
        <RadioButton x:Name="rb43" GroupName="RBGroup4" Content="" ></RadioButton>
        <RadioButton x:Name="rb44" GroupName="RBGroup4" Content="" ></RadioButton>

        <Label Padding="0,15,0,0" Text="5. Виберіть вірну відповідь" FontSize="Body"
FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
        <Label x:Name="lb5" Text="In this label write some text..." FontSize="Body"
FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
        <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img5"
Source=""></ImageButton>
        <RadioButton x:Name="rb51" GroupName="RBGroup5" Content="" ></RadioButton>
        <RadioButton x:Name="rb52" GroupName="RBGroup5" Content="" ></RadioButton>
        <RadioButton x:Name="rb53" GroupName="RBGroup5" Content="" ></RadioButton>
        <RadioButton x:Name="rb54" GroupName="RBGroup5" Content="" ></RadioButton>

        <Label Padding="0,15,0,0" Text="6. Виберіть вірну відповідь"
FontSize="Body" FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
        <Label x:Name="lb6" Text="In this label write some text..." FontSize="Body"
FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
        <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img6"
Source=""></ImageButton>
        <RadioButton x:Name="rb61" GroupName="RBGroup6" Content="" ></RadioButton>
        <RadioButton x:Name="rb62" GroupName="RBGroup6" Content="" ></RadioButton>
        <RadioButton x:Name="rb63" GroupName="RBGroup6" Content="" ></RadioButton>
        <RadioButton x:Name="rb64" GroupName="RBGroup6" Content="" ></RadioButton>

        <Label Padding="0,15,0,0" Text="7. Виберіть вірну відповідь" FontSize="Body"
FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
        <Label x:Name="lb7" Text="In this label write some text..." FontSize="Body"
FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
        <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img7"
Source=""></ImageButton>
        <RadioButton x:Name="rb71" GroupName="RBGroup7" Content="" ></RadioButton>
        <RadioButton x:Name="rb72" GroupName="RBGroup7" Content="" ></RadioButton>
        <RadioButton x:Name="rb73" GroupName="RBGroup7" Content="" ></RadioButton>
        <RadioButton x:Name="rb74" GroupName="RBGroup7" Content="" ></RadioButton>

        <Label Padding="0,15,0,0" Text="8. Виберіть вірну відповідь" FontSize="Body"
FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
        <Label x:Name="lb8" Text="In this label write some text..." FontSize="Body"
FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
        <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img8"
Source=""></ImageButton>
        <RadioButton x:Name="rb81" GroupName="RBGroup8" Content="" ></RadioButton>
        <RadioButton x:Name="rb82" GroupName="RBGroup8" Content="" ></RadioButton>
        <RadioButton x:Name="rb83" GroupName="RBGroup8" Content="" ></RadioButton>
        <RadioButton x:Name="rb84" GroupName="RBGroup8" Content="" ></RadioButton>

        <Label Padding="0,15,0,0" Text="9. Виберіть вірну відповідь" FontSize="Body"
FontAttributes="Bold" FontFamily="18" HorizontalOptions="Start"></Label>
        <Label x:Name="lb9" Text="In this label write some text..." FontSize="Body"
FontAttributes="Italic" FontFamily="18" HorizontalOptions="Start"></Label>
        <ImageButton VerticalOptions="Fill" HorizontalOptions="Fill" x:Name="Img9"
Source=""></ImageButton>
        <RadioButton x:Name="rb91" GroupName="RBGroup9" Content="" ></RadioButton>
        <RadioButton x:Name="rb92" GroupName="RBGroup9" Content="" ></RadioButton>
        <RadioButton x:Name="rb93" GroupName="RBGroup9" Content="" ></RadioButton>
        <RadioButton x:Name="rb94" GroupName="RBGroup9" Content="" ></RadioButton>

```



```

private void CreateEvents()
{
    rb11.CheckedChanged += Rb11_CheckedChanged;
    rb12.CheckedChanged += Rb12_CheckedChanged;
    rb13.CheckedChanged += Rb13_CheckedChanged;
    rb14.CheckedChanged += Rb14_CheckedChanged;
    rb21.CheckedChanged += Rb21_CheckedChanged;
    rb22.CheckedChanged += Rb22_CheckedChanged;
    rb23.CheckedChanged += Rb23_CheckedChanged;
    rb24.CheckedChanged += Rb24_CheckedChanged;
    rb31.CheckedChanged += Rb31_CheckedChanged;
    rb32.CheckedChanged += Rb32_CheckedChanged;
    rb33.CheckedChanged += Rb33_CheckedChanged;
    rb34.CheckedChanged += Rb34_CheckedChanged;
    rb41.CheckedChanged += Rb41_CheckedChanged;
    rb42.CheckedChanged += Rb42_CheckedChanged;
    rb43.CheckedChanged += Rb43_CheckedChanged;
    rb44.CheckedChanged += Rb44_CheckedChanged;
    rb51.CheckedChanged += Rb51_CheckedChanged;
    rb52.CheckedChanged += Rb52_CheckedChanged;
    rb53.CheckedChanged += Rb53_CheckedChanged;
    rb54.CheckedChanged += Rb54_CheckedChanged;
    rb61.CheckedChanged += Rb61_CheckedChanged;
    rb62.CheckedChanged += Rb62_CheckedChanged;
    rb63.CheckedChanged += Rb63_CheckedChanged;
    rb64.CheckedChanged += Rb64_CheckedChanged;
    rb71.CheckedChanged += Rb71_CheckedChanged;
    rb72.CheckedChanged += Rb72_CheckedChanged;
    rb73.CheckedChanged += Rb73_CheckedChanged;
    rb74.CheckedChanged += Rb74_CheckedChanged;
    rb81.CheckedChanged += Rb81_CheckedChanged;
    rb82.CheckedChanged += Rb82_CheckedChanged;
    rb83.CheckedChanged += Rb83_CheckedChanged;
    rb84.CheckedChanged += Rb84_CheckedChanged;
    rb91.CheckedChanged += Rb91_CheckedChanged;
    rb92.CheckedChanged += Rb92_CheckedChanged;
    rb93.CheckedChanged += Rb93_CheckedChanged;
    rb94.CheckedChanged += Rb94_CheckedChanged;
    rb101.CheckedChanged += Rb101_CheckedChanged;
    rb102.CheckedChanged += Rb102_CheckedChanged;
    rb103.CheckedChanged += Rb103_CheckedChanged;
    rb104.CheckedChanged += Rb104_CheckedChanged;
    btn1.Clicked += Btn1_Clicked;
}

private void CheckedIsTorF(Boolean TorF)
{
    rb11.IsChecked = TorF;
    rb12.IsChecked = TorF;
    rb13.IsChecked = TorF;
    rb14.IsChecked = TorF;

    rb21.IsChecked = TorF;
    rb22.IsChecked = TorF;
    rb23.IsChecked = TorF;
    rb24.IsChecked = TorF;

    rb31.IsChecked = TorF;
    rb32.IsChecked = TorF;
    rb33.IsChecked = TorF;
    rb34.IsChecked = TorF;
}

```

```

        rb41.IsChecked = TorF;
        rb42.IsChecked = TorF;
        rb43.IsChecked = TorF;
        rb44.IsChecked = TorF;

        rb51.IsChecked = TorF;
        rb52.IsChecked = TorF;
        rb53.IsChecked = TorF;
        rb54.IsChecked = TorF;

        rb61.IsChecked = TorF;
        rb62.IsChecked = TorF;
        rb63.IsChecked = TorF;
        rb64.IsChecked = TorF;

        rb71.IsChecked = TorF;
        rb72.IsChecked = TorF;
        rb73.IsChecked = TorF;
        rb74.IsChecked = TorF;

        rb81.IsChecked = TorF;
        rb82.IsChecked = TorF;
        rb83.IsChecked = TorF;
        rb84.IsChecked = TorF;

        rb91.IsChecked = TorF;
        rb92.IsChecked = TorF;
        rb93.IsChecked = TorF;
        rb94.IsChecked = TorF;

        rb101.IsChecked = TorF;
        rb102.IsChecked = TorF;
        rb103.IsChecked = TorF;
        rb104.IsChecked = TorF;
    }

    private void InitializationText(Label label, string[,] Array, int count1, int[]
count2, int i)
    {
        label.Text = Convert.ToString(Array[count1, count2[i]]);
    }

    private void InitializationSource(ImageButton imagebutton, string[,] Array, int
count1, int[] count2, int i)
    {
        imagebutton.Source = Convert.ToString(Array[count1, count2[i]]);
    }

    private void RandomTasks()
    {
        Random rnd = new Random();
        Count[0] = rnd.Next(0, 15);
        do
        {
            Count[1] = rnd.Next(0, 15);
        }
        while (Count[1] == Count[0]);
        do
        {
            Count[2] = rnd.Next(0, 15);
        }
    }

```



```

while (Count[2] == Count[0] || Count[2] == Count[1]);
do
{
    Count[3] = rnd.Next(0, 15);
}
while (Count[3] == Count[0] || Count[3] == Count[1] || Count[3] == Count[2]);
do
{
    Count[4] = rnd.Next(0, 15);
}
while (Count[4] == Count[0] || Count[4] == Count[1] || Count[4] == Count[2] ||
Count[4] == Count[3]);
do
{
    Count[5] = rnd.Next(0, 15);
}
while (Count[5] == Count[0] || Count[5] == Count[1] || Count[5] == Count[2] ||
Count[5] == Count[3] || Count[5] == Count[4]);
do
{
    Count[6] = rnd.Next(0, 15);
}
while (Count[6] == Count[0] || Count[6] == Count[1] || Count[6] == Count[2] ||
Count[6] == Count[3] || Count[6] == Count[4] || Count[6] == Count[5]);
do
{
    Count[7] = rnd.Next(0, 15);
}
while (Count[7] == Count[0] || Count[7] == Count[1] || Count[7] == Count[2] ||
Count[7] == Count[3] || Count[7] == Count[4] || Count[7] == Count[5] || Count[7] == Count[6]);
do
{
    Count[8] = rnd.Next(0, 15);
}
while (Count[8] == Count[0] || Count[8] == Count[1] || Count[8] == Count[2] ||
Count[8] == Count[3] || Count[8] == Count[4] || Count[8] == Count[5] || Count[8] == Count[6]
|| Count[8] == Count[7]);
do
{
    Count[9] = rnd.Next(0, 15);
}
while (Count[9] == Count[0] || Count[9] == Count[1] || Count[9] == Count[2] ||
Count[9] == Count[3] || Count[9] == Count[4] || Count[9] == Count[5] || Count[9] == Count[6]
|| Count[9] == Count[7] || Count[9] == Count[8]);
}

private void RandomTexts(RadioButton rb1, RadioButton rb2, RadioButton rb3,
RadioButton rb4, int Count, string[,] aArray)
{
    Random rnd = new Random();
    value = rnd.Next(1, 5);
    rb1.Content = aArray[value, Count];
    if (rb1.Content == Mas[1, Count])
    {
        rb1.Value = 1;
    }
    do
    {
        value = rnd.Next(1, 5);
        rb2.Content = aArray[value, Count];
    }
}

```

```

while (rb2.Content == rb1.Content);
if (rb2.Content == Mas[1, Count])
{
    rb2.Value = 1;
}
do
{
    value = rnd.Next(1, 5);
    rb3.Content = aArray[value, Count];
}
while (rb3.Content == rb1.Content || rb3.Content == rb2.Content);
if (rb3.Content == Mas[1, Count])
{
    rb3.Value = 1;
}
do
{
    value = rnd.Next(1, 5);
    rb4.Content = aArray[value, Count];
}
while (rb4.Content == rb1.Content || rb4.Content == rb2.Content || rb4.Content ==
rb3.Content);
if (rb4.Content == Mas[1, Count])
{
    rb4.Value = 1;
}
}

public PageTest2()
{
    InitializeComponent();
    CreateEvents();

    RandomTasks();

    InitializationText(lb1, Mas, 0, Count, 0);
    InitializationText(lb2, Mas, 0, Count, 1);
    InitializationText(lb3, Mas, 0, Count, 2);
    InitializationText(lb4, Mas, 0, Count, 3);
    InitializationText(lb5, Mas, 0, Count, 4);
    InitializationText(lb6, Mas, 0, Count, 5);
    InitializationText(lb7, Mas, 0, Count, 6);
    InitializationText(lb8, Mas, 0, Count, 7);
    InitializationText(lb9, Mas, 0, Count, 8);
    InitializationText(lb10, Mas, 0, Count, 9);

    InitializationSource(Img1, Mas, 5, Count, 0);
    InitializationSource(Img2, Mas, 5, Count, 1);
    InitializationSource(Img3, Mas, 5, Count, 2);
    InitializationSource(Img4, Mas, 5, Count, 3);
    InitializationSource(Img5, Mas, 5, Count, 4);
    InitializationSource(Img6, Mas, 5, Count, 5);
    InitializationSource(Img7, Mas, 5, Count, 6);
    InitializationSource(Img8, Mas, 5, Count, 7);
    InitializationSource(Img9, Mas, 5, Count, 8);
    InitializationSource(Img10, Mas, 5, Count, 9);

    RandomTexts(rb11, rb12, rb13, rb14, Count[0], Mas);
    RandomTexts(rb21, rb22, rb23, rb24, Count[1], Mas);
    RandomTexts(rb31, rb32, rb33, rb34, Count[2], Mas);
    RandomTexts(rb41, rb42, rb43, rb44, Count[3], Mas);

```

```

        RandomTexts(rb51, rb52, rb53, rb54, Count[4], Mas);
        RandomTexts(rb61, rb62, rb63, rb64, Count[5], Mas);
        RandomTexts(rb71, rb72, rb73, rb74, Count[6], Mas);
        RandomTexts(rb81, rb82, rb83, rb84, Count[7], Mas);
        RandomTexts(rb91, rb92, rb93, rb94, Count[8], Mas);
        RandomTexts(rb101, rb102, rb103, rb104, Count[9], Mas);
    }

    protected override void OnAppearing()
    {

    }

    protected override void OnDisappearing()
    {
        CheckedIsTorF(false);
    }

    private async void Btn1_Clicked(object sender, EventArgs e)
    {
        Answer = 0;
        Answer = Answer + CheckRBGValue[0] + CheckRBGValue[1] + CheckRBGValue[2] +
        CheckRBGValue[3] + CheckRBGValue[4] +
        CheckRBGValue[5] + CheckRBGValue[6] + CheckRBGValue[7] + CheckRBGValue[8] +
        CheckRBGValue[9];
        await DisplayAlert("Гарячі Клавіші", "Ваш результат - " + Convert.ToString(Answer)
        + "%", "Ok");
    }

    private void Rb104_CheckedChanged(object sender, CheckedChangedEventArgs e)
    {
        if (rb104.IsChecked && rb104.Value != null)
        {
            CheckRBGValue[9] = 10;
        }
        else
        {
            CheckRBGValue[9] = 0;
        }
    }

    private void Rb103_CheckedChanged(object sender, CheckedChangedEventArgs e)
    {
        if (rb103.IsChecked && rb103.Value != null)
        {
            CheckRBGValue[9] = 10;
        }
        else
        {
            CheckRBGValue[9] = 0;
        }
    }

    private void Rb102_CheckedChanged(object sender, CheckedChangedEventArgs e)
    {
        if (rb102.IsChecked && rb102.Value != null)
        {
            CheckRBGValue[9] = 10;
        }
        else
        {

```

```
        CheckRBGValue[9] = 0;
    }
}

private void Rb101_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb101.IsChecked && rb101.Value != null)
    {
        CheckRBGValue[9] = 10;
    }
    else
    {
        CheckRBGValue[9] = 0;
    }
}

private void Rb94_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb94.IsChecked && rb94.Value != null)
    {
        CheckRBGValue[8] = 10;
    }
    else
    {
        CheckRBGValue[8] = 0;
    }
}

private void Rb93_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb93.IsChecked && rb93.Value != null)
    {
        CheckRBGValue[8] = 10;
    }
    else
    {
        CheckRBGValue[8] = 0;
    }
}

private void Rb92_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb92.IsChecked && rb92.Value != null)
    {
        CheckRBGValue[8] = 10;
    }
    else
    {
        CheckRBGValue[8] = 0;
    }
}

private void Rb91_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb91.IsChecked && rb91.Value != null)
    {
        CheckRBGValue[8] = 10;
    }
    else
    {
        CheckRBGValue[8] = 0;
    }
}
```

```
    }  
}  
  
private void Rb84_CheckedChanged(object sender, CheckedChangedEventArgs e)  
{  
    if (rb84.IsChecked && rb84.Value != null)  
    {  
        CheckRBGValue[7] = 10;  
    }  
    else  
    {  
        CheckRBGValue[7] = 0;  
    }  
}  
  
private void Rb83_CheckedChanged(object sender, CheckedChangedEventArgs e)  
{  
    if (rb83.IsChecked && rb83.Value != null)  
    {  
        CheckRBGValue[7] = 10;  
    }  
    else  
    {  
        CheckRBGValue[7] = 0;  
    }  
}  
  
private void Rb82_CheckedChanged(object sender, CheckedChangedEventArgs e)  
{  
    if (rb82.IsChecked && rb82.Value != null)  
    {  
        CheckRBGValue[7] = 10;  
    }  
    else  
    {  
        CheckRBGValue[7] = 0;  
    }  
}  
  
private void Rb81_CheckedChanged(object sender, CheckedChangedEventArgs e)  
{  
    if (rb81.IsChecked && rb81.Value != null)  
    {  
        CheckRBGValue[7] = 10;  
    }  
    else  
    {  
        CheckRBGValue[7] = 0;  
    }  
}  
  
private void Rb74_CheckedChanged(object sender, CheckedChangedEventArgs e)  
{  
    if (rb74.IsChecked && rb74.Value != null)  
    {  
        CheckRBGValue[6] = 10;  
    }  
    else  
    {  
        CheckRBGValue[6] = 0;  
    }  
}
```

```
}

private void Rb73_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb73.IsChecked && rb73.Value != null)
    {
        CheckRBGValue[6] = 10;
    }
    else
    {
        CheckRBGValue[6] = 0;
    }
}

private void Rb72_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb72.IsChecked && rb72.Value != null)
    {
        CheckRBGValue[6] = 10;
    }
    else
    {
        CheckRBGValue[6] = 0;
    }
}

private void Rb71_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb71.IsChecked && rb71.Value != null)
    {
        CheckRBGValue[6] = 10;
    }
    else
    {
        CheckRBGValue[6] = 0;
    }
}

private void Rb64_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb64.IsChecked && rb64.Value != null)
    {
        CheckRBGValue[5] = 10;
    }
    else
    {
        CheckRBGValue[5] = 0;
    }
}

private void Rb63_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb63.IsChecked && rb63.Value != null)
    {
        CheckRBGValue[5] = 10;
    }
    else
    {
        CheckRBGValue[5] = 0;
    }
}
```

```
private void Rb62_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb62.IsChecked && rb62.Value != null)
    {
        CheckRBGValue[5] = 10;
    }
    else
    {
        CheckRBGValue[5] = 0;
    }
}

private void Rb61_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb61.IsChecked && rb61.Value != null)
    {
        CheckRBGValue[5] = 10;
    }
    else
    {
        CheckRBGValue[5] = 0;
    }
}

private void Rb54_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb54.IsChecked && rb54.Value != null)
    {
        CheckRBGValue[4] = 10;
    }
    else
    {
        CheckRBGValue[4] = 0;
    }
}

private void Rb53_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb53.IsChecked && rb53.Value != null)
    {
        CheckRBGValue[4] = 10;
    }
    else
    {
        CheckRBGValue[4] = 0;
    }
}

private void Rb52_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb52.IsChecked && rb52.Value != null)
    {
        CheckRBGValue[4] = 10;
    }
    else
    {
        CheckRBGValue[4] = 0;
    }
}
```

```
private void Rb51_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb51.IsChecked && rb51.Value != null)
    {
        CheckRBGValue[4] = 10;
    }
    else
    {
        CheckRBGValue[4] = 0;
    }
}

private void Rb44_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb44.IsChecked && rb44.Value != null)
    {
        CheckRBGValue[3] = 10;
    }
    else
    {
        CheckRBGValue[3] = 0;
    }
}

private void Rb43_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb43.IsChecked && rb43.Value != null)
    {
        CheckRBGValue[3] = 10;
    }
    else
    {
        CheckRBGValue[3] = 0;
    }
}

private void Rb42_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb42.IsChecked && rb42.Value != null)
    {
        CheckRBGValue[3] = 10;
    }
    else
    {
        CheckRBGValue[3] = 0;
    }
}

private void Rb41_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb41.IsChecked && rb41.Value != null)
    {
        CheckRBGValue[3] = 10;
    }
    else
    {
        CheckRBGValue[3] = 0;
    }
}

private void Rb34_CheckedChanged(object sender, CheckedChangedEventArgs e)
```



```
{
    if (rb34.IsChecked && rb34.Value != null)
    {
        CheckRBGValue[2] = 10;
    }
    else
    {
        CheckRBGValue[2] = 0;
    }
}

private void Rb33_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb32.IsChecked && rb32.Value != null)
    {
        CheckRBGValue[2] = 10;
    }
    else
    {
        CheckRBGValue[2] = 0;
    }
}

private void Rb32_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb32.IsChecked && rb32.Value != null)
    {
        CheckRBGValue[2] = 10;
    }
    else
    {
        CheckRBGValue[2] = 0;
    }
}

private void Rb31_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb31.IsChecked && rb31.Value != null)
    {
        CheckRBGValue[2] = 10;
    }
    else
    {
        CheckRBGValue[2] = 0;
    }
}

private void Rb24_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb24.IsChecked && rb24.Value != null)
    {
        CheckRBGValue[1] = 10;
    }
    else
    {
        CheckRBGValue[1] = 0;
    }
}

private void Rb23_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
```

```

        if (rb23.IsChecked && rb23.Value != null)
        {
            CheckRBGValue[1] = 10;
        }
        else
        {
            CheckRBGValue[1] = 0;
        }
    }

private void Rb22_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb22.IsChecked && rb22.Value != null)
    {
        CheckRBGValue[1] = 10;
    }
    else
    {
        CheckRBGValue[1] = 0;
    }
}

private void Rb21_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb21.IsChecked && rb21.Value != null)
    {
        CheckRBGValue[1] = 10;
    }
    else
    {
        CheckRBGValue[1] = 0;
    }
}

private void Rb14_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb14.IsChecked && rb14.Value != null)
    {
        CheckRBGValue[0] = 10;
    }
    else
    {
        CheckRBGValue[0] = 0;
    }
}

private void Rb13_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb13.IsChecked == true && rb13.Value != null)
    {
        CheckRBGValue[0] = 10;
    }
    else
    {
        CheckRBGValue[0] = 0;
    }
}

private void Rb12_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb12.IsChecked == true && rb12.Value != null)

```

```

    {
        CheckRBGValue[0] = 10;
    }
    else
    {
        CheckRBGValue[0] = 0;
    }
}

private void Rb11_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    if (rb11.IsChecked == true && rb11.Value != null)
    {
        CheckRBGValue[0] = 10;
    }
    else
    {
        CheckRBGValue[0] = 0;
    }
}
}
}
}

```

PageTheory1.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="For3dsMax.Views.PageTheory1"
    Title="Інтерфейс програми">
<ContentPage.Content>
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <ScrollView>
            <StackLayout Grid.Row="1"
                x:Name="stack"
                Margin="15, 15, 15, 15"
                Padding="5"
                WidthRequest="400">
                <Label Text="1 Інтерфейс програми"
                    HorizontalOptions="CenterAndExpand"
                    FontAttributes="Bold"
                    FontSize="Title"/>
                <Label Text="1.1 Головне вікно"
                    HorizontalOptions="CenterAndExpand"
                    FontAttributes="Bold"

```

```

        FontSize="Title"/>
        <Label Text="За замовчуванням сцена 3d Max відображається в чотирьох
вікнах проекцій - вид зліва, зверху, спереду і в перспективі (рис. 1.1). При цьому в трьох
вікнах об'єкти показані каркасно, а у вікні перспективи - в тривимірному поданні. Вікно
проекції, в якому на даний момент ведеться робота, підсвічується жовтим кольором і називається
активним. Активне вікно можна розгорнути на весь екран за допомогою кнопки Min / Max Toggle
(Збільшення вікна проекції до розмірів екрану) у правому нижньому кутку вікна 3ds Max."
        HorizontalOptions="StartAndExpand"
        FontAttributes="None"
        FontSize="Default"/>
        <Image      Source="https://mix.sumdu.edu.ua/textbooks/30251/481361/file-
assets/2.1_1_okno_.png"/>
        <Image      Source="https://mix.sumdu.edu.ua/textbooks/30251/481361/file-
assets/2.1_4_okna_.png"/>
        <Label Text="Рис. 1.1. Вікно програми 3ds Max"
        HorizontalOptions="CenterAndExpand"
        FontAttributes="Italic"
        FontSize="Default"/>
        <Label Text="Співвідношення розмірів вікон проекцій можна змінювати
аналогічно зміні розміру діалогових вікон Windows: підведіть покажчик миші до межі між вікнами
(при цьому покажчик набуде вигляду двобічної стрілки), натисніть ліву кнопку миші і, утримуючи
її, перемістіть покажчик на потрібну відстань (рис 1.2). Для виконання зворотної операції
підведіть покажчик миші до кордону між вікнами проекцій, клацніть правою кнопкою миші і в
контекстному меню виберіть команду Reset Layout (Скинути положення)."
        HorizontalOptions="StartAndExpand"
        FontAttributes="None"
        FontSize="Default"
        Padding="0,0,0,80"/>
        <Image      Scale="5"
Source="https://mix.sumdu.edu.ua/textbooks/30251/481361/file-assets/Bezemyannyj.jpg"/>
        <Label Padding="0,0,0,120"/>
        <Image      Scale="5"
Source="https://mix.sumdu.edu.ua/textbooks/30251/481361/file-assets/2.2.png"/>
        <Label Text="Рис. 1.2. Зміна розмірів вікна проекцій"
        HorizontalOptions="CenterAndExpand"
        FontAttributes="Italic"
        FontSize="Default"
        Padding="0,80,0,0"/>
        <Label Text="За необхідності вид в кожному з вікон проекцій можна
змінювати. На сцену можна дивитися не тільки спереду, зверху і зліва, а й справа, знизу і

```

ззаду. Щоб змінити вигляд у вікні проєкції потрібно клацнути на назві вікна правою кнопкою миші і вибрати варіант відображення сцени в меню Views (Вид)."

```
HorizontalOptions="StartAndExpand"
```

```
FontAttributes="None"
```

```
FontSize="Default"/>
```

<Label Text="У тривимірному просторі за замовчуванням немає об'єктів, тому щоб в ньому було легше орієнтуватися, в вікна проєкцій додана допоміжна сітка. Її центр збігається з умовним центром віртуального простору. Для кожного вікна проєкції можна включити або виключити відображення сітки. Для цього виберіть в контекстному меню вікна проєкції команду Show Grid (Показати сітку)."

```
HorizontalOptions="StartAndExpand"
```

```
FontAttributes="None"
```

```
FontSize="Default"/>
```

<Label Text="У верхній частині вікна програми розташовано головне меню, а під ним - головна панель інструментів Main Toolbar (Основна панель інструментів). Пункти головного меню частково повторюють інструменти і команди основний панелі інструментів, а також панелі Command Panel (Командна панель)."

```
HorizontalOptions="StartAndExpand"
```

```
FontAttributes="None"
```

```
FontSize="Default"/>
```

<Label Text="Command Panel (Командна панель) розташована в правій частині вікна програми (рис. 1.3.). Вона містить налаштування всіх об'єктів сцени, а також параметри багатьох операцій, які використовуються в роботі. За допомогою командної панелі можна створювати об'єкти і управляти ними."

```
HorizontalOptions="StartAndExpand"
```

```
FontAttributes="None"
```

```
FontSize="Default"
```

```
Padding="0,0,0,100"/>
```

```
<Image
```

```
Scale="2.2"
```

```
Source="https://mix.sumdu.edu.ua/textbooks/30251/481361/file-assets/2.3.png"/>
```

```
<Label Text="Рис. 1.3. Командна панель"
```

```
HorizontalOptions="CenterAndExpand"
```

```
FontAttributes="Italic"
```

```
FontSize="Default"
```

```
Padding="0,100,0,0"/>
```

<Label Text="Дуже часто всі параметри, розташовані на командній панелі, не поміщаються на екран. В такому випадку необхідно прокручувати вікно командної панелі."

```
HorizontalOptions="StartAndExpand"
```

```
FontAttributes="None"
```

```
FontSize="Default"/>
```

<Label Text="Командна панель містить шість вкладок: Create (Створення), Modify (Зміна), Hierarchy (Ієрархія), Motion (Рух), Display (Відображення) і Utilities (Службові програми). Найбільш часто використовуються вкладки Create (Створення) і Modify (Зміна). На початку освоєння 3ds Max ви будете найчастіше звертатися до двох вкладках: Create (Створення) і Modify (Зміна)."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

<Label Text="Вкладка Create (Створення) служить для створення основних (примітиви, криві і ін.) і допоміжних (джерела світла, віртуальні камери, об'ємні деформації та ін.) об'єктів сцени."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

<Label Text="Вкладка Modify (Зміна) дозволяє змінювати параметри будь-якого виділеного об'єкту сцени. Також з її допомогою до виділеного об'єкту можна призначити модифікатор - певну дію, що деформує об'єкт. Модифікатор містить свої налаштування, які можна перетворювати за допомогою вкладки Modify (Зміна)."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

<Label Text="У нижній частині вікна 3ds Max розташована шкала анімації, під нею - координати перетворень, рядок стану, а також кнопки управління анімацією і виглядом у вікнах проекцій. Щоб дізнатися призначення кнопки на будь-якій панелі інструментів, досить підвести до неї курсор миші. При цьому біля кнопки виникне підказка, зміст якої також відобразиться в рядку стану."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

<Label Text="Вивчаючи інтерфейс 3ds Max, ви випадково можете змінити розташування панелей інструментів, приховати деякі елементи інтерфейсу і т.д. Якщо ви не знаєте, як привести програму в первинний вигляд, виконайте команду Customize> Load Custom UI Scheme (Налаштування> Завантажити схему інтерфейсу). У вікні вкажіть шлях до файлу DefaultUI (за замовчуванням це C: \ Program Files \ Autodesk \ 3ds Max\ ui.) і натисніть кнопку Відкрити. В результаті буде завантажена схема інтерфейсу, яка використовується за замовчуванням."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

<Label Text="1.2 Управління видом у вікнах проекцій"

```
HorizontalOptions="CenterAndExpand"
FontAttributes="Bold"
```

```

        FontSize="Title"/>
        <Label Text="Для управління видом з вікна проєкції використовуються
кнопки, які знаходяться в правому нижньому кутку вікна програми (рис. 1.4.)."
        HorizontalOptions="StartAndExpand"
        FontAttributes="None"
        FontSize="Default"
        Padding="0,0,0,50"/>
        <Image                                                    Scale="4"
Source="https://mix.sumdu.edu.ua/textbooks/30251/481361/file-assets/2.4.png"/>
        <Label Text="Рис. 1.4. Кнопки управління виглядом у вікнах проєкцій"
        HorizontalOptions="CenterAndExpand"
        FontAttributes="Italic"
        FontSize="Default"
        Padding="0,50,0,0"/>
        <Label Text="Розглянемо ці кнопки."
        HorizontalOptions="StartAndExpand"
        FontAttributes="None"
        FontSize="Default"/>
        <Label Text="Zoom (Масштаб) - наближення / видалення сцени."
        HorizontalOptions="StartAndExpand"
        FontAttributes="None"
        FontSize="Default"/>
        <Label Text="Zoom All (Масштаб всього) - наближення / видалення відразу
всіх об'єктів у всіх вікнах проєкцій."
        HorizontalOptions="StartAndExpand"
        FontAttributes="None"
        FontSize="Default"/>
        <Label Text="Zoom Extents / Zoom Extents Selected (Масштаб у межах/ Масштаб
виділеного) - наближення / видалення вибраного об'єкту / всіх об'єктів в межах видимості
поточного вікна проєкції."
        HorizontalOptions="StartAndExpand"
        FontAttributes="None"
        FontSize="Default"/>
        <Label Text="Zoom Extents All / Zoom Extents Selected (Масштаб обраного
об'єкта / Масштаб всіх об'єктів) - наближення / видалення вибраного об'єкту / всіх об'єктів
сцени в межах видимості всіх вікон проєкції. Цю кнопку зручно використовувати в тих випадках,
коли потрібно подивитися на сцену з такої точки, щоб у вікні проєкції відображалися всі
об'єкти.."
        HorizontalOptions="StartAndExpand"
        FontAttributes="None"
        FontSize="Default"/>

```

```

<Label Text="Field-of-View / Region Zoom (Видове поле / Масштаб області)
- зміна всього поля зору / виділеного за допомогою миші."
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="Pan View / Walk Through (Панорама / Перейти) - Кнопка Pan
(Панорама) служить для переміщення зображення на екрані вручну. Кнопка Walk Through (Перейти)
активує режим переміщення по сцені від першої особи. При включенні цього режиму змінювати вид
з вікна проєкції можна за допомогою клавіш управління курсором. Режим Walk Through (Перейти)
можна застосовувати тільки, якщо активне вікно Perspective (Перспектива)."
```

```

HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="Arc Rotate / Arc Rotate Selected / Arc Rotate SubObject
(Обертання по дузі / Обертання вибраного по дузі / Обертання навколо підоб'єкта по дузі) -
обертання сцени навколо центру поля зору / навколо виділених об'єктів / навколо підоб'єкта."
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="Min / Max Toggle (Збільшення вікна проєкції до розмірів
екрану) - збільшення активного вікна проєкції до розмірів екрану"
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="Щоб працювати в 3ds Max найбільш ефективно, обов'язково
потрібно вміти орієнтуватися у вікнах проєкцій. Часто користувачі-початковці втрачають з поля
зору об'єкти, направляють камери не в ту сторону і т. д. В 3ds Max можна повернути колишній
вигляд у вікні проєкції. Для цього необхідно натиснути правою кнопкою миші в лівому верхньому
кутку активного вікна проєкції і в контекстному меню вибрати рядок Undo View (Повернути
вигляд). Ця команда також дуже зручна, якщо ви випадково зачепили рукою мишу і порушили вид з
вікна проєкції."
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="1.3 Навігація за допомогою навігаційного куба"
HorizontalOptions="CenterAndExpand"
FontAttributes="Bold"
FontSize="Title"/>
<Label Text="В 3ds Max є інструмент ViewCube (Навігаційний куб), що помітно
спрощує навігацію в тривимірних сценах і дозволяє швидко змінювати вид з вікна проєкції і
розглядати сцену кількох точок."

```



```

HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="Навігаційний куб представлений у вікні програми у вигляді
значка віртуального куба, який прив'язаний до простору 3ds Max (рис. 1.5). Куб розташований в
правому верхньому кутку вікон проекції."
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"
Padding="0,0,0,40"/>
<Image Source="https://mix.sumdu.edu.ua/textbooks/30251/481361/file-assets/3.jpg"/> Scale="1.5"
<Label Text="Рис. 1.5. Навігаційний куб "
HorizontalOptions="CenterAndExpand"
FontAttributes="Italic"
FontSize="Default"
Padding="0,40,0,0"/>
<Label Text="Обертання цього куба викликає поворот простору тривимірної
сцени щодо центру глобальної системи координат, яка використовується програмою для визначення
положення об'єктів у віртуальному просторі."
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="Кожна сторона куба звернена до певного виду проекції і містить
напис. Наприклад, напис Right (Праворуч) говорить про те, що ця сторона повернута в напрямку
проекції Right, тобто на вигляд справа. При натисканні на стороні, ребрі або вершині значка
ViewCube (Навігаційний куб), простір тривимірної сцени повертається так, щоб обраний елемент
був у фокусі."
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="Цікаво, що перемикання відбувається не миттєво, а
сповільнено. Крім цього, ViewCube (Навігаційний куб) являє собою своєрідний компас, який дає
можливість користувачу зорієнтуватися у віртуальному просторі, відображаючи умовний Південь-
Захід-Північ-Схід, а також напрямки проекцій."
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
<Label Text="Поруч із позначкою ViewCube (Навігаційний куб) є крихітний
значок у вигляді будиночка. Клацання по цьому значку повертає те положення виду вікна проекції
Perspective (Перспектива), яке використовується програмою за замовчуванням."

```

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

<Label Text="Клацанням правою кнопкою миші по ViewCube (Навігаційний куб) можна викликати невелике меню, за допомогою якого можна змінити 'домашній' вид перспективи, а також відкрити вікно налаштувань цієї опції."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

<Label Text="У вікні налаштувань навігаційного куба (рис. 1.6) міститься безліч параметрів. Якщо зняти прапорець Show The View Cube (Показати навігаційний куб), то він перестане відображатися. За допомогою перемикача In All Views / Only in Active View (У всіх видах / Тільки в активному вікні) можна включити відображення куба у всіх вікнах проекцій або ж тільки в поточному. За допомогою списку ViewCube Size (Розмір навігаційного куба) встановлюється розмір цього елемента навігації. Якщо роздільна здатність вашого екрану не дуже велика, то навігаційний куб напевно буде займати занадто велику частину вікна проекції. В цьому випадку ви можете встановити для нього розмір Small (Маленький) або навіть Tiny (Крихітний)."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"
Padding="0,0,0,100"/>
```

<Image Scale="1.6"
Source="https://mix.sumdu.edu.ua/textbooks/30251/481361/file-assets/2.6.png"/>

```
<Label Text="Рис. 1.6. Вікно налаштувань навігаційного куба"
HorizontalOptions="CenterAndExpand"
FontAttributes="Italic"
FontSize="Default"
Padding="0,100,0,0"/>
```

<Label Text="За допомогою параметра Inactive Opacity (Непрозорість при відсутності активності) можна управляти непрозорістю куба в той час, коли він неактивний. Ця можливість дуже зручна, оскільки при встановленні значення 0% навігаційний куб буде показуватися тільки при наведенні курсору, а значить, не буде захищувати місце у вікні проекції."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

```
<Label Text="1.4 Тимчасове збереження сцени"
HorizontalOptions="CenterAndExpand"
FontAttributes="Bold"
FontSize="Title"/>
```

<Label Text="Для тимчасового збереження поточної сцени в програмі 3ds max передбачена команда меню Edit> Hold (Правка ^ Записати). Після внесення змін до сцену тимчасово збережений стан сцени можна повернути за допомогою команди Edit> Fetch (Правка> Відновити). Однак після великої кількості важливих і складних змін рекомендується зберігати сцену не тимчасово, а постійно - в файл."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

<Label Text="Тимчасово можна зберігати тільки один стан сцени, а при повторному виборі команди меню Edit> Hold колишній збережений стан буде замінено новим, тобто поточним станом."

```
HorizontalOptions="StartAndExpand"
FontAttributes="None"
FontSize="Default"/>
```

```
</StackLayout>
```

```
</ScrollView>
```

```
</Grid>
```

```
</ContentPage.Content>
```

```
</ContentPage>
```

PageTheory1.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace For3dsMax.Views
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PageTheory1 : ContentPage
    {
        double PositionX = 0;
        public PageTheory1()
        {
            InitializeComponent();
        }

        protected override void OnAppearing()
        {
        }

        protected override void OnDisappearing()
        {
        }
    }
}
```

AppShell.xaml

```

<?xml version="1.0" encoding="UTF-8"?>
<Shell xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:local="clr-namespace:For3dsMax.Views"
  Title="For3dsMax"
  x:Class="For3dsMax.AppShell">

  <Shell.Resources>
    <ResourceDictionary>
      <Style x:Key="BaseStyle" TargetType="Element">
        <Setter Property="Shell.BackgroundColor" Value="{StaticResource Primary}" />
        <Setter Property="Shell.ForegroundColor" Value="White" />
        <Setter Property="Shell.TitleColor" Value="White" />
        <Setter Property="Shell.DisabledColor" Value="#B4FFFFFF" />
        <Setter Property="Shell.UnselectedColor" Value="#95FFFFFF" />
        <Setter Property="Shell.TabBarBackgroundColor" Value="{StaticResource
Primary}" />
        <Setter Property="Shell.TabBarForegroundColor" Value="White"/>
        <Setter Property="Shell.TabBarUnselectedColor" Value="#95FFFFFF"/>
        <Setter Property="Shell.TabBarTitleColor" Value="White"/>
      </Style>
      <Style TargetType="TabBar" BasedOn="{StaticResource BaseStyle}" />
      <Style TargetType="FlyoutItem" BasedOn="{StaticResource BaseStyle}" />

      <Style Class="FlyoutItemLabelStyle" TargetType="Label">
        <Setter Property="TextColor" Value="White"></Setter>
      </Style>
      <Style Class="FlyoutItemLayoutStyle" TargetType="Layout"
ApplyToDerivedTypes="True">
        <Setter Property="VisualStateManager.VisualStateGroups">
          <VisualStateGroupList>
            <VisualStateGroup x:Name="CommonStates">
              <VisualState x:Name="Normal">
                <VisualState.Setters>
                  <Setter Property="BackgroundColor" Value="{x:OnPlatform
UWP=Transparent, iOS=White}" />
                  <Setter TargetName="FlyoutItemLabel"
Property="Label.TextColor" Value="{StaticResource Primary}" />
                </VisualState.Setters>
              </VisualState>
              <VisualState x:Name="Selected">
                <VisualState.Setters>
                  <Setter Property="BackgroundColor"
Value="{StaticResource Primary}" />
                </VisualState.Setters>
              </VisualState>
            </VisualStateGroup>
          </VisualStateGroupList>
        </Setter>
      </Style>

      <Style Class="MenuItemLayoutStyle" TargetType="Layout"
ApplyToDerivedTypes="True">
        <Setter Property="VisualStateManager.VisualStateGroups">
          <VisualStateGroupList>
            <VisualStateGroup x:Name="CommonStates">
              <VisualState x:Name="Normal">
                <VisualState.Setters>

```

```

        <Setter TargetName="FlyoutItemLabel"
Property="Label.TextColor" Value="{StaticResource Primary}" />
        </VisualState.Setters>
    </VisualState>
</VisualStateGroup>
</VisualStateGroupList>
</Setter>
</Style>
</ResourceDictionary>
</Shell.Resources>

<FlyoutItem Title="Головна" Icon="icon_about.png">
    <ShellContent Route="AboutPage" ContentTemplate="{DataTemplate local:AboutPage}" />
</FlyoutItem>
<ToolbarItem Text="Теорія" Icon="icon_feed.png"></ToolbarItem>
<FlyoutItem Title="Інтерфейс програми">
    <ShellContent Route="PageTheory1" ContentTemplate="{DataTemplate local:PageTheory1}"
/>
</FlyoutItem>
<FlyoutItem Title="Об'єкти в 3ds Max">
    <ShellContent Route="PageTheory2" ContentTemplate="{DataTemplate local:PageTheory2}"
/>
</FlyoutItem>
<FlyoutItem Title="Робота з виділеними об'єктами">
    <ShellContent Route="PageTheory3" ContentTemplate="{DataTemplate local:PageTheory3}"
/>
</FlyoutItem>
<FlyoutItem Title="Модифікатори">
    <ShellContent Route="PageTheory4" ContentTemplate="{DataTemplate local:PageTheory4}"
/>
</FlyoutItem>
<FlyoutItem Title="Булеві операції">
    <ShellContent Route="PageTheory5" ContentTemplate="{DataTemplate local:PageTheory5}"
/>
</FlyoutItem>
<FlyoutItem Title="Слайни">
    <ShellContent Route="PageTheory6" ContentTemplate="{DataTemplate local:PageTheory6}"
/>
</FlyoutItem>
<FlyoutItem Title="Лофт-об'єкти">
    <ShellContent Route="PageTheory7" ContentTemplate="{DataTemplate local:PageTheory7}"
/>
</FlyoutItem>
<FlyoutItem Title="Полігональне моделювання">
    <ShellContent Route="PageTheory8" ContentTemplate="{DataTemplate local:PageTheory8}"
/>
</FlyoutItem>

<ToolbarItem Text="Практика" Icon="icon_feed.png"></ToolbarItem>
<FlyoutItem Title="Інтерфейс">
    <ShellContent Route="PagePractice1" ContentTemplate="{DataTemplate
local:PagePractice1}" />
</FlyoutItem>
<FlyoutItem Title="Модифікатори">
    <ShellContent Route="PagePractice2" ContentTemplate="{DataTemplate
local:PagePractice2}" />
</FlyoutItem>
<FlyoutItem Title="Складені об'єкти">
    <ShellContent Route="PagePractice3" ContentTemplate="{DataTemplate
local:PagePractice3}" />
</FlyoutItem>

```

```

    <FlyoutItem Title="Слайни">
        <ShellContent Route="PagePractice4" ContentTemplate="{DataTemplate
local:PagePractice4}" />
    </FlyoutItem>
    <FlyoutItem Title="Loft">
        <ShellContent Route="PagePractice5" ContentTemplate="{DataTemplate
local:PagePractice5}" />
    </FlyoutItem>
    <FlyoutItem Title="Полігони">
        <ShellContent Route="PagePractice6" ContentTemplate="{DataTemplate
local:PagePractice6}" />
    </FlyoutItem>
    <ToolbarItem Text="Тести" Icon="icon_feed.png"></ToolbarItem>
    <FlyoutItem Title="Гарячі клавіши">
        <ShellContent Route="PageTest2" ContentTemplate="{DataTemplate local:PageTest2}" />
    </FlyoutItem>
</Shell>

```

AppShell.xaml.cs

```

//using For3dsMax.ViewModels;
using For3dsMax.Views;
using System;
using System.Collections.Generic;
using Xamarin.Forms;

namespace For3dsMax
{
    public partial class AppShell : Xamarin.Forms.Shell
    {
        public AppShell()
        {
            InitializeComponent();
        }
    }
}

```