

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнес-технологій «УАБС»
Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему «РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДГОТОВКИ ДО
ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ»

Виконав студент 2 курсу, групи ЕК.м-61а
(номер курсу) (шифр групи)

Спеціальності 051 «Економіка («Економічна
кібернетика»)

Соляник К. В.
(прізвище, ініціали студента)

Керівник доц., к. ф.-м. н, Коломієць С. В.
(посада, науковий ступінь, прізвище, ініціали)

Суми – 2018 рік

РЕФЕРАТ

кваліфікаційної магістерської роботи на тему «РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ»

студента Соляника Костянтина Вікторовича
(прізвище, ім'я, по батькові)

Сучасний рівень розвитку інформаційних технологій відкриває перспективи використання користувачами принципово нових сервісів в освіті – «мобільного навчання», що надає можливість навчатися незалежно від місця і часу, забезпечуючи неперервність і максимальну гнучкість навчального процесу.

Мобільне навчання є новою освітньою парадигмою, на основі якої створюється інше навчальне середовище, яке робить сам процес навчання всеохоплюючим та мотивує до неперервної освіти і навчання протягом життя. Одним із завдань мобільного навчання є розробка та впровадження відповідних мобільних додатків.

Актуальність даної теми зумовлюється тим, що мобільні технології входять в життя молодих людей досить стрімко. Кожний школяр або студент має мобільний телефон, тому одним із завдань мобільного навчання є впровадження відповідних мобільних додатків, які допоможуть підвищити інтерес до навчання, сприятимуть підвищенню рівня освіти.

Мета кваліфікаційної магістерської роботи полягає в розробці мобільного додатку для підготовки до зовнішнього незалежного оцінювання.

Об'єктом дослідження є мобільне навчання, як найбільш сучасна, зручна, інтегративна форма навчання.

Предметом дослідження є мобільний додаток, як засіб підготовки до зовнішнього незалежного оцінювання.

Результатом роботи є розроблений та впроваджений мобільний додаток для платформи Android для підготовки до зовнішнього незалежного оцінювання

«ЗНО 2017 КОМПЛЕКС».

Додаток «ЗНО 2017 КОМПЛЕКС» містить теоретичні матеріали для підготовки до зовнішнього незалежного оцінювання з української мови, української літератури, історії України, географії, математики, англійської мови, біології, фізики та хімії; корисний функціонал: перекладач онлайн, новини ЗНО, пошук репетитора.

Успішне впровадження додатку підтверджується значним показником завантажень додатку, а також позитивними відгуками користувачів додатку.

Результати кваліфікаційної роботи було представлено на Міжнародній науковій конференції «Інноваційний потенціал соціально-економічних систем: виклики глобального світу» (Лісабон, Португалія, 22 грудня 2017р.)

Ключові слова: мобільне навчання, мобільний додаток, ЗНО, android, google play, Ionic Framework.

Основний зміст кваліфікаційної магістерської роботи викладено на 72 сторінках, зокрема список використаних джерел із 49 найменувань, розміщений на 6 сторінках. Робота містить 2 таблиці, 29 рисунків, а також 5 додатків, розміщених на 16 сторінках.

Рік виконання кваліфікаційної роботи – 2017 рік

Рік захисту роботи – 2018 рік

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнес-технологій «УАБС»
Кафедра економічної кібернетики

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.е.н., доцент
_____ О.В. Кузьменко
“ _ ” _____ 2017 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ
(спеціальність 051 «Економіка («Економічна кібернетика»))
студенту 2 курсу, групи ЕК.м-61а

Солянику Костянтину Вікторовичу

(прізвище, ім'я, по батькові студента)

1. Тема роботи Розробка мобільного додатку для підготовки до зовнішнього незалежного оцінювання

затверджена наказом по університету від «08» грудня 2017 року № 2748-III

2. Термін подання студентом закінченої роботи «12» січня 2018 року

3. Мета кваліфікаційної роботи розробка та впровадження мобільного додатку для підготовки до зовнішнього незалежного оцінювання

4. Об'єкт дослідження мобільне навчання як найбільш сучасна, зручна, інтегративна форма навчання

5. Предмет дослідження мобільний додаток як засіб підготовки до зовнішнього незалежного оцінювання

6. Кваліфікаційна робота виконується на матеріалах ТОВ «РЕЗОНАНС.НЕТ»

7. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети

Розділ 1 Технології мобільного навчання в системі освіти

13 листопада 2017 року

(назва – термін подання)

У розділі 1 Сутність мобільного навчання; напрями впровадження мобільного навчання в систему освіти; особливості розробки мобільних додатків

(зміст конкретних завдань до розділу, які повинен виконати студент)

Розділ 2 Аналіз методів розробки мобільних додатків

12 грудня 2017 року

(назва – термін подання)

У розділі 2 Аналіз технологій розробки мобільних додатків; вибір технології розробки мобільного додатку «ЗНО 2017 КОМПЛЕКС»; архітектура вирішення задачі та функціональна структура додатку «ЗНО 2017 КОМПЛЕКС»

(зміст конкретних завдань до розділу, які має виконати студент)

Розділ 3 Реалізація програмного продукту «ЗНО 2017 КОМПЛЕКС»

4 січня 2018 року

(назва – термін подання)

У розділі 3 Створення прототипу мобільного додатку; життєвий цикл розробки мобільного додатку до періоду впровадження та тестування; впровадження та тестування мобільного додатку

(зміст конкретних завдань до розділу, які повинен виконати студент)

8. Консультації з роботи:

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| 1 | | | |
| 2 | | | |
| 3 | | | |

9. Дата видачі завдання: «10» жовтня 2017 року

Керівник кваліфікаційної роботи _____
(підпис)

С.В. Коломієць

Завдання до виконання одержав _____
(підпис)

К.В. Соляник

ЗМІСТ

| | |
|--|----|
| ВСТУП | 7 |
| 1 ТЕХНОЛОГІЇ МОБІЛЬНОГО НАВЧАННЯ В СИСТЕМІ ОСВІТИ | 9 |
| 1.1 Сутність мобільного навчання. Впровадження мобільного навчання в систему освіти | 9 |
| 1.2 Особливості розробки мобільних додатків | 15 |
| 1.3 Використання мобільних додатків при підготовці до ЗНО | 18 |
| 2 АНАЛІЗ МЕТОДІВ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ | 20 |
| 2.1 Мобільні додатки та засоби їх розробки | 20 |
| 2.2 Аналіз технологій розробки мобільних додатків. Вибір технології для розробки додатка «ЗНО 2017 КОМПЛЕКС» | 27 |
| 2.3 Архітектура вирішення задачі та функціональна структура додатку «ЗНО 2017 КОМПЛЕКС» | 34 |
| 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ «ЗНО 2017 КОМПЛЕКС» | 42 |
| 3.1 Створення прототипу мобільного додатку | 42 |
| 3.2 Життєвий цикл розробки мобільного додатку до періоду впровадження та тестування | 47 |
| 3.3 Впровадження та тестування мобільного додатку | 61 |
| ВИСНОВОК | 67 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 68 |
| ДОДАТКИ | 73 |

ВСТУП

Стрімка еволюція технологій визначає майбутній розвиток системи освіти. Технологічність освіти стає все більш затребованою, оскільки накопичення знань і фактичного матеріалу не є першочерговим завданням сучасного освітнього процесу. Сьогодні основним пріоритетом розвитку будь-якої розвиненої країни та її стратегії є покращення якості освіти і, в першу чергу, ефективне запровадження інформаційних та комунікаційних технологій в освіту.

Сучасний рівень розвитку інформаційних технологій відкриває перспективи використання користувачами принципово нових сервісів в освіті – «мобільного навчання», що надає можливість навчатися в незалежно від місця і часу, забезпечуючи неперервність і максимальну гнучкість навчального процесу.

Мобільне навчання є новою освітньою парадигмою, на основі якої створюється інше навчальне середовище, яке робить сам процес навчання всеохоплюючим та мотивує до неперервної освіти і навчання протягом життя. Одним із завдань мобільного навчання є розробка та впровадження відповідних мобільних додатків.

Використання мобільних додатків дозволяє ефективно організувати процес навчання, мати безперервний доступ до навчальних матеріалів, надає можливість покращувати знання будь-де та будь-коли, що є вкрай важливим для учнів та студентів при підготовці до іспитів, зокрема при підготовці до зовнішнього незалежного оцінювання.

Актуальність даної теми зумовлюється тим, що мобільні технології входять в життя молодих людей досить стрімко. Кожний школяр або студент має мобільний телефон, тому одним із завдань мобільного навчання є впровадження відповідних мобільних додатків, що допоможуть підвищити інтерес до навчання, а також сприятимуть підвищенню рівня освіти.

Актуальність даної роботи обумовила вибір об'єкта, предмета і мети.

Мета дослідження – розробка та впровадження мобільного додатку для підготовки до зовнішнього незалежного оцінювання.

Об'єкт дослідження – мобільне навчання, як найбільш сучасна, зручна, інтегративна форма навчання.

Предмет дослідження – мобільний додаток як засіб при підготовці до зовнішнього незалежного оцінювання.

Мета обумовила вибір наступних завдань:

- дослідити сутність мобільного навчання та шляхи його впровадження в систему освіти;
- провести аналіз методів та технологій розробки мобільних додатків;
- обрати технологію розробки для створення мобільного додатку;
- сформулювати функціональну структуру мобільного додатку для підготовки до зовнішнього незалежного оцінювання;
- розробити програмний продукт та провести його тестування.

1 ТЕХНОЛОГІЇ МОБІЛЬНОГО НАВЧАННЯ В СИСТЕМІ ОСВІТИ

1.1 Сутність мобільного навчання. Впровадження мобільного навчання в систему освіти

Сучасні ритми використання та розширення функціональності мобільних електронних пристроїв змушують навчальну спільноту думати про їх застосування з освітньою метою. Останнім часом науковці все частіше говорять про виникнення нових технологій та методів навчання за допомогою мобільних пристроїв, тобто мобільне навчання.

Характерною рисою останнього десятиріччя є активне використання засобів мобільного зв'язку та різноманітних електронних пристроїв. Сучасний мобільний телефон має можливості, що не поступаються комп'ютерам початкового рівня, а в деяких випадках – і середньої потужності. В-першу чергу це відноситься до смартфонів та персональних комунікаторів. На початок 2008 року в світі нараховувалося більше 1,5 мільярди мобільних телефонів та комунікаторів – майже втричі більше, ніж комп'ютерів. Потужність більшості мобільних пристроїв перевищує потужності персональних комп'ютерів середини 90-х років, які й досі ефективно використовуються в процесі навчання.

Перші відгуки про мобільне навчанням відносяться до 90-х років минулого століття (за інформацією Рон Веттер). Ціла концепція M-Learning була запропонована Д. Кіганом у 2001 р. подальший розвиток вона дістала у роботах Ф. Манг'яваччі, Р. Мейсона, Л. Родіна, М. Рончетті, А. Трифонові та Д. Хойла (2002–2003 рр.). В дисертації Фенг-Хуан Ю Янга (2003 р.) запропонована архітектура розподіленої системи мобільного навчання. У 2001 році Єврокомісія заснувала проекти MOBLearn та M-Learning, у 2002 році в Канаді створено Консорціум мобільного навчання (The m-Learning Consortium),

а в Австралії – державний стандарт на мобільне навчання. У 2004 році корпорацією Intel було розпочато проект «Навчання завжди та всюди», метою якого є надання кожному з учнів персонального доступу до мобільних комп'ютерних пристроїв та забезпечення безпроводного зв'язку у школах [1].

В останні декілька років значного розвитку набувають теоретичні та практичні дослідження з питань використання мобільних пристроїв у навчальному процесі. Дана тенденція набуває актуальності поряд із започаткуванням «ери смартфонів і планшетів», адже саме дані пристрої символізують сучасне інформаційне суспільство. За даними дослідницької компанії GfK Ukraine за 2015 рік сучасний ринок мобільних пристроїв та смартфонів, у кількісних показниках збільшився на 68 %. Також простежується тенденція до збільшення діагоналі екрану «розумного пристрою»: частка мобільних смартфонів, розмір екрана від трьох дюймів, зростає з 74 % до 81 %, також від 0 до 13 % зростає частка 4-дюймових смартфонів [2]. Сучасний ринок планшетів також відзначився динамічним підйомом. Аналітики України обґрунтовують масове поширення планшетних комп'ютерів із великим списку домашніх завдань, які можна вирішити за їх допомогою, а саме пошук необхідної інформації в мережі Інтернет, читання різної літератури, спілкування в соціальних мережах тощо.

Різні оператори мобільного зв'язку пропонують кожному клієнту велику кількість можливостей щодо підключення до мережі Інтернет. Випуск фірмою Apple планшетів iPad зумовив виникнення за кордоном цілого напрямку щодо їх використання в навчальному процесі початкової, середньої та вищої школи, а також зумовив достатньо велику кількість розробок навчальних додатків для даних мобільних пристроїв

Практичні аспекти мобільного навчання, які можливо застосовувати в навчальному процесі, досліджувались багатьма авторами, зокрема І. М. Голіцина [3], Т. А. Калуга [4], В. О. Куклев [5], С. О. Семеріков [6], Ю.В. Триус [7] та інші.

В сучасній літературі пропонуються декілька визначень мобільного навчання, для яких спільним є те, що за цієї технології навчання фізичне з'єднання з кабельною мережею виявляється непотрібним.

Мобільне навчання (m-learning) – це передавання знань на мобільні пристрої з використанням WAP і GPRS технологій [8, с. 233].

Мобільне навчання – електронне навчання за допомогою мобільних засобів, незалежно від часу та місця, з використанням спеціального програмного забезпечення на педагогічній основі міждисциплінарного та модульного підходів [5].

Мобільне навчання – підхід до навчання, при якому на основі мобільних електронних пристроїв створюється мобільне освітнє середовище, де студенти можуть використовувати їх у якості засобу доступу до навчальних матеріалів, що містяться в Інтернеті, будь-де та будь-коли [6].

Мобільне навчання – це діяльність, здійснювана регулярно допомогою компактних, портативних мобільних пристроїв і технологій та дозволяє учням курсу стати більш продуктивними, спілкуючись, отримуючи або створюючи інформацію [9].

Аналіз літературних джерел показує, що мобільне навчання є, з одного боку, різновидом дистанційного навчання, а з іншого – електронного навчання. Але у порівнянні з цими видами навчання, мобільне навчання надає студенту або учню вищу інтерактивність, більшу свободу руху, більшу кількість технічних засобів.

Моделі традиційного, електронного і мобільного навчання представлені на рисунку 1.2 [6]. Аналіз наведених моделей освіти показує, що найбільш динамічною та активною по відношенню до основного суб'єкта освіти є модель мобільного навчання.

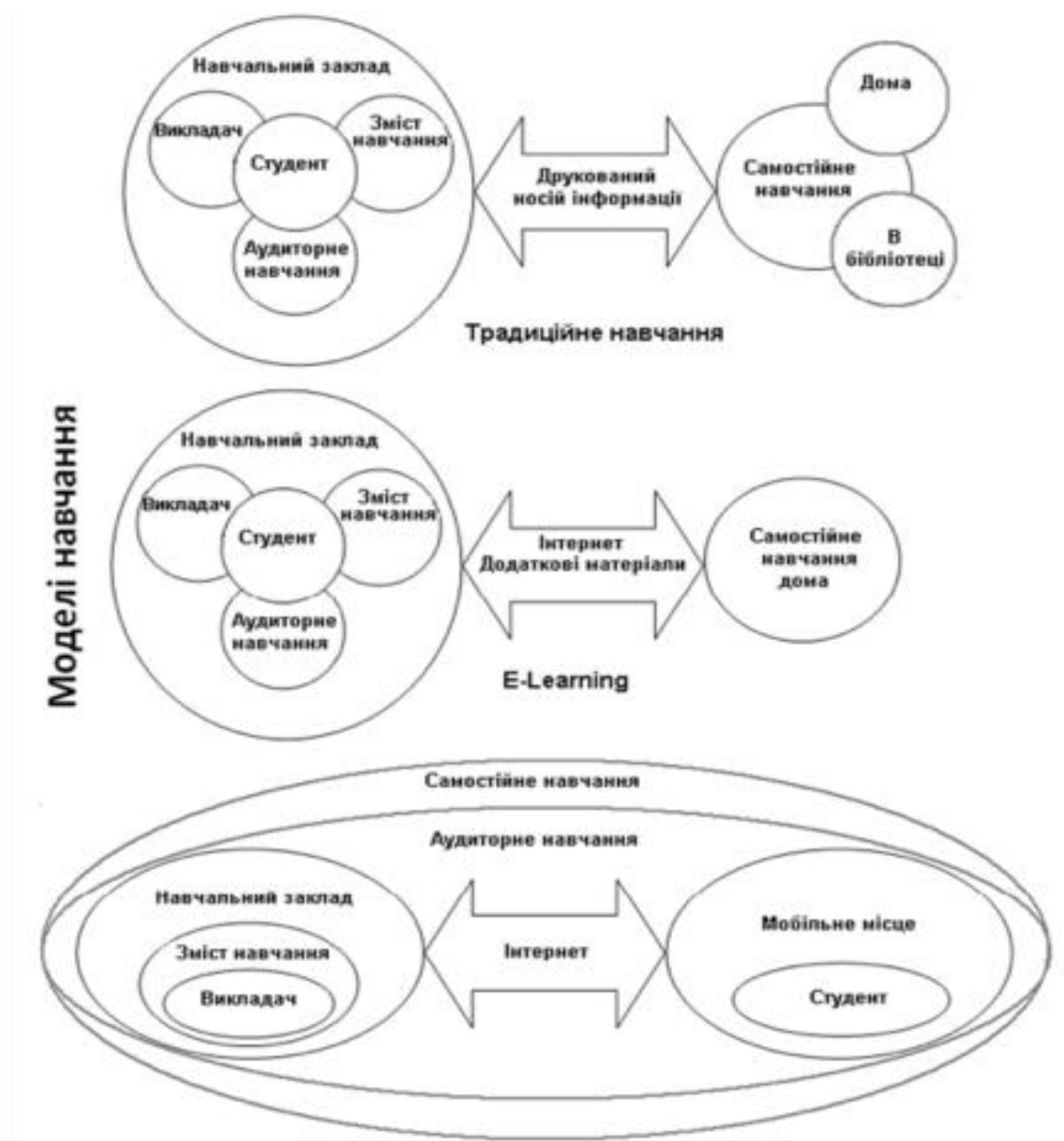


Рисунок 1.2 Моделі традиційного, електронного та мобільного навчання

Аналізуючи особливості мобільного навчання, дослідники виділяють наступні аспекти [9]:

- студенти та учні готові використовувати мобільні пристрої для навчання в тих випадках, коли вони не можуть скористатися книгою чи комп'ютером;

- мобільне навчання надає можливість студентам та учням використовувати вільні проміжків часу;

- мобільне навчання надає можливість здійснювати спільну онлайн роботу над проектом, мобільний блогінг, персоналізоване навчання, роботу у групах, онлайнві дослідження, рівний доступ до навчання;

- мобільні додатки повинні бути компактними й активізуватися з того місця, на якому була перервана робота;

- мобільні додатки повинні бути доступними в мережі Інтернет, а також бути синхронізованими з мобільними засобами навчання.

Основними властивостями мобільного навчання є:

- придатність до одночасної взаємодії як з одним студентом, так і з групою студентів або учнів;

- можливість динамічного генерування навчального матеріалу в залежності від місця знаходження студентів, контексту навчання та способу використання мобільних пристроїв;

- можливість виконання окремих дискретних у часі навчальних дій студентів та учнів у будь-який час і в будь-якому місці;

- можливість реалізації змішаного навчання.

Порівняльний аналіз електронного та мобільного навчання за різними показниками наведений у таблиці 1.1 [10].

Серед основних переваг запровадження m-learning у системі освіти можна відмітити наступні:

- взаємодія студента та викладача у зручній для обох час;

- застосування однакових прийомів навчальної діяльності в аудиторії та поза нею;

- створення можливості безперервного обміну інформацією між усіма учасниками навчального процесу;

- можливість навчатися будь-де і будь-коли;

- підвищення інтересу до навчання навіть серед тих учнів, що втратили інтерес до навчання, через використання мобільних телефонів, гаджетів тощо.

Таблиця 1.1 – Порівняльний аналіз електронного та мобільного навчання

| Електронне навчання | Мобільне навчання |
|---|---|
| Навчальний процес | |
| -навчальні матеріали: текстові та графічні в комп'ютерному або Інтернет-класі на ПК | -навчальні матеріали: текстові та графічні та голосові будь-де та будь-коли |
| Комунікація між учнями | |
| -безпосередня | -опосередкована |
| -через e-mail | -через e-mail, SMS, MMS |
| Обернений зв'язок з учнями | |
| -опосередкований через e-mail, Web-сайти (форуми, чати тощо) | -прямий через мобільні пристрої |
| -модельні та лабораторні експерименти (на практичних заняттях) | -прикладні з реального життя |
| Оцінювання та контроль знань | |
| -в класі | -в будь-якому місці |
| -обмежений в часі | -без обмежень в часі |
| Подання навчального матеріалу та іспити | |
| -теоретично-орієнтовані текстові іспити | -практично-орієнтовані іспити |
| -спостереження та дослідження процесів та явищ в лабораторії | -спостереження процесів та явищ на відстані |
| -застосування однієї мови | -автоматичний переклад та подання матеріалу різними мовами |

За допомогою використання мобільних технологій існує можливість автоматизувати процес навчання шляхом перерозподілу навчального часу, модернізації навчально-виховного процесу, введення нових підходів до подання теоретичного матеріалу, наприклад за допомогою відео-уроків, мобільних додатків, електронних бібліотек зі всіма потрібними джерелами

інформації у вільному доступі.

Таким чином, серед основних переваг m-learning можна виділити наступні [11]:

- m-learning через мобільний пристрій робить навчання справді індивідуальним. Учні мають можливість вибору змісту навчання з урахуванням власних інтересів, унаслідок чого m-learning є орієнтованим на студента або учня;

- гнучкість, негайний доступ до інформації, необхідної для конкретної роботи, за допомогою мобільних пристроїв дає змогу підвищити продуктивність праці людини;

- самостійне навчання і негайне надання контенту за запитом є характерними рисами m-learning. Воно надає користувачам можливість пройти навчання в неробочий час і створює умови для спільного навчання та взаємодії.

Сам процес навчання кожного студента має стандартний набір для навчальної роботи, а саме [12]:

- самостійна робота з електронними ресурсами навчальної організації, які адаптовані до роботи на мобільному пристрої;

- самостійна робота з іншими електронними посібниками, посилення на які вказані в електронних навчальних підручниках;

- робота з груповими та самостійними проектами;

- виконання тестування для самоконтроля;

- консультування з викладачем тощо.

1.2 Особливості розробки мобільних додатків

Актуальним питанням впровадження мобільного навчання є розробка відповідних інформаційних ресурсів, що має свої особливості [13]:

- мобільні пристрої мають невеликий розмір екрану, маленьку

клавіатуру та обмежені пропускну здатністю для мультимедійних файлів. Необхідно враховувати, щоб дизайн мобільної інформації враховував ці особливості, а не просто повторював існуючу структуру звичайного контенту. Наприклад, актуальною є розробка мобільних версій наукових, навчальних сайтів, які будуть забезпечувати швидкий доступ до них, а також будуть використовувати оптимальні параметри саме для роботи з інформацією за допомогою мобільних пристроїв;

- при розробці мультимедіа, необхідно враховувати формати для фото, аудіо та відео файлів, що підтримуються мобільними пристроями. Доречним є створення мобільних матеріалів саме у форматах: PNG, GIF, 3GP, MP3, MP4;

- найбільш ефективною є розробка навчальних програм, розрахованих на 10-15 хвилин навчання. Зокрема, короткі модулі навчальних ресурсів академії Хана (Khan Academy), що виконують вказані рекомендації, оскільки мобільний пристрій використовується саме у перервах між основними видами діяльності, і саме через це мобільне навчання має займати короткі проміжки часу;

- розробка навчальних матеріалів повинна враховувати анатомічні особливості кожного учня (товсті чи худі пальці), також якість зору учня. Саме тому, мобільні навчальні матеріали повинні мати крупний чіткий шрифт, зрозумілі зображення тощо;

- потрібно не перевантажувати мобільні навчальні матеріали великими мультимедіа, адже саме вони впливають на швидкість завантаження даної інформації;

- потрібно постійно слідкувати за оновленням навчальних матеріалів.

Для розробки електронних курсів існує безліч додатків (більшість з яких є веб-додатками), які доступні за ліцензією типу open source.

До першої групи належать системи управління навчальним контентом (англ. Learning content management system, LCMS): Microsoft Learning Content Development System, Moodle та інші. Найбільш часто дані LCMS вбудовані в систему управління навчанням (англ. LMS, Learning Management Systems) на базі SCORM – стандартів (англ. Sharable Content Object Reference Model) для

програмних продуктів електронного навчання, що передбачають обмін навчальними матеріалами на основі XML через web-інтерфейс, серед яких найбільш визнаними є:

- Moodle (<https://moodle.org/>);
- Microsoft SharePoint (<http://sharepointlms.com>);
- Google App for education (<http://www.elearninglearning.com/google/lms>);
- WebTutor (<http://www.mylms.ru>).

До другої групи додатків належать платформи зі створення інтерактивних книг і підручників:

- iBooksAuthor (<http://www.apple.com/ibooks-author/>);
- додаток для розробки електронної книги з розміщенням в хмарному сервісі AppStore;
- CourseBuilder від Google (<http://code.google.com/p/coursebuilder/>);
- додаток для створення електронних підручників з подальшим розміщенням в GoogleeBookstore;
- Magic Info-IPremium Author від Samsung (<http://magicinfoshop.com/ru/soft>)
- додаток для створення ефектних презентацій та інтерактивного вмісту.

Засоби мобільних інформаційно-комунікаційних технологій навчання можна поділити на апаратні та програмні. Існують різні програмні мобільні засоби навчання такі як [14]:

- MobileELDIT;
- MLEX;
- MLE-Moodle;
- AmadeusLMSMobile;
- LearnCast;
- Mobl21 та інші.

Сучасні апаратні пристрої на які встановлені різні операційні системи такі як Android, BlackBerry, iPhone, WindowsCE, безкоштовно підтримують різні мовні функціональні можливості без додаткового програмного

забезпечення.

Функціональні можливості – різні сервіси, які пропонують операційній системі і самі пристрої своїм користувачам. У таких систем спільним є те, що на їх платформи можна встановити безкоштовні додатки браузер GoogleChrome.

Інтернет браузер GoogleChrome, встановлений на платформі ОС Android, має доступ до додатків на 30 мовах і в ньому присутні різні функції, а саме користувач може запитати настільну версію web-сайту, якщо він відмовляється переглядати мобільну версію сайту, також можна додавати закладки що значно спрощує пошук улюблених сайтів.

Оскільки Інтернет-ресурси та комп'ютери стали невід'ємними освітніми інструментами і з'являються більш прості та ефективні пристрої, відкриваються широкі можливості для розширення сфер використання. Варто відзначити, що мобільні пристрої коштують дешевше ніж настільні ПК і є більш дешевим засобом доступу до Інтернет ресурсів. Тому особливої актуальності набуває пошук нових підходів до організації навчального процесу і створення навчальних матеріалів і технологій, які б враховували можливості мобільних пристроїв. Важливим є створення відповідного середовища мобільного навчання, що потребує розробки відповідних додатків.

1.3 Використання мобільних додатків при підготовці до ЗНО

Одним із основних елементів забезпечення якості освіти є зовнішнє незалежне оцінювання (ЗНО) навчальних досягнень учнів. Зовнішнє незалежне оцінювання – одна з найпоширеніших у світі та ефективних систем оцінювання навчальних досягнень учнів, яка дозволяє провести як підсумкову атестацію, так і селекцію для вищих навчальних закладів. Результати зовнішнього незалежного оцінювання якості освіти є основою для вироблення освітньої

стратегії і тактики, оцінювання роботи закладів освіти зокрема й освітніх систем в цілому.

На даний час в Україні існує багато онлайн курсів з підготовки до зовнішнього незалежного оцінювання, пропонуються заняття онлайн з різних предметів, свої послуги пропонують відкриті освітні ресурси, зокрема <https://prometheus.org.ua/zno/> тощо.

Протягом останніх років показник, що характеризує рівень попиту на мобільні пристрої, зокрема на освітні мобільні пристрої, значно зростає. За даними сайту <http://osvita.mediasapiens.ua/> ринок мобільних додатків очікує значне зростання – на 270% до 2020 року. За оцінками компанії AppAnnie, ринок мобільних додатків очікує значне зростання – з \$ 70 млрд в 2015 році до \$ 189 млрд в 2020. Користувачі проводитимуть в мобільних додатках на 114% більше часу до 2020 року, а мобільні ігри буду приносити 55% всіх доходів. Зростає число встановлення додатків на смартфонах і планшетах – цей показник очікується на рівні 6,2 млрд до 2020. Враховуючи загальносвітову тенденцію використання мобільних додатків, а також всі переваги мобільного навчання, можна зробити висновок, що використання мобільних додатків при підготовці до ЗНО є дуже доцільним та дозволить учням більш ефективно підготуватися до зовнішнього незалежного оцінювання.

Отже, розробка мобільних додатків з підготовки до зовнішнього незалежного оцінювання є актуальною і доцільною, оскільки надасть можливість сумістити розваги і навчання. Але тільки корисна розробка отримає гідне визнання з боку користувачів. З метою створення якісного продукту необхідно провести аналіз методів та технологій розробки мобільних додатків та обрати відповідну технологію розробки для створення мобільного додатку.

2 АНАЛІЗ МЕТОДІВ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

2.1 Мобільні додатки та засоби їх розробки

Мобільний додаток – це програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Багато мобільних додатків вже встановлені на самому пристрої або можуть бути завантажені з онлайн-магазинів, зокрема:

- App Store;
- BlackBerry;
- App World;
- Google play;
- Windows Store.

Раніше мобільні додатки використовувалися для швидкої перевірки електронної пошти, але їх високий попит призвів до розширення їхніх призначень і в інших областях, таких як навчання, ігри для мобільних телефонів, спілкування, перегляд відео та користування інтернетом.

Android – операційна система та платформа для мобільних телефонів і планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення та застосування спираються на цю реалізацію Java [15].

iOS – це власницька мобільна операційна система від Apple. Розроблена спочатку для iPhone, вона стала операційною системою також для iPod Touch, iPad і Apple TV. Apple не дозволяє роботу ОС на мобільних телефонах інших фірм. iOS є похідною від OS X, отже, є за своєю природою Unix-подібною операційною системою. Користувацький інтерфейс iOS заснований на концепції прямої маніпуляції з використанням жестів Multi-Touch. Елементи

інтерфейсу управління складаються з повзунків, перемикачів і кнопок. Він призначений для безпосереднього контакту користувача з екраном пристрою. Внутрішній акселерометр використовуються деякими програмами для реагування на струшування пристрою, яке є також загальною командою скасування, або обертання пристрою у трьох вимірах, що є загальною командою перемикання між книжковим та альбомним режимами.

Windows Phone – мобільна операційна система, розроблена Microsoft, яка почала працювати 11 жовтня 2010 року . 21 жовтня 2010 року почалися поставки перших пристроїв на базі нової платформи. 9 Жовтня 2017 року виконавчий директор Microsoft Джо Бельфіор заявив про припинення створення нових пристроїв і оновлень Windows 10 Mobile.

На рисунку 2.1 представлені найбільш популярні мобільні операційні системи, роки впровадження, та OS – сімейства до яких їх відносять [16].

| |  |  |  |
|--------------|---|--|---|
| Developer | Google | Apple | Microsoft |
| Released | 2008 | 2007 | 2010 |
| Source model | Open | Closed | Closed |
| OS family | • Unix-like | • Unix-like • Darwin (BSD) • OS X | • Microsoft Windows |

Рисунок 2.1 Найпопулярніші мобільні операційні системи

На рисунку 2.2 представлено порівняння кількості завантажень мобільних додатків із Google play та App Store. Отже, можемо зробити висновок, що найбільша частка завантажень мобільних додатків припадає на Google play і

має постійну тенденцію до зростання [16].

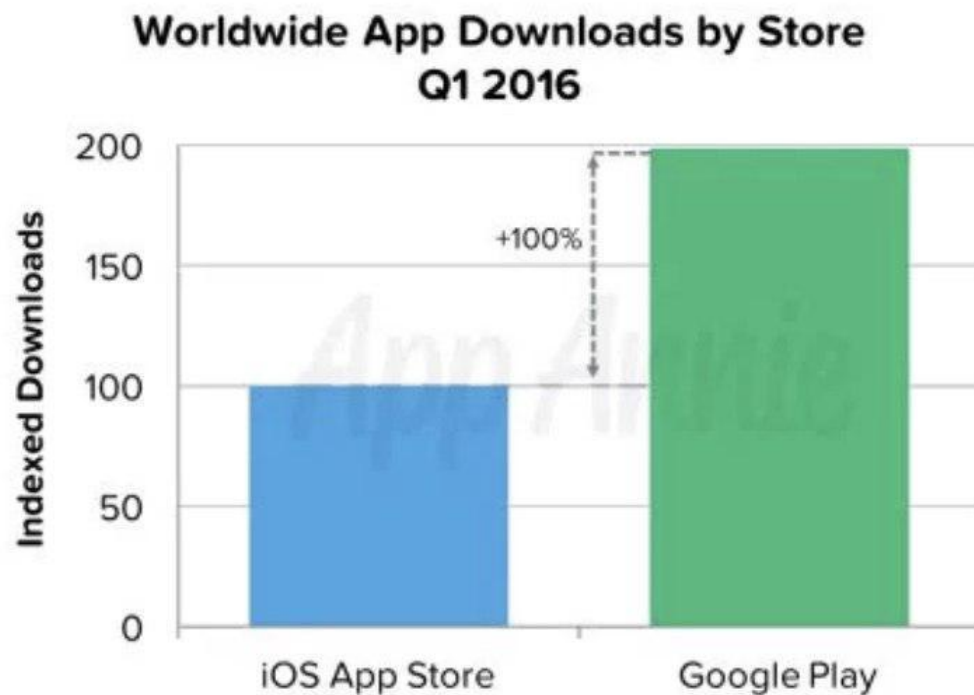


Рисунок 2.2 Порівняння завантажень мобільних додатків із найпопулярніших магазинів за 2016 рік

Для розробки додатків на базі операційної системи Android може використовуватися платформа Eclipse, яка доповнюється різними модулями. На основі Eclipse Platform створюються різні Eclipse IDE (IDE, Integrated development environment, або integrated debugging environment, - середовище розробки) для написання ПО на безлічі мов, наприклад на найбільш популярній мові програмування Java [17].

Одним з таких модулів є Android Development Tools (ADT) - плагін для Eclipse IDE, призначений для розробки Android-додатків. ADT розширює можливості Eclipse і дозволяє швидко створювати нові проекти для пристроїв на основі Android. Поряд з усім іншим створюється користувацький інтерфейс додатка, додаються компоненти, засновані на Android Framework API, проводиться налагодження додатків за допомогою інструментів Android

SDK. Цей комплект засобів розробки (Software Development Kit, SDK) дозволяє створювати додатки для певного пакету програм. На рисунку 2.3 представлена ієрархія класів Android SDK [17].

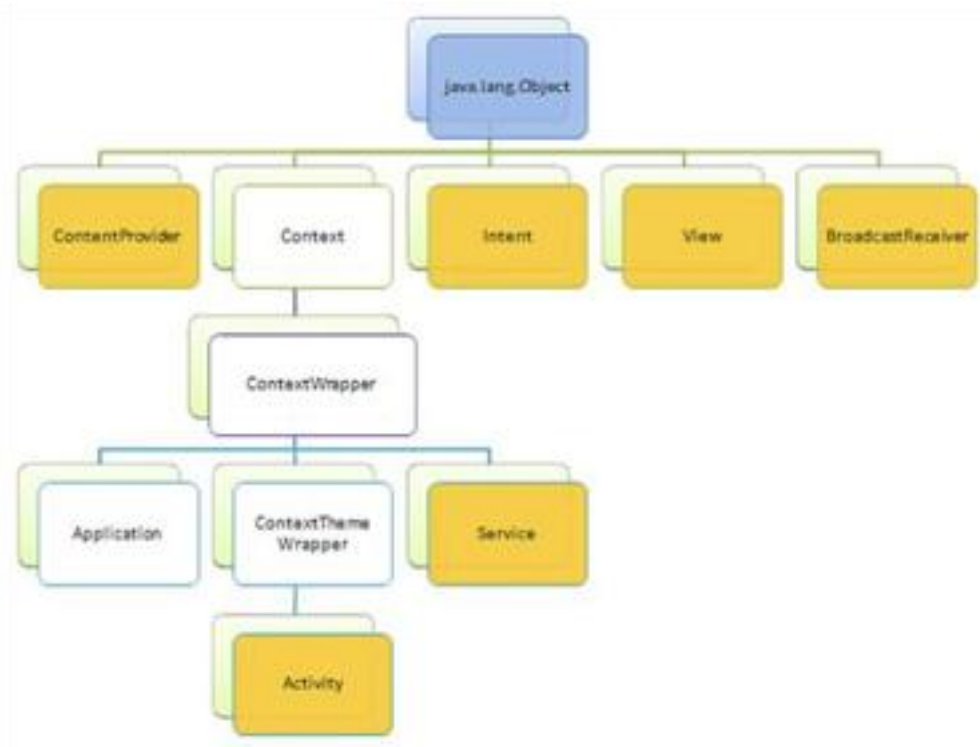


Рисунок 2.3 Ієрархія класів Android SDK

Темним кольором виділені класи, з якими розробник працює безпосередньо та успадковує від них свої класи. Інші класи не менш важливі, але вони використовуються рідше.

Середовище розробки Android Studio.

Android Studio є офіційним IDE для розробки додатків Android на основі IntelliJ IDEA.

Студія пропонує:

- гнучку Gradle-систему збірки;
- побудувати варіанти і кілька APK покоління файлу;
- шаблони коду, щоб допомогти побудувати загальні риси додатку;
- багатий редактор макетів з підтримкою перетягування і падіння редагування теми;

- інструменти, щоб фіксувати продуктивність, зручність використання, сумісність версії, і інші проблеми.

Sample Code – Android SDK надає типові додатки, які демонструють деякі з можливостей Android, прості програми, які показують, як використовувати індивідуальні особливості API у програмному коді. Перед початком розробки мобільного додатку потрібно забезпечити сумісність з пристроями (Android API Level), на які воно буде встановлюватися [18]. Рівень API – цілочисельне значення, яке однозначно визначає версію API платформи Android. Платформа забезпечує структуру API, які додатки можуть використовувати для взаємодії з системою Android. Кожна наступна версія платформи Android може містити оновлення API. Оновлення API-структури розроблені так, щоб новий API залишався сумісним з попередніми версіями API. Таким чином, більшість змін в API є сукупною і вводить нові функціональні можливості або виправляє попередні. Оскільки частина API постійно оновлюється, застарілі API не рекомендуються до використання, але не видаляються з міркувань сумісності з наявними додатками. Рівень API, який використовує додаток для Android, визначається цілочисловим ідентифікатором, який вказується у файлі конфігурації кожного Android-дodatku. У таблиці 2.1 представлені рівні API і версії Android [19].

Для платформи iOS використовується мова ObjectiveC та Swift. Swift – нова мова програмування, представлена компанією Apple в 2014 році. Вона успадкувала більшість характеристик від ObjectiveC.

Рішення Oracle Application Development Framework (ADF) Mobile, яке з'явилося в кінці 2011 року дозволяє розробникам швидко створювати додатки для багатьох мобільних платформ. Бізнес-додатки, розроблені за допомогою інструменту ADF Mobile, надають великі можливості для мобільних пристроїв, оптимізують користувацького інтерфейсу з використанням HTML5 та інтеграцію сервісів. Рішення ADF Mobile застосовується для того, щоб включити в бізнес-додатки Oracle Application підтримку мобільних пристроїв [19].

Таблиця 2.1 - Рівні API та версії платформи Android

| Code name | Version number | Initial release date | API level |
|--------------------|-----------------------|----------------------|-----------|
| | 1.0 | September 23,2008 | 1 |
| | 1.1 | February 9,2009 | 2 |
| Cupcake | 1.5 | April 27,2009 | 3 |
| Donut | 1.6 | September 15,2009 | 4 |
| Éclair | 2.0-2.1 | October 26,2009 | 5-7 |
| Froyo | 2.2-2.2.3 | May 20,2010 | 8 |
| Gingerbread | 2.3-2.3.7 | December 6,2010 | 9-10 |
| Honeycomb | 3.0-3.2.6 | February 22,2011 | 11-13 |
| Ice Cream Sandwich | 4.0-4.0.4 | October 18,2011 | 14-15 |
| Jeity Bean | 4.1-4.3.1 | July 9,2012 | 16-18 |
| KitKat | 4.4-4.4.4,4.4W-4.4W.2 | October 31,2013 | 19-20 |
| Lollipop | 5.0-5.1.1 | November 12,2014 | 21-22 |
| Marshmallow | 6.0-6.0.1 | October 5,2015 | 23 |
| N | Developer Preview 3 | | |

Також до основних засобів розробки мобільних додатків відносяться UI/UX Design. UI/UX-дизайн – в перекладі означає «досвід взаємодії» і включає в себе різні UX-компоненти:

- інформаційну архітектуру;
- проектування взаємодії;
- графічний дизайн;
- контент.

В цілому, UX дизайн – комплексний підхід до взаємодії користувача з

інтерфейсом, це може бути веб-додаток, мобільний додаток або будь-яка інша програма.

User Interface Design або користувальницький інтерфейс – більш широке поняття, яке включає в себе певний набір графічно оформлених технічних елементів (кнопки, чекбокси, селектори та інші поля). Його основними завданнями є:

- допомогти користувачеві організувати взаємодію з програмою (на сьогоднішній момент існують деякі правила UI дизайну);
- організування елементів інтерфейсу (вони повинні бути логічно структуровані і взаємопов'язані);
- угруповання елементів інтерфейсу (об'єднання в групи логічно пов'язаних елементів (меню, форми));
- вирівнювання елементів інтерфейсу;
- єдиний стиль елементів інтерфейсу (стильове оформлення зберігається в пам'яті користувача);
- наявність вільного простору (дозволить розмежовувати інформаційні блоки, зосереджуючи увагу на основному).

Розроблений за всіма правилами користувальницький інтерфейс значно підвищує ефективність ресурсу і дає йому конкурентні переваги.

Іноді пропонують дизайн-рішення на основі шаблонів, що прискорює роботу. Кінцевим продуктом роботи є дизайн-макет: картинка, що представляє передбачуваний майбутній зовнішній вигляд сторінок додатку, розміром приблизно 960x640 пікселів – розмір, відповідний середньому стандарту, пов'язаний з необхідністю подальшої прив'язки до різних дозволами екрану монітора. Картинка є багат шаровою, де майже кожна деталь – окремий шар, прикладений до інших верств-картинок, за рахунок чого може легко виконуватися доробка, заміна, перекомпонування або інші завдання. Залежно від ідеї і цілей макет може включати фотографії, складні колажі, ілюстрації, текстові шари, унікальні іконки. Для головної сторінки і внутрішніх іноді малюються окремі макети з доповненнями або змінами відповідно до тематики

сторінки.

Зображення спочатку може бути векторним або растровим, виконаним в Adobe Illustrator, Adobe Photoshop, GIMP або іншому візуальному редакторі (наприклад, Scribus або Inkscape), але для верстальника зображення, як правило, переводиться в растровий формат [20].

2.2 Аналіз технологій розробки мобільних додатків. Вибір технології для розробки додатка «ЗНО 2017 КОМПЛЕКС»

На сьогодні, мобільні додатки надають можливість поєднувати розваги та навчання, навіть існують додатки для наймолодшої вікової категорії. Найбільша складність у створенні таких додатків – розробка педагогічної складової. Важливо, не лише якісно намалювати ілюстрації та правильно запрограмувати додаток, потрібно також визначити, які саме здібності буде розвивати створена програма та чого вона зможе навчити користувача [21].

Вибір засобів розробки є одним з важливих етапів при створенні програмного мобільного додатку. Для реалізації мобільного додатку «ЗНО 2017 КОМПЛЕКС» на платформі Android та IOS, потрібно знайти оптимальне рішення, яке допомогло б скоротити час на розробку додатку та його тестування, одразу на дві різні платформи. Тому за технологію було обрано крос-платформенне рішення. Порівнюємо поняття крос-платформенних додатків та розробку за допомогою нативних додатків.

Нативні додатки.

Нативний додаток (при розробці на платформу Apple iOS) – це рішення, розроблене на мові програмування Objective-c або Swift (мова програмування Apple для розробки додатків під iOS і OS X, представлений на WWDC 2014 року), в рамках інтегрованого середовища розробки Xcode. З точки зору користувача, нативний додаток нічим не відрізняється від якісно виконаного

крос-платформенного додатку, адже нативні додатки створені під конкретну операційну систему мають зрозумілий користувачеві (нативний) інтерфейс. Зручність полягає в тому, що при розробці нативних додатків зв'язка UI / UX фахівця і програміста найбільш ефективна, адже дизайнер точно знає, які UI підходи звичні користувачам конкретної операційної системи (наприклад кнопка «Back» і нижній Tab Bar в iOS), а iOS розробник точно знає, як реалізувати ті чи інші UI фішки саме для мобільних пристроїв виробництва Apple. Саме завдяки цьому, користувач відкриває нативний додаток і інтуїтивно взаємодіє з новим інтерфейсом [22].

Крос-платформенні мобільні додатки.

Багато-платформність – програмне забезпечення, яке працює більш ніж на одній апаратній платформі або операційній системі. Крос-платформенна розробка – використання спеціальних утиліт (фреймворків) для створення програми на основі сімейства мов JavaScript. Вся структура і логіка додатка створюється за допомогою таких інструментів як PhoneGap, Titanium, Xamarin, Cordova на JavaScript, а потім обертається в нативний запускаючий елемент, тобто інтегрується в базовий проект для XCode або Android Studio, що дозволяє створювати збірки проекту з однієї і тієї ж логікою під кілька операційних систем відразу. Отже, крос-платформенний додаток - рішення розроблене таким чином, щоб мати можливість, з мінімальними зусиллями, скомпілювати вихідний код для виконання на кількох мобільних платформах, але результатом кожної окремої компіляції будуть окремі виконувані файли. Наприклад, під iOS виконуваний файл має розширення – «*.ipa», під Android – «*.apk» і т.д.

Гібридні додатки.

Гібридні технології досить популярні серед розробників крос-платформенних додатків. В більшій мірі, це обумовлюється тим, що з функцією веб-браузера може впоратися практично будь-яка мобільна операційна система, саме тому запустити її на будь-якій іншій не складе труднощів (рисунок 2.4).

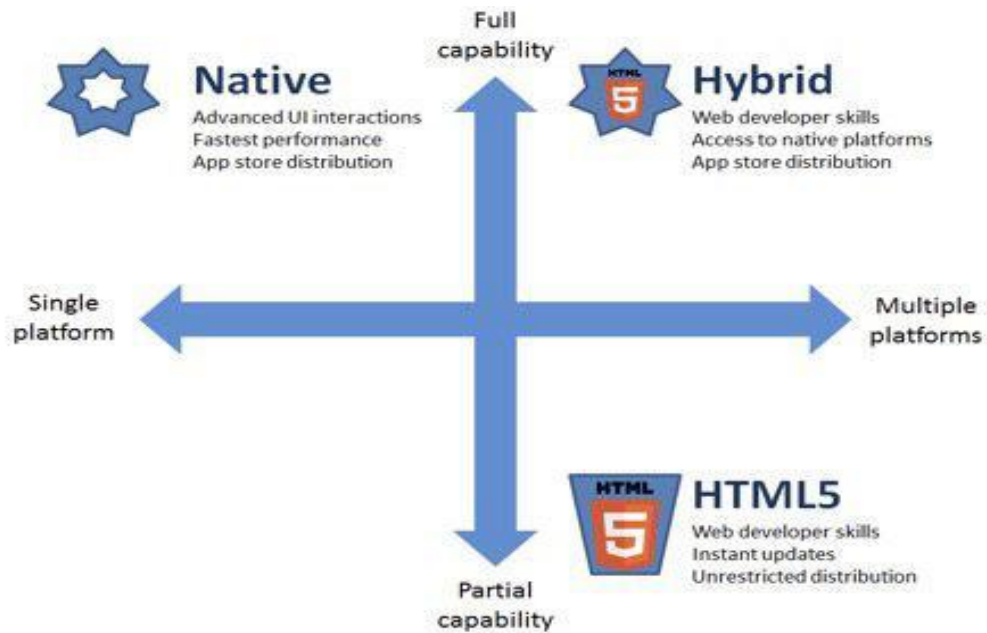


Рисунок 2.4 Порівняння нативної та гібридної розробки мобільного додатку

На рисунку 2.4 зображено, що ми можемо отримати від розробки мобільного додатку на гібридній основі та зробити висновок, чи підійде дана технологія для розробки мобільного додатку.

Отже, аналізуючи вище наведені дані, можемо зробити висновок, що використовуючи кросс-платформенну розробку мобільного додатку буде можливість отримати більшу швидкість розробки та його адаптивність під різні платформи. Для їх створення необхідні мінімальні знання вбудованих мобільних додатків IOS та Android, а також знання web-технологій. Гібридні додатки, як і web-додатки, програмуються з використанням web-технологій, але пакуються, як вбудовані додатки. Найчастіше це додатки, при розробці яких використовуються веб-технології, а не розробка в нативному середовищі. В результаті виходить веб-додаток, який запускається в «обгортці» і подається, як окремий додаток, що вимагає установки та має окрему іконку. Дані мобільні додатки поширюються через загальнодоступні сховища додатків, аналогічно до вбудованих. Основною відмінністю від web-додатків, є те, що вони мають прямий доступ до ресурсів мобільного пристрою, що робить їх більш

функціональними.

Найбільш популярним інструментальним засобом для розробки гібридних мобільних додатків є програмна платформа Ionic Framework. Платформа Ionic Framework – безкоштовний open-source фреймворк, який дозволяє створювати додатки для мобільних пристроїв, використовуючи JavaScript, AngularJS, HTML5, CSS3 та з допоміжними плагінами та програмними засобами, без необхідності знання нативних мов програмування під всі мобільні операційні системи. Приклад моделі розробки додатку за допомогою «Ionic Framework» представлений на рисунку 2.5.

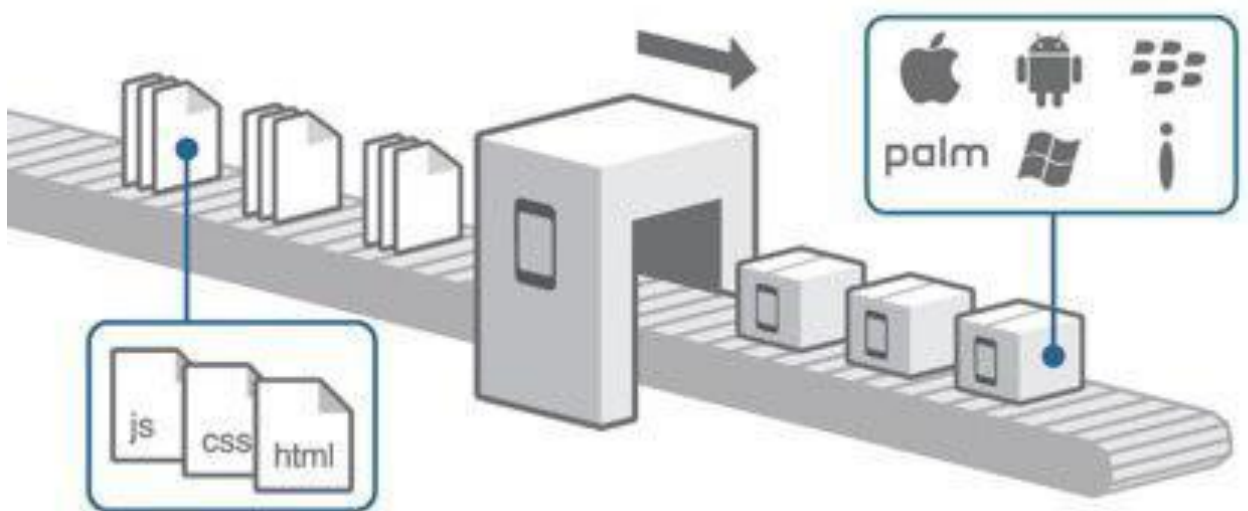


Рисунок 2.5 Приклад моделі розробки додатку за допомогою «Ionic Framework»

Движок Ionic Framework розширює API браузера і додає наступні можливості [23]:

- доступ до акселерометру;
- доступ до фотокамери;
- доступ до компасу;
- доступ до списку контактів;

- доступ до карти;
- доступ до файлової системи;
- доступ до системи сповіщення;
- запис і прослуховування аудіо файлів;
- дозволяє працювати з різними HTML5 сховищами localStorage, Web SQL;
- дозволяє безболісно звертатися до будь-якого крос-доменного адресу.

Перелік основних технологій та платформ для створення мобільного додатку представлений на рисунку 2.6.

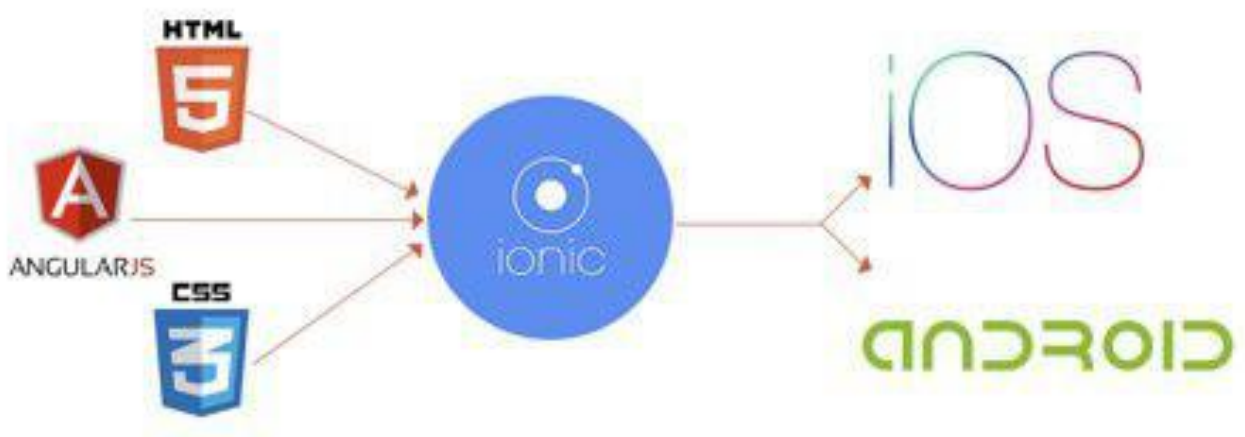


Рисунок 2.6 Перелік основних технологій та платформ для створення додатку

Ionic Framework надає набір JavaScript-інтерфейсів для доступу до багатьох функцій пристрою, недоступним з мобільних Web-браузерів для типових Web-додатків. Це досягається шляхом використання Web-додатків всередині вбудованої програми-оболочки. Перелік технологій, які використовуються в Ionic Framework зображено на рисунку 2.7.

Ionic Framework об'єднує код Web-додатків з візуалізатором браузера пристрою для створення «рідного» додатка, яке можна розгорнути в сховище додатків і встановити на пристрої. Базовий функціонал Ionic Framework можна розширити шляхом використання додаткових плагінів зі спеціального сховища.

На даний момент існує набір плагінів для операційних систем: iOS, Android, BlackBerry і Palm. Крім самої платформи для мобільних додатків, Ionic Framework надає доступ до «хмари», в якій можна зібрати проект, і завантажити на мобільний пристрій вже виконуваний файл. Для доступу до «хмари» необхідно зареєструватися на офіційному сайті Ionic Framework Build і після він стане доступний онлайн-«збирач» програм [23].

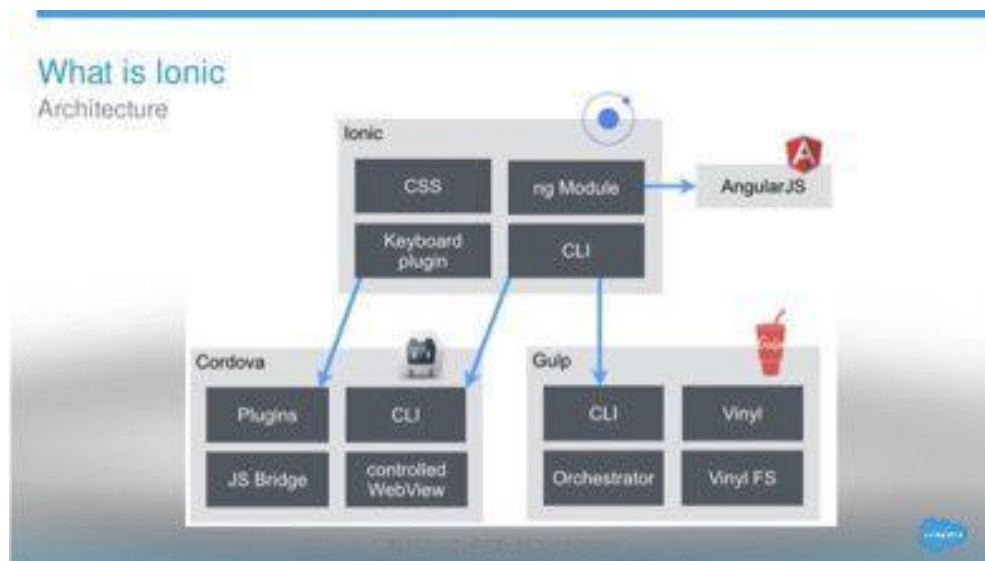


Рисунок 2.7 Перелік технологій, які використовуються в Ionic Framework

Перевагою даного сервісу можна виділити його загальнодоступність і велика бібліотека розробок інших користувачів з відкритим кодом. З іншого боку, саме це і є недоліком. В безкоштовній версії сервісу доступна збірка лише додатків з відкритим кодом, тобто після складання код вашої програми публікується в бібліотеці додатків і стає доступним іншим розробникам, як Open Source проект. Доступ до збірки програм з закритим кодом знаходиться тільки в платній версії сервісу.

Ionic Framework – технологія, яка дозволяє «переробити» web-сайт і зробити його умовно вбудованим додатком. Завдяки використанню вбудованих JavaScript-інтерфейсів, здійснюється доступ до багатьох функцій пристроїв, недоступних з мобільних Web-браузерів. При використанні Ionic Framework можна значно розширити функціонал web-сервісу, але він буде залишатися

ресурсом в мережі Інтернет. Для того, щоб web-сервіс став мобільним додатком необхідно також створити мобільний інтерфейс, який буде частиною вбудованого додатка, тобто Ionic Framework дозволяє створити власне вікно для перегляду веб-сторінок або навіть створити свій клон браузера за допомогою елемента для побудови гібридних мобільних додатків WebView. Існує велика кількість фреймворків для створення мобільного інтерфейсу, наприклад одним з таких є Dojo, який представлено на рисунку 2.8.

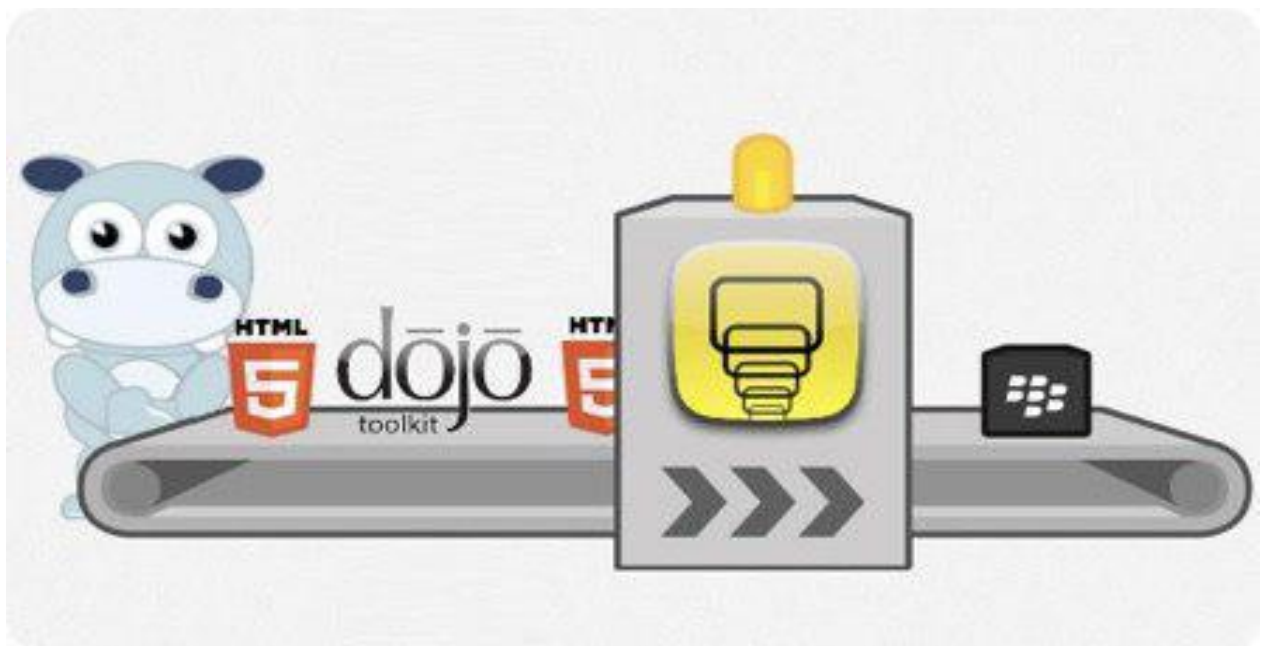


Рисунок 2.8 Логотип фреймворку Dojo

Елементи інтерфейсу Dojo – пакети, сформовані з компонентів: JavaScript-коду, розмітки HTML і CSS. Дані компоненти є View-елементами для Android-додатків, які спеціально адаптовані і стилізовані для мобільних пристроїв. Таким чином, у користувача формується відчуття, що він працює з вбудованим мобільним додатком за допомогою WebView, який дозволяє вбудовувати

веб-сторінки в мобільні додатки, формуючи своєрідний міні браузер.

При розробці гібридних мобільних додатків спочатку необхідно створити простий web-додаток з використанням web-технологій, ретельно його

налагодити, протестувати в браузері комп'ютера, а потім приступати до його адаптації для мобільних пристроїв, шляхом створення інтегрованої частини програми .

2.3 Архітектура вирішення задачі та функціональна структура додатку «ЗНО 2017 КОМПЛЕКС»

Найпростішою архітектурою для реалізації мобільного додатку є архітектура "файл-сервер" (рисунок 2.9), але вона ж володіє і найбільшою кількістю недоліків, які обмежують спектр вирішуваних нею задач. Найпростіший випадок, коли дані розташовуються фізично на тому ж комп'ютері, що і сам додаток [23].

При роботі з системою побудованої за такою архітектурою, можна виділити наступні недоліки:

- труднощі при забезпечуванні несуперечливості і цілісності даних;
- суттєве завантаження локальної мережі переданими даними;
- невисока швидкість обробки та подання інформації;
- високі вимоги до ресурсів комп'ютерів;
- неможливість організації рівноправного одночасного доступу користувачів до однієї і тієї ж ділянки бази даних.

При цьому система має одну важливу перевагу - низька вартість .

Архітектура «файл-сервер» передбачає концентрацію обробки на робочих станціях. Основною перевагою даного варіанту є простота і відносно низька вартість. Подібне рішення прийнятно, поки число користувачів, що одночасно працюють з базою даних, не перевищує 5-10 чоловік. При збільшенні кількості користувачів система може «захлинутися» через завантаженість великим потоком необробленої інформації.

Сервер – найпотужніший та самий надійний комп'ютер. Сервер обов'язково підключається до джерела безперебійного живлення, в ньому передбачаються системи подвійного чи потрійного дублювання. В виключних випадках, можна підключити декілька серверів, що при виході з ладу одного в роботу автоматично включиться «дублер». Таким чином, при концентрації обробки даних на сервері надійність системи обмежується лише матеріальними засобами, які замовники готові вкласти в технічне оснащення.

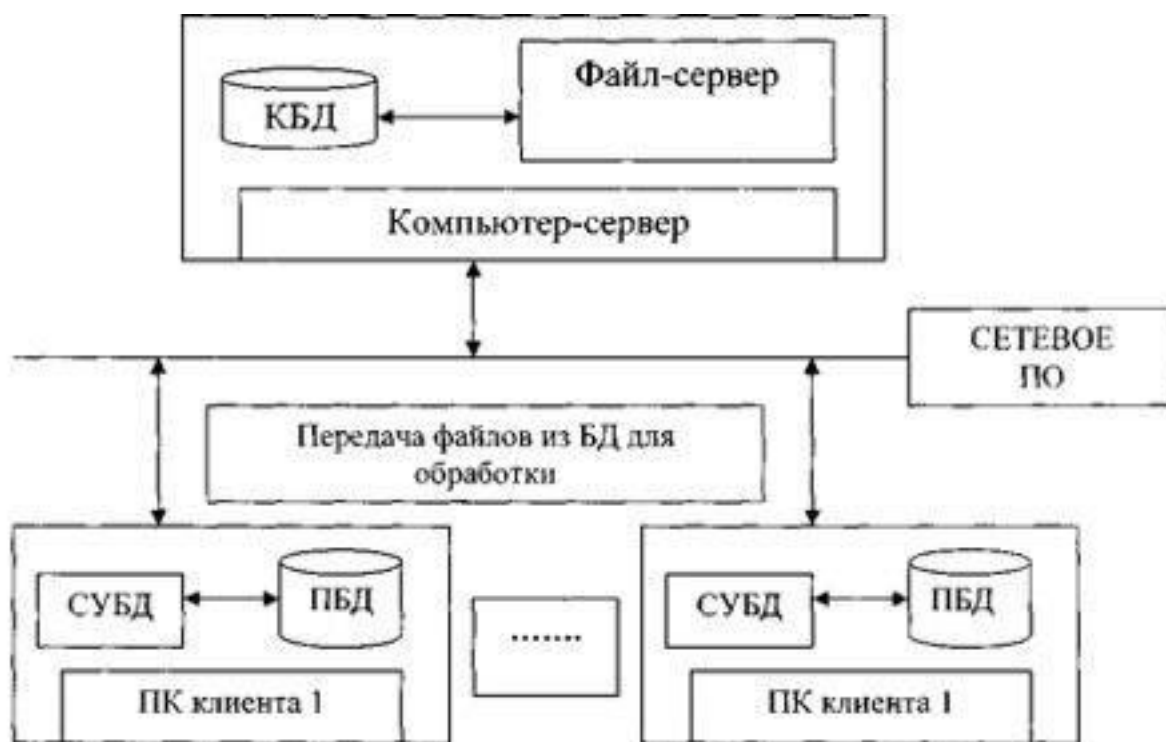


Рисунок 2.9 Структура інформаційної системи з «файл-сервером»

У базу даних на «файл-сервері» набагато простіше вносити зміни в таблиці, міняючи програми. Дана можливість полегшується тією обставиною, що у локальних СУБД база даних - набір окремих таблиць, що співіснують в єдиному каталозі на диску. Це говорить про низький рівень безпеки, як з точки зору розкрадання інформації так і внесення помилкових змін. Недостатньо розвинений апарат транзакцій для локальних СУБД служить потенційним джерелом помилок, як з точки зору одночасного внесення змін в один запис, а

також відкату результатів серій об'єднаних за змістом в єдине ціле операцій над базою, коли деякі з них завершилися успішно, а деякі – ні. Всі дані фактори можуть порушити смислову цілісність бази даних [24].

Недоліки архітектури «файл-сервер» вирішуються при перекладі додатків в архітектуру «клієнт-сервер», яка знаменує собою наступний етап у розвитку СУБД. Характерною особливістю архітектури «клієнт-сервер» є перенесення обчислювального навантаження на сервер бази даних (SQL-сервер), а також істотне зміцнення безпеки даних – розкрадання інформації чи внесення помилкових даних. База даних в цьому випадку розташована на мережевому сервері, аналогічно архітектурі «файл-сервер», проте прямого доступу до бази даних з додатків відбуватися не буде. Функцію прямого звернення до бази даних здійснює спеціальна керуюча програма - сервер бази даних (SQL-сервер).

Іншим видом архітектури для реалізації мобільного додатку є архітектура «клієнт-сервер».

Клієнт-серверна архітектура набула своєї популярності завдяки динамічному розвитку мережі інтернет та зосередження значної частини інформації в базах даних на серверах.

Клієнт-серверна архітектура – концепція інформаційної мережі в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів.

Відображення роботи архітектури «клієнт-сервер» представлено на рисунку 2.10.

Архітектура «клієнт-сервер» визначає такі типи компонентів:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Правило взаємодії між клієнтом і сервером називається протоколом обміну (протоколом взаємодії).

Модель клієнт-серверної взаємодії визначається розподілом обов'язків між клієнтом та сервером [25]. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який являє собою інтерфейс користувача, відповідає за представлення даних і відведення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку використання і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

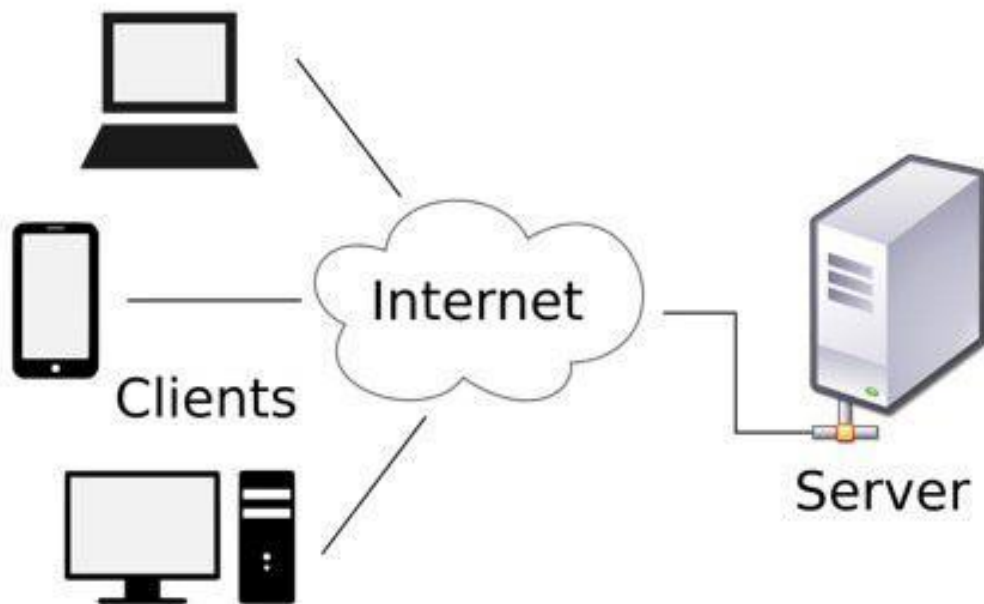


Рисунок 2.10 Відображення роботи архітектури «клієнт-сервер»

Приклад клієнт-серверної взаємодії – сервіс www. Існує велика кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Основна ідея архітектури «клієнт-сервер» полягає в розподілі мережевого додатку на кілька компонентів, кожен з яких реалізує набір сервісів [26].

Компоненти такого додатку можуть виконуватися на різних комп'ютерах, виконуючи серверні або клієнтські функції. Це дозволяє підвищити надійність, безпеку і продуктивність мережевих додатків та мережі в цілому [27].

Ще одним різновидом є архітектура «веб-сервер». До основних функцій «веб-сервера» відноситься:

Основні функції Веб-сервера:

- управління з'єднанням;
- прийом і обробка запиту;
- віддача статичного вмісту;
- взаємодія з додатками для отримання і подальшої віддачі клієнту динамічного змісту.

Моделі обробки з'єднань:

- Forking server;
- Pre-forking server;
- Worker threads;
- Event-driven.

Веб-сервер – сервер, що приймає HTTP-запити від клієнтів (веб-браузерів) і видає їм HTTP-відповіді, разом з HTML-сторінкою, зображенням, файлом, медіа-потоків або іншими даними. Веб-сервер – програмне забезпечення, яке виконує функції веб-сервера та безпосередньо комп'ютер-сервера, на якому дане програмне забезпечення функціонує. Клієнт, яким зазвичай виступає веб-браузер, передає веб-серверу запити на отримання ресурсів, позначені URL-адресами. Ресурси – HTML-сторінки, зображення, файли, медіа-потоків або інші дані, які необхідні клієнту. У відповідь веб-сервер передає клієнту запитувані дані. Цей обмін відбувається по протоколу HTTP.

На даний момент, широкої популярності в сфері software-рішень набирає використання безсерверної архітектури. Фактами даного явища, є поява спеціальної літератури, фреймворків, проведення конференцій тощо.

Безсерверна архітектура дозволяє розробникам більше не налаштовувати і адмініструвати сервери, на яких запущені керуючі програми всієї системи. В

даній архітектурі вся інфраструктура підтримується сторонніми провайдерами, а необхідна функціональність пропонується в формі сервісів, що відповідають за процеси аутентифікації, передачі повідомлень тощо. Безсерверні додатки широко використовують сторонні сервіси, які виконують завдання. Дані сервіси можуть являти собою цілі екосистеми (наприклад Azure і AWS) або бути самостійними рішеннями (наприклад Parse або Firebase).

Одним з головних завдань серверних веб-додатків є контроль циклу запитів та відповідей. Контролери на стороні сервера обробляють вхідні запити, запускають необхідні додатки і вибудовують відповіді. У разі безсерверної архітектури контролери на стороні сервера замінюються процесами генерації динамічного контенту. Суттєвою частиною серверних веб-додатків є бізнес-логіка, коли на сервері розташовується код, який виконує обробку запитів протягом всього часу роботи програми.

У безсерверних додатках спеціальні програмні компоненти активізуються тоді, коли надходить запит. Після обробки запиту компонент «засинає». Цей код часто розташовується в керованому середовищі, який піклується про масштабність коду і стежить за життєвим циклом.

Безсерверні системи не вимагають значних зусиль в підтримці зі сторони розробників. Сторонні компанії, що надають «хмарні» послуги, докладають безліч зусиль для забезпечення їх доступності та масштабності. Будь-яка оптимізація продуктивності FaaS-додатків автоматично знижує операційні витрати

(наприклад, виконання дії займало 1 секунду, а після внесення правок стало займати 200 мс – це призведе до зменшення вартості послуги на 80% без змін в інфраструктурі). На рисунку 2.11 зображена структура безсерверної архітектури.

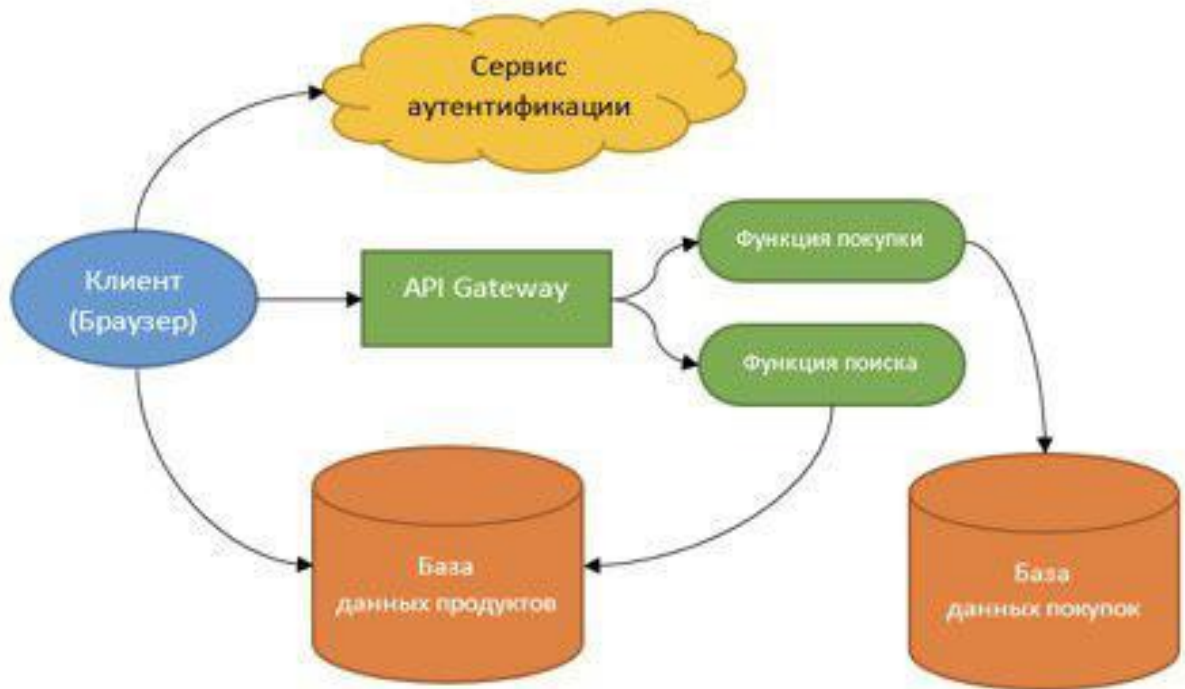


Рисунок 2.11 Структура безсерверної архітектури

Завдання аутентифікації тепер вирішує BaaS-сервіс (BaaS - Backend-as-a-Service). Через BaaS-сервіс також відбувається звернення до бази даних, яка хоститься «на стороні». Передача управління частиною системи іншій стороні визначає деякі недоліки, це може призвести до втрати контролю, наприклад, непередбачені апгрейди API. Також розбиття однієї програми на декілька частин з вплетеними в структуру різноманітними сервісами призводить до збільшення «числа точок перетину», що може ускладнити процедуру тестування.

Firebase надає «хмарну» NoSQL БД для real-time додатків як сервіс. Даний сервіс надає API для розробників, що дозволяє синхронізувати дані програми між клієнтами і зберігати їх в «хмарі» FireBase. Компанія також врахувала можливість інтеграції з Android, iOS, JavaScript, Java, Objective-C і Node.js додатками. Робота безпосередньо з базою даних реалізована через REST-сервіси для деяких JavaScript фреймворків, таких як AngularJS, React, Ember.js і Backbone.js. Для шифрування даних розробники можуть скористатися API, що надається також Firebase [27].

Firebase Hosting – це хостинг сервіс, запущений 13 травня 2014 року. Даний сервіс підтримує зберігання статичних файлів, таких як CSS, HTML, JavaScript. Сервіс передає файли по Content Delivery Network (CDN) з використанням HTTPS і SSL.

Проаналізувавши вище наведені дані, при виборі архітектури було віддано перевагу безсерверній системі при створенні мобільного додатку «ЗНО 2017 КОМПЛЕКС», яку буде зручно доповнювати, редагувати та підтримувати.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ «ЗНО 2017 КОМПЛЕКС»

3.1 Створення прототипу мобільного додатку

Визначивши програмну технологію реалізації додатка, предметну та об'єктну частину можемо приступити до прототипування головних частин додатку. Проаналізувавши обов'язкові та основні предмети на ЗНО можемо виділити предмети, які будуть входити до додатку: Історія України, Українська мова, Українська література, Географія, Математика, Англійська мова, Біологія, Фізика та Хімія.

Аналіз ринку мобільних операційних систем дозволяє зробити висновок, що ОС Android найбільш популярна та доступна серед молоді. Оскільки мобільний додаток, розроблений в даній роботі, буде орієнтований на майбутніх абітурієнтів, спочатку було розроблено та опубліковано саме версію мобільного додатку для цієї операційної системи.

З метою розуміння майбутнього інтерфейсу, за допомогою сервіса «Mocking Bot» було спроектовано декілька екранів мобільного додатку. На цьому етапі продумується вся функціональність додатку. Фактично будуть використані всі функції зазначені в технічному завданні та буде здійснено визначення того, як буде працювати додаток.

Mocking Bot – безкоштовний сервіс для прототипування мобільних додатків на різні платформи та з багатим функціональним набором інструментів. Для того, щоб скористуватися даним сервісом потрібно зареєструватися.

На рисунку 3.1 зображено інтерфейс для прототипування мобільних додатків в середині платформи «Mocking Bot».

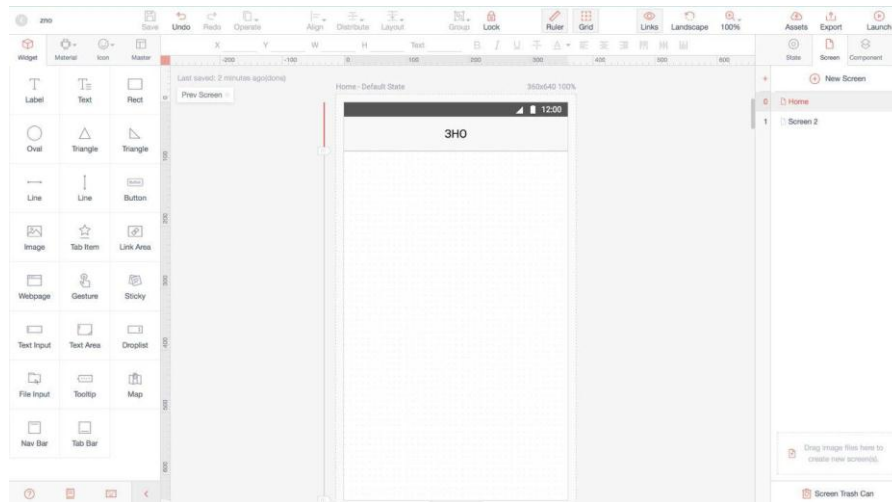


Рисунок 3.1 Інтерфейс прототипування «Mocking Bot».

Інтерфейс «Mocking Bot» має велику кількість корисних функцій для прототипування додатків: сітку для розміщення елементів, елементи інтерфейсу такі як status bar, progress bar, button, tooltip, nav bar, lines, label, text area тощо.

Обов'язкові шкільні предмети було представлені першими у вигляді списку, щоб користувач мав доступ до усіх предметів на одному головному екрані додатку, як представлено на рисунку 3.2.

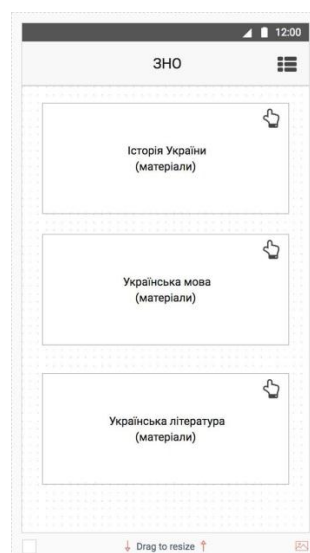


Рисунок 3.2 Прототип головного екрану мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Щоб викладений матеріал був структурно поданий, зручний для використання та легко запам'ятовувався, його необхідно розділити на категорії та згрупувати для кожного предмета, який буде містити додаток.

Основні категорії мобільного додатку:

- теорія;
- персоналії;
- хронологія дат;
- термінологія;
- дослідження;
- таблиці;
- корисні матеріали.

Кожний предмет буде мати свою унікальну підкатегорію, таку як формули, аналіз творів, псевдоніми, літературні жанри тощо. На рисунку 3.3 виділені прототипи предметів Історія України та Українська література, їх категорії.

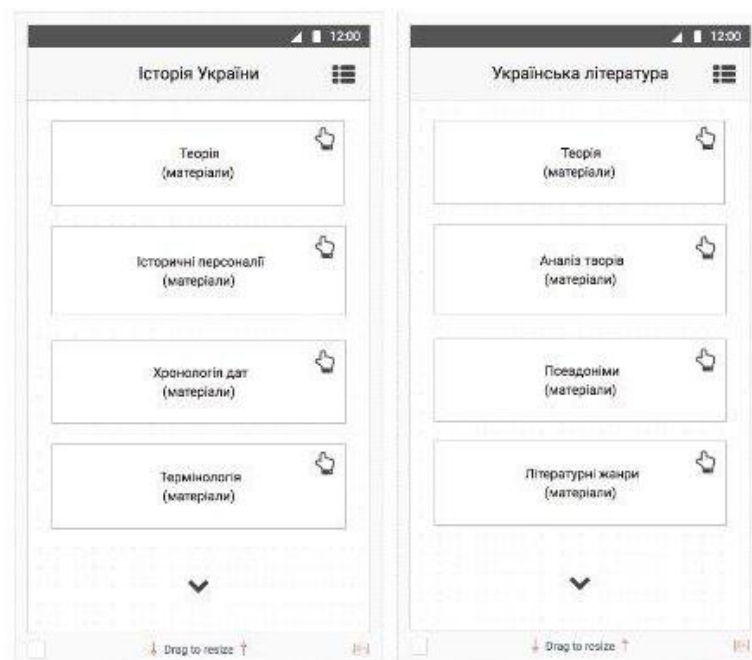


Рисунок 3.3 Прототипи категорій різних предметів мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Щоб користувачі мобільного додатку мали змогу бути в курсі новин пов'язаних з інформацією про ЗНО була створена ідея розробки категорії меню «Новини ЗНО». Ми вважаємо, це дасть абітурієнту можливість будь-де та будь-коли, отримати свіжі новини такі як, дати проведення, зміни в умовах ЗНО, терміни та особливості проведення тестування. На рисунку 3.4 представлений прототип екрану «Новини ЗНО».

По такому ж принципу було розроблено усі інші прототипи додатку, досліджено категорії та підкатегорії кожного предмету, також додано та спроектовано корисні функції додатку, такі як «Перекладач» та «Новини ЗНО».



Рисунок 3.4 Прототип екрану «Новини ЗНО» мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Процес створення прототипу, прообразу майбутнього об'єкта, стану та способів його виготовлення дає чітке розуміння того, як буде виглядати

майбутній проект. У проектуванні був використаний системний підхід, який полягав у встановленні структури системи, типу зв'язків, визначенні атрибутів, аналізі впливів зовнішнього середовища тощо.

Проектування мобільного додатку надало сукупність документації в єдиному вигляді, яка містить в собі технічні рішення, що дають повне уявлення про обладнання, яке проектується, а також всі початкові дані, необхідні для розробки робочої документації на стадії конструювання.

Маючи готові прототипи та задокументовані рішення можна побудувати наступну схему діяльності мобільного додатку, яка зображена на рисунку 3.5.

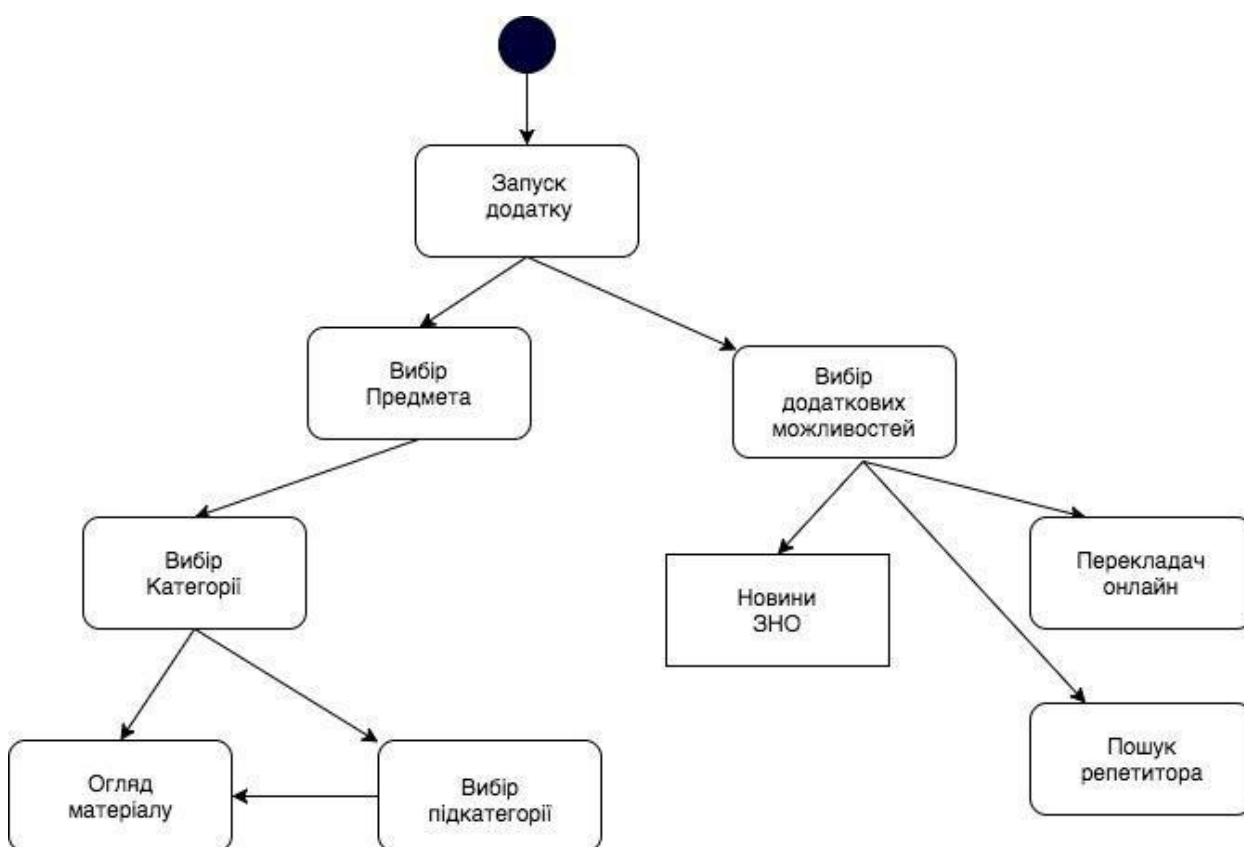


Рисунок 3.5 Схема діяльності мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Отримавши готову схему діяльності додатку та його функціональну частину, можна приступити до візуальної та програмної реалізації додатку.

3.2 Життєвий цикл розробки мобільного додатку до періоду впровадження та тестування

Модель життєвого циклу – структура, що складається із процесів, робіт та задач, які включають в себе розробку, експлуатацію та супровід програмного продукту, охоплює життя системи від визначення вимог до неї до припинення її використання. На сьогодні найбільшого розповсюдження набули дві моделі життєвого циклу:

- каскадна модель;
- спіральна модель.

Каскадна модель – модель, де кожен етап роботи виконується лише раз. На кожному етапі робота виконується настільки ретельно, що потреби повертатись до попереднього кроку не виникає. Результат виконання кожного етапу, перед передачею на наступний, піддається верифікації.

Спіральна модель – розробка ітераціями, яка відображає об'єктивно існуючий спіральний цикл створення системи. Неповне завершення робіт на кожному етапі дозволяє переходити на наступний етап, не чекаючи повного завершення роботи на поточному. При ітеративному способі розробки відсутню роботу можна буде виконати на наступній ітерації. Головне завдання – якнайшвидше показати користувачам системи працездатний продукт, тим самим активізуючи процес уточнення і доповнення вимог.

Для розробки мобільного додатку, було обрано каскадну модель, адже вона дає чіткість дій на кожному етапі розробки. Етапи розробки життєвого циклу за каскадною моделлю представлено на рисунку 3.6.



Рисунок 3.6 Каскадна модель життєвого циклу розробки мобільного додатку

Отже, обравши модель життєвого циклу розробки, спроектувавши прототипи, визначивши цілі та завдання мобільного додатку перейдемо до програмної реалізації.

Для розробки мобільного додатку була обрана технологія – Ionic Framework. Ionic Framework – безкоштовний open-source framework для створення мобільних додатків, який застосовує JavaScript, HTML5 та CSS3, з доповнюючими компонентами такі, як Cordova, AngularJS та інші.

Структура мобільного додатку розроблена за допомогою Ionic Framework v1 має аналогічну структуру веб-додатків, які створюються за допомогою JavaScript framework AngularJS, яку представлено на рисунку 3.7.

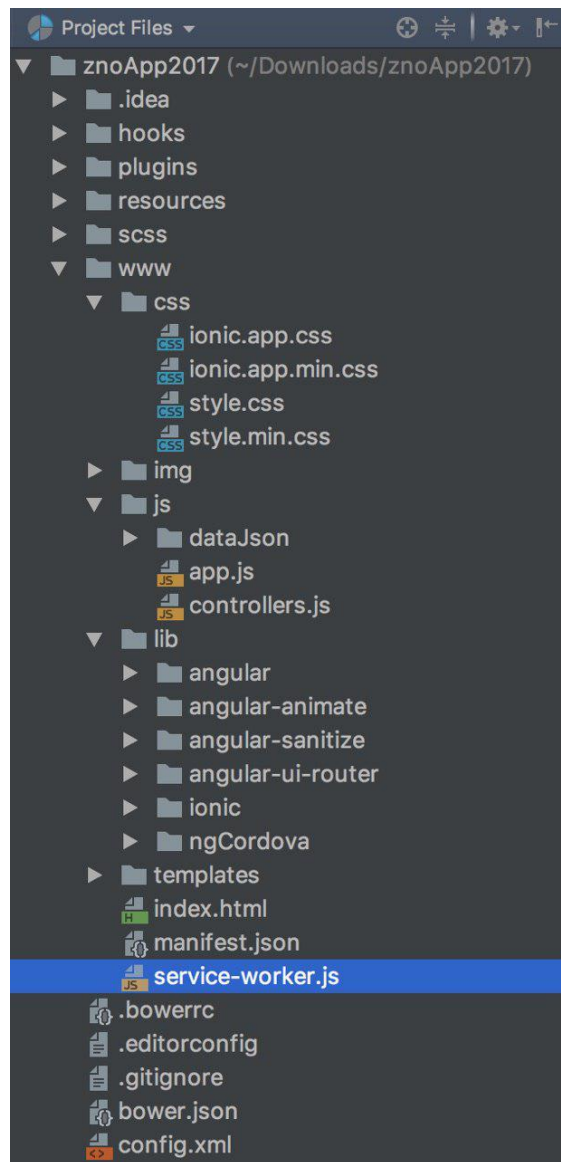


Рисунок 3.7 Структура мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Для створення макету мобільного додатку було використано програму Adobe Photoshop.

Опираючись на технічне завдання було створено ключові екрани. Ними виявились – головний екран, екран категорій, та екран матеріалу. Дизайн та інтерфейс додатку спроектовано відповідаючи всім сучасним стандартам по usability. Головним навігаційним елементом додатку – є вертикальне меню зліва меню. Дане меню присутнє на всіх екранах мобільного додатку (рисунок 3.8).

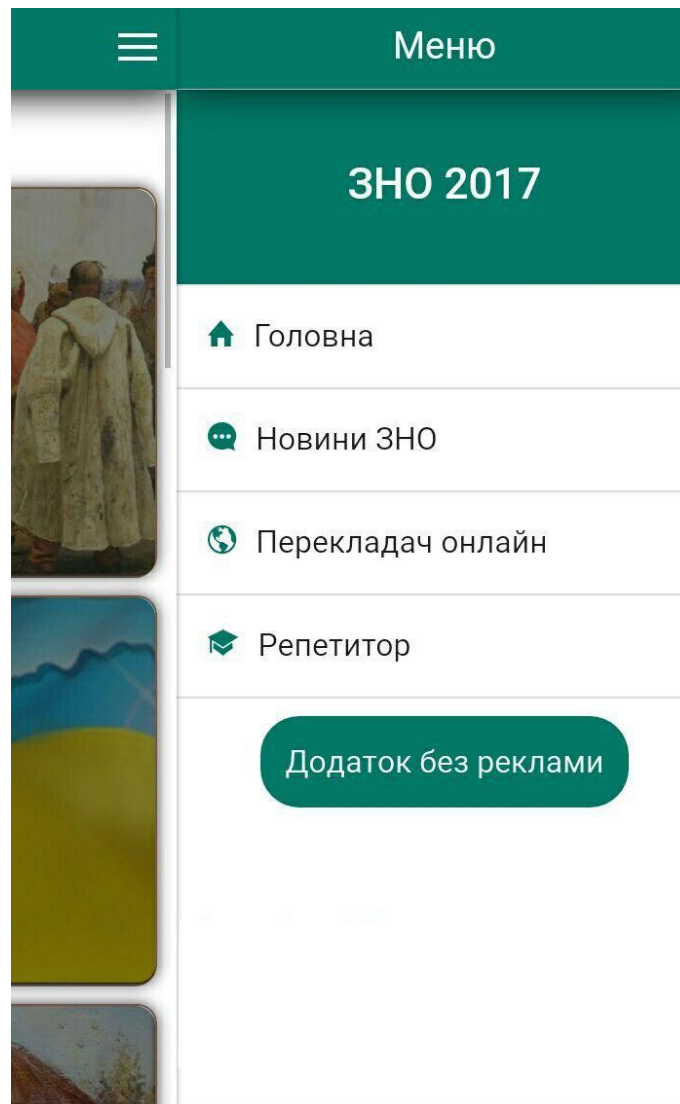


Рисунок 3.8 Дизайн головного меню мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Дизайн головного екрану, категорій та підкатегорій є одночасно насичений елементами та простий. На головному екрані знаходяться «обложки» – з графічним відображенням предмету та назвою самого предмета. Екрани категорій та підкатегорій мають таку ж структуру подання інформації (рисунок 3.9).



Рисунок 3.9 Головний екран мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Окремо потрібно виділити дизайн екранів із контентом. До його створення ми віднеслись з особливою увагою, адже ці екрани повинні бути максимально простими, без відволікаючих елементів, фонів та іншого, щоб користувач міг максимально сфокусувати свою увагу на представлений матеріал. Навчальний матеріал повинен бути в свою чергу структурований та відповідати усім вимогам оформлення. Екран теоретичного матеріалу на прикладі предмету Історія України представлений на рисунку 3.10 та рисунку 3.11.

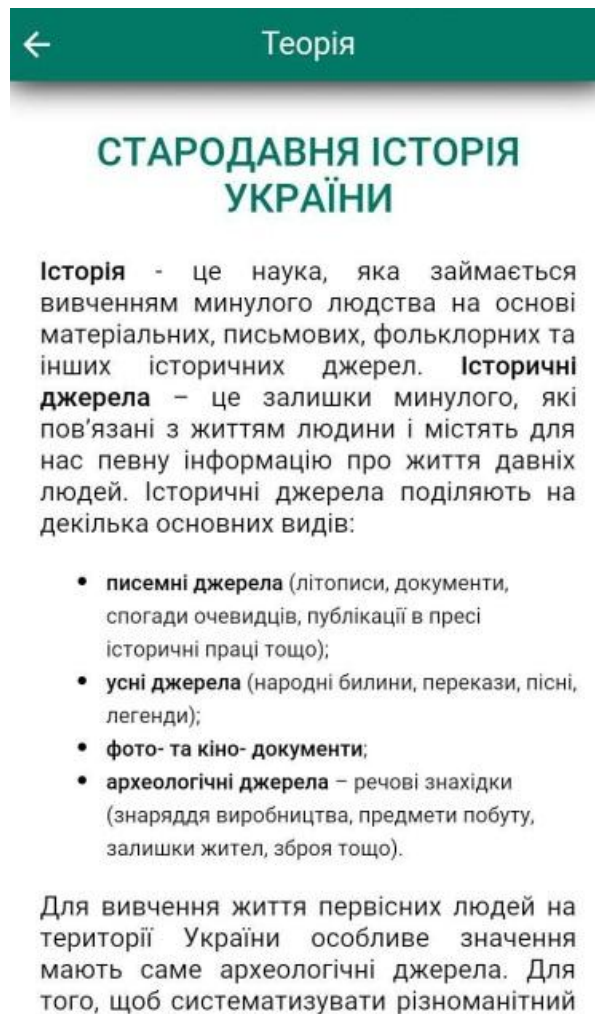


Рисунок 3.10 Екран теоретичного матеріалу мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Всі інші екрани є дуже типовими і тому для них не розроблявся окремий дизайн. Всі вони притримуються стилю та кольоровій гаммі раніше показаних екранів.



Рисунок 3.11 Екран хронології дат мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Після створення дизайну мобільного додатку потрібно перейти до його реалізації. Для розробки мобільних додатків під платформу Android, зокрема представленого в даній дипломній роботі, потрібно середовище розробки. Були обрані наступні середовища Web Storm та Android Studio.

Маючи чітке уявлення, як буде виглядати майбутній додаток та маючи раніше створену структуру проекту на Ionic Framework, який працює на базі AngularJS, було почато розробку розмітки всіх сторінок мобільного додатку за допомогою HTML5 та CSS3, адже, як шаблон AngularJS використовує комбінацію бажаних HTML-тегів і виразів.

Інформація на екрані смартфона буде візуалізується HTML5 шаблонами

та CSS3 селекторами. Нижче наведено приклад HTML код головного екрану (на рисунку 3.9 представлено скріншот початкового вікна системи, де користувач має можливість вибрати предмет для вивчення матеріалу).

HTML код головного екрану додатку:

```
<ion-view view-title="Головна">
  <ion-content class="padding">
    <h4>Оберіть предмет:</h4>
    <section class="subject subject__bc-history" ng-click="goToSubjects('Історія України')">
      <span class="subject_title">Історія України</span>
    </section>
    <section class="subject subject__bc-ukr-lang" ng-click="goToSubjects('Українська мова')">
      <span class="subject_title">Українська мова</span>
    </section>
    <section class="subject subject__bc-ukr-lit" ng-click="goToSubjects('Українська література')">
      <span class="subject_title">Українська література</span>
    </section>
    <section class="subject subject__bc-geography" ng-click="goToSubjects('Географія')">
      <span class="subject_title">Географія</span>
    </section>
    <section class="subject subject__bc-math" ng-click="goToSubjects('Математика')">
      <span class="subject_title">Математика</span>
    </section>
    <section class="subject subject__bc-english" ng-click="goToSubjects('Англійська')">
      <span class="subject_title">Англійська</span>
    </section>
    <section class="subject subject__bc-biologi" ng-click="goToSubjects('Біологія')">
      <span class="subject_title">Біологія</span>
    </section>
    <section class="subject subject__bc-physics" ng-click="goToSubjects('Фізика')">
      <span class="subject_title">Фізика</span>
    </section>
    <section class="subject subject__bc-chemistry" ng-click="goToSubjects('Хімія')">
      <span class="subject_title">Хімія</span>
    </section>
```

```
</ion-content>
```

```
</ion-view>
```

Для забезпечення переходів між усіма екранами всередині додатку було створено файл Routes.js короткий код якого наведено нижче, повний код файлу можна переглянути в додатку Б.

JavaScript код файлу Routes.js

```
.config(function ($stateProvider, $urlRouterProvider, $ionicConfigProvider) {
  $ionicConfigProvider.navBar.alignTitle('center');
  $stateProvider
    .state('app', {
      url: '/app',
      abstract: true,
      templateUrl: 'templates/menu.html',
      controller: 'AppCtrl'
    })
    .state('app.main', {
      url: '/main',
      views: {
        'menuContent': {
          templateUrl: 'templates/main.html',
          controller: 'MainCtrl'
        }
      }
    })
    .state('app.subject', {
      url: '/subject',
      views: {
        'menuContent': {
          templateUrl: 'templates/subject.html',
          controller: 'MainCtrl'
        }
      }
    })
  $urlRouterProvider.otherwise('/app/main');
```

```
});
```

JavaScript код функції, що контролює вибір предмета на головному екрані мобільного додатку представлений нижче.

```
$scope.goToSubjects = function (name) {  
  switch (name) {  
    case "Історія України":  
      $rootScope.titleSubject = name;  
      $state.go('app.subject');  
      $rootScope.initEvent("Історія України");  
      break;  
    case "Українська мова":  
      $rootScope.titleSubject = name;  
      $state.go('app.subject');  
      $rootScope.initEvent("Українська мова");  
      break;  
    case "Українська література":  
      $rootScope.titleSubject = name;  
      $state.go('app.subject');  
      $rootScope.initEvent("Українська література");  
      break;  
    case "Географія":  
      $rootScope.titleSubject = name;  
      $state.go('app.subject');  
      $rootScope.initEvent("Географія");  
      break;  
    case "Математика":  
      $rootScope.titleSubject = name;  
      $state.go('app.subject');  
      $rootScope.initEvent("Математика");  
      break;  
    case "Англійська":  
      $rootScope.titleSubject = name;  
      $state.go('app.subject');  
      $rootScope.initEvent("Англійська");
```



```

    break;
case "Біологія":
    $rootScope.titleSubject = name;
    $state.go('app.subject');
    $rootScope.initEvent("Біологія");
    break;
case "Фізика":
    $rootScope.titleSubject = name;
    $state.go('app.subject');
    $rootScope.initEvent("Фізика");
    break;
case "Хімія":
    $rootScope.titleSubject = name;
    $state.go('app.subject');
    $rootScope.initEvent("Хімія");
    break;
}
};

```

Оскільки матеріал був створений та заготовлений заздалегіть із безкоштовно доступних в інтернеті матеріалів, було почато його додавання у додаток у вигляді JSON формату.

JSON (JavaScript Object Notation) – простий формат обміну даними, зручний для читання і написання як людиною, так і комп'ютером. Він заснований на підмножині мови програмування JavaScript, визначеного в стандарті ECMA-262. JavaScript Object Notation – текстовий формат, повністю незалежний від мови реалізації. Завдяки цьому, це робить JSON ідеальним для обміну даними.

Код формату JSON теоретичних матеріалів для предмета - Історія України.

```

{{
  "title": "Стародавня історія України",
  "number": 1
},

```

```

{
  "title": "Виникнення та розквіт Київської Русі",
  "number": 2
},
{
  "title": "Київська Русь за часів роздробленості. Галицько- Волинська держава",
  "number": 3
},
{
  "title": "Політичний устрій, соціально- економічний, культурний розвиток Київської Русі та Галицько- Волинської держави у IX - XIV ст.",
  "number": 4
},
{
  "title": "Українські землі у складі Великого князівства Литовського та інших держав (у другій половині XIV - першій половині XVI ст.)",
  "number": 5
}]

```

Ключ «number» кожного об'єкта у JSON файлі вказує на тему теоретичного матеріалу, який був заздалегіть підготовлений у файлі HTML та контролюється JavaScript функцією, повний код якої можна переглянути в додатку В.

Процес завантаження коректного HTML5 файлу, назва якого відповідає ключу «number» в JSON файлі, який було продемонстровано вище керувався JavaScript функцією зображеною нижче:

```

$scope.initTheoryHTML = function () {
  switch ($rootScope.titleSubject) {
    case "Історія України":
      $scope.link = 'templates/History_Theory/theory_' + $stateParams.theory_ID + '.html';
      break;
    case "Українська мова":
      $scope.link = 'templates/Ukr_Theory/theory_' + $stateParams.theory_ID + '.html';
      break;
  }
}

```

```

case "Українська література":
    $scope.link = 'templates/Ukr_lit_Theory/theory_' + $stateParams.theory_ID + '.html';
    break;
case "Географія":
    $scope.link = 'templates/Geography_Theory/theory_' + $stateParams.theory_ID + '.html';
    break;
case "Математика":
    $scope.link = 'templates/Math_Formula/theory_' + $stateParams.theory_ID + '.html';
    break;
case "Англійська":
    $scope.link = 'templates/English_Theory/theory_' + $stateParams.theory_ID + '.html';
    break;
case "Біологія":
    $scope.link = 'templates/Biology_Theory/theory_' + $stateParams.theory_ID + '.html';
    break;
case "Фізика":
    $scope.link = 'templates/Physics_Theory/theory_' + $stateParams.theory_ID + '.html';
    break;
case "Хімія":
    $scope.link = 'templates/Chemistry_Theory/theory_' + $stateParams.theory_ID + '.html';
    break;
}
};

```

Таким чином, було створено усі інші HTML5 та JavaScript файли для кожного предмета, категорії та підкатегорії, що дало максимально чітко створити всю архітектуру проекту, структурно та легко згрупувати кожний компонент.

Основний функціонал та матеріали для вивчення, розміщені на мобільному телефоні, однак існує і підключення до Firebase сервера, де тримаються дані для відображення останніх новин пов'язаних із ЗНО. Firebase дає можливість розробнику в режимі реального часу додавати матеріали новин до додатку у відповідний пункт меню «Новини ЗНО». Код сервісу та котроллера роботи Firebase можна переглянути в додатку Д. Зовнішній вигляд

готової реалізації зображений на рисунку 3.12.



Рисунок 3.12 Екран «Новини ЗНО» мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Для зручного вивчення предмету «Англійська мова», було створено функцію «Перекладач», в якій користувач має змогу перекладати текст з української мови на англійську мову та навпаки, яка допоможе при підготовці до тестування або в повсякденному житті. Для функції «Перекладач» за основу взято API yandex translate, що надає свої послуги безкоштовно. Код виконання даної функції можна переглянути в додатку Г. Зовнішній вигляд готової реалізації функції «Перекладач» зображений на рисунку 3.13

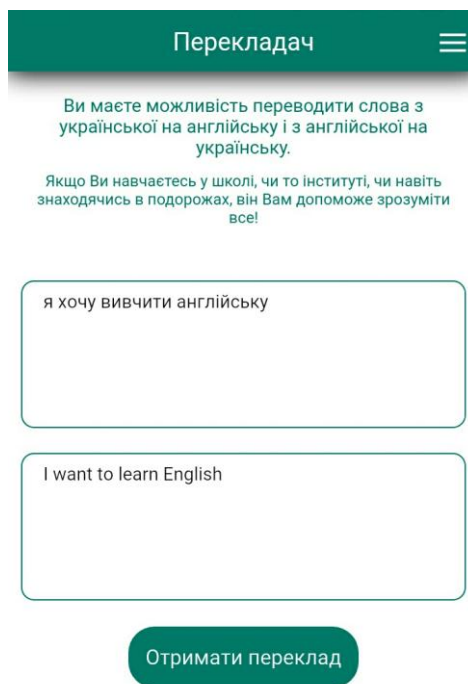


Рисунок 3.13 Екран функції «Перекладач» мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Отже, розроблений основний функціонал мобільного додатку, також продемонстровано реалізацію ключових функцій додатку, а саме: головної сторінки, категорій, роутингу між екранами, функції новин та перекладача, екран відображення теоретичного матеріалу та функції керування цими процесами всередині додатку.

3.3 Впровадження та тестування мобільного додатку

Невід’ємною частиною будь-якої реалізації програмного продукту є процес його тестування, так званої перевірки роботи програми згідно поставлених вимог, які були зазначенні при початку реалізації продукту. Під час перевірки програмного продукту зазвичай виявляються дефекти системи, при яких система може не працювати, блокуватися або працювати не вірно. В

разі виявлення дефекту або помилки, розробник має виправити її.

Тестування мобільного додатку поділяється на два види: ручне тестування та автоматизоване. Під час ручного тестування всі операції виконуються вручну, тобто розробники самі перевіряють та оцінюють, як працює та чи інша функція. Автоматизоване тестування, в свою чергу, дає змогу перевірити код реалізації функцій, за допомогою якого маємо можливість протестувати функціонал і перевірити, чи відповідає поставленій задачі потрібний результат. Кожен з цих видів тестування містить в собі вже свої види тестування.

Під час проведення ручного тестування мобільний додаток було успішно встановлено на смартфони з такими операційними системами:

- Android 4.4.4(Kit Kat);
- Android 5.0(Lollipop);
- Android 5.1(Lollipop);
- Android 6.0(Marshmallow).

На даний операційних системах додаток пройшов тестування в ручному режимі успішно, продукт готовий до використання та експлуатації, фатальних багів виявлено не було. Для розгортання програмного продукту потрібна операційна система Android версії 4 та вище.

В результаті виконання завдань роботи було вирішено, що перед впровадженням готового продукту та його підтримкою, доцільно було б інтегрувати аналітику, яка допоможе зрозуміти які предмети, категорії, чи розділи найбільш популярні серед користувачів, що дало б чітку картину того, що потрібно покращити чи додати. Отже, для аналітики було використано сервіс Google Analytics.

Google Analytics (скорочено GA) – безкоштовний сервіс, що надається Google для створення детальної статистики відвідувачів різних мобільних додатків. Статистика збирається на сервері Google, користувач тільки розміщує JS-код на сторінках свого додатку. Код відстеження спрацьовує, коли користувач відкриває сторінку в своєму веб-браузері або ж додатку (за умови

дозволеного виконання Javascript у браузері).

Для інтегрування аналітики використовується плагін «cordova-plugin-google-analytics». Ініціалізація якого виглядає наступним чином.


```
if (typeof analytics !== undefined) {
  analytics.startTrackerWithId(UA-(my-number-app));
}
$rootScope.trackView = function (name) {
  if (typeof analytics !== "undefined") {
    analytics.trackView("Предмет:" + name);
  }
};
```

За допомогою JavaScript функції «trackView», що вказана в коді вище, можемо відстежувати предмет на якому зараз знаходиться користувач.

Наступним кроком було завантаження мобільного додатку, для мобільної платформи Google Play Market, для мобільної платформи ОС Android. Для цього було створено обліковий запис google play developer console. Для налаштування сторінки додатку в Play Market необхідно:

- логотип додатку, який було створено за допомогою програмного продукту Adobe Photoshop;
- банер додатку;
- короткий опис додатку;
- повний опис додатку;
- відповіді на декілька запитань, щоб з'ясувати чи відповідає додаток стандартам мобільного магазину додатків.

Після успішно доданих матеріалів додаток став доступний для усіх користувачів, які мають доступ до Google Play Market (рисунок 3.14).

| Название приложения | Установок (активных/всего) | Сред. оценка / Всего | Последнее обновление | Статус |
|--|----------------------------|----------------------|----------------------|--------------|
|  ЗНО 2017 Комплекс com.TigerSoft.ZNO | 11 869 / 72 399 | ★ 4,60 / 367 | 11 июля 2017 г. | Опубликовано |

Страница 1 из 1

Рисунок 3.14 Опублікований мобільний додаток «ЗНО 2017 КОМПЛЕКС»

Протягом першого місяця роботи додатку його завантажило 1757 користувачів. Статистика завантажень стрімко зростала, всі користувачі активно користувалися мобільним додатком, результати завантажень зображено на рисунку 3.15.

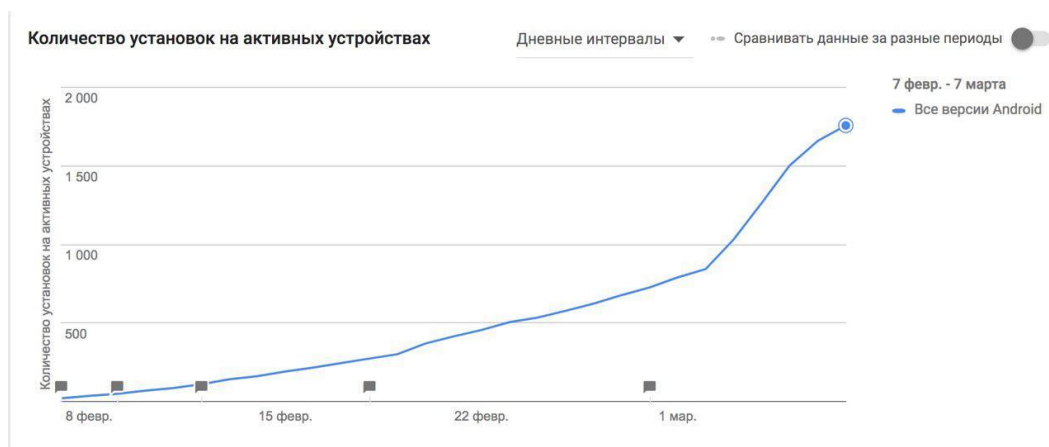


Рисунок 3.15 Графік завантажень мобільного додатку «ЗНО 2017 КОМПЛЕКС» за перший місяць роботи

З кожним місяцем завантаження збільшувалися, користувачі почали залишати гарні відгуки, що дало змогу додатку зайняти одну з лідируючих позицій в Google Play по запиту ЗНО, і за перші майже чотири місяці додаток набрав 31 704 тис. завантажень (рисунок 3.16).

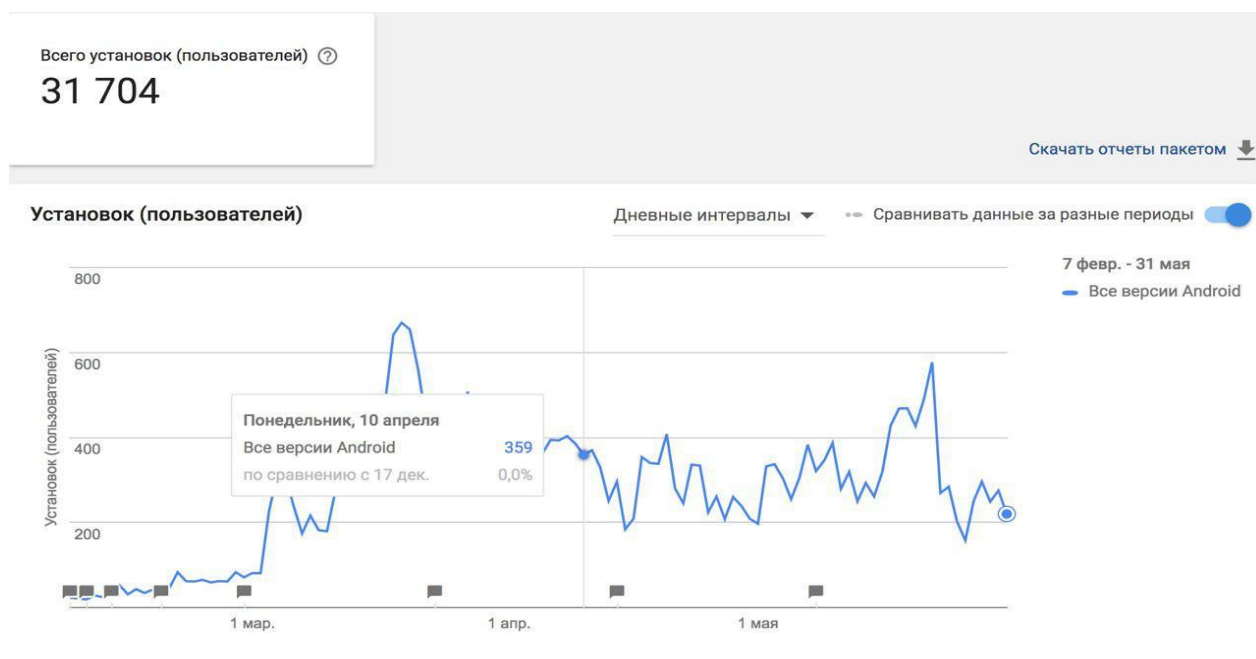


Рисунок 3.16 Завантаження за перші чотири місяці доступності мобільного додатку «ЗНО 2017 КОМПЛЕКС»

Аналіз графіку показує, що кожного дня кількість завантажень мобільного додатку склала близько 350 користувачів, а іноді майже 600 користувачів. За весь час додаток набрав 72 626 тис. користувачів (рисунок 3.17)

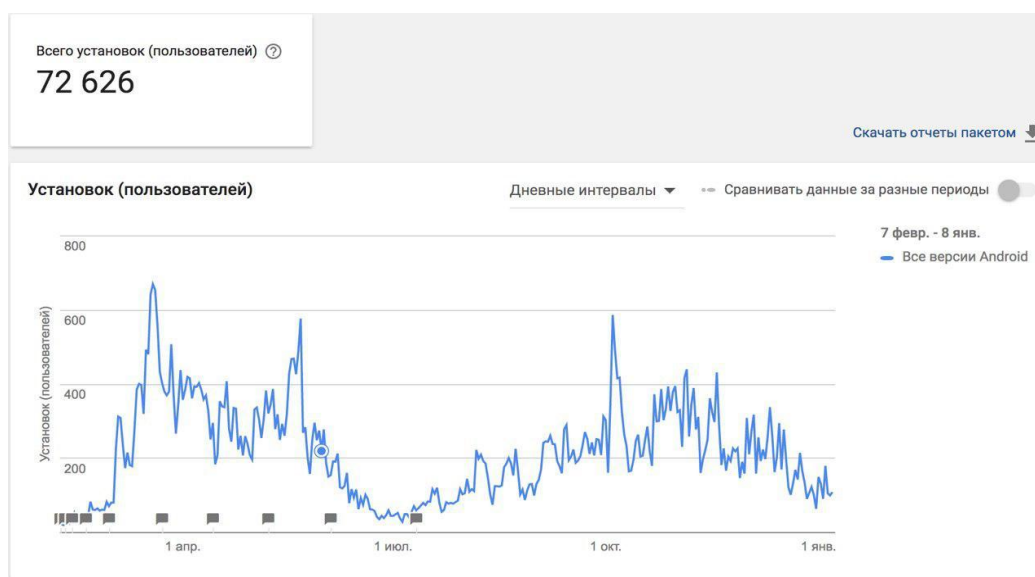


Рисунок 3.17 Графік завантажень мобільного додатку «ЗНО 2017 КОМПЛЕКС» за період з лютого 2017 по січень 2018 років

Аналізуючи вище наведені дані, можемо зробити висновок, що мобільний додаток «ЗНО 2017 Комплекс» показав хороші та ефективні результати при підготовці абітурієнтів до тестування. Про це свідчить досить значний показник завантажень додатку, а також позитивні відгуки користувачів.

Розроблений додаток містить комплексний теоретичний матеріал для підготовки до зовнішнього незалежного оцінювання з Історії України, Української мови, Української літератури, Географії, Математики, Англійської мови, Біології, Фізики та Хімії та містить корисний функціонал: перекладач онлайн, новини ЗНО, пошук репетитора.

Успішне впровадження чітко спроектованого та розробленого мобільного додатку підтвердило, що використання мобільних додатків з теоретичними матеріалами та корисними функціями для підготовки до зовнішнього незалежного оцінювання є досить затребуваним та актуальним. Додаток надав користувачам безперебійний та безкоштовний доступ до матеріалів з мобільних пристроїв, що дало змогу ефективно готуватися до ЗНО в зручній для користувачів час. Про це свідчить статистика зібрана на основі впровадженого додатку. Додаток з комплексом матеріалів для підготовки стрімко набрав популярність серед користувачів, так за перші чотири місяці роботи додаток набрав 31 тис. користувачів, та гарних відгуків.

ВИСНОВОК

Сучасний розвиток інформаційно-комунікаційних технологій розкриває широкі можливості для використання мобільних технологій в системі освіти. Мобільне навчання, що базується на інтенсивному застосуванні сучасних мобільних засобів та технологій, надає можливість зробити процес гнучким, дос-тупним і персоніфікованим. Мобільне навчання стає новою освітньою парадигмою, на основі якої створюється нове навчальне середовище, де студенти та учні можуть отримати доступ до навчальних матеріалів у будь-який час, що стимулює того, хто навчається, до самоосвіти та навчання протягом життя.

В процесі написання дипломної роботи були вирішені всі поставлені задачі:

- досліджена сутність мобільного навчання та шляхи його впровадження в систему освіти;
- проведено аналіз методів та технологій розробки мобільних додатків;
- обрано технологію розробки для створення мобільного додатку;
- сформовано функціональну структуру мобільного додатку для підготовки до зовнішнього незалежного оцінювання;
- робленено та впроваджено мобільний додаток для платформи Android для підготовки до зовнішнього незалежного оцінювання «ЗНО 2017 КОМПЛЕКС»;
- розроблений додаток було протестовано.

Додаток для платформи Android «ЗНО 2017 КОМПЛЕКС» містить матеріали для підготовки до зовнішнього незалежного оцінювання з української мови, української література, історії України, географії, математики, англійської мови, біології, фізики та хімії; корисний функціонал: перекладач онлайн, новини ЗНО, пошук репетитора. Для розробки мобільного додатку були обрані технології: HTML5, CSS3, JavaScript, JSON, Firebase, Ionic.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мазурок И.Е., Мазурок Т.Л. Использование мобильных коммуникационных устройств в образовательных целях // Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць. Випуск V: В 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2009. – Т. 3. – С. 175–179.
2. Семеріков С.О., Теплицький І.О., Шокалюк С.В. Нові засоби дистанційного навчання інформаційних технологій математичного призначення // Вісник. Тестування і моніторинг в освіті. – 2008. – №2
3. Голицына И. Н. Мобильное обучение как новая технология в образовании [Электронный ресурс] / И. Н. Голицына. – Режим доступа : http://ifets.ieee.org/russian/depository/v14_i1/html/1.htm.
4. Калуга Т. А. Мобильное обучение в дистанционном образовании [Электронный ресурс] / Т. А. Калуга // Вісник ЛНУ імені Тараса Шевченка. – 2011. – № 12 (223), Ч. I. – С. 113–123. – Режим доступа до журн. : http://www.nbu.gov.ua/portal/Soc_Gum/Vlush/Ped/2011_12_1/15.pdf.
5. Куклев В. А. Сущностные характеристики мобильного обучения как педагогической инновации [Электронный ресурс] / В. А. Куклев // Мир науки, культуры, образования. – 2008. – № 5(12). – С. 204–207. Режим доступа до журн.: <http://www.iwep.ru:88/journal/12/pages%20204-207.pdf>.
6. Семеріков С. О. Фундаменталізація навчання інформаційних дисциплін у вищій школі : монографія / С. О. Семеріков; науковий редактор академік АПН України, д. пед. н., проф. М. І. Жалдак. – К. : НПУ ім. М. П. Драгоманова, 2009. – 340 с.
7. Триус Ю. В. Інноваційні інформаційні технології у навчанні математичних дисциплін / Ю. В. Триус // Інноваційні комп'ютерні технології у вищій школі : матеріали 3-ої Науково-практичної конференції, 8–12 жовтня 2011 року, Львів / Національний університет "Львівська політехніка". – Львів : Видавництво Львівської політехніки, 2011. – С. 61–68.

8. Гуревич Р.С. Інформаційно-комунікаційні технології в професійній освіті майбутніх фахівців / Р.С. Гуревич, М.Ю. Кадемія, М.М. Козяр; за ред. член-кор. НАПН України Гуревича Р.С. – Львів: ЛДУ БЖД, 2012. – 380 с.
9. Триус Ю.В., Франчук В.М., Франчук Н.П. Організаційні й технічні аспекти використання систем мобільного навчання // Науковий часопис НПУ імені М. П. Драгоманова. Серія 2. Комп'ютерно-орієнтовані системи навчання: Зб. наук. праць./Педрада. – К.: НПУ ім. М. П. Драгоманова, 2011. – №12(19). – С. 53 –62.
10. Мобильное обучение [Електронний ресурс]. – Режим доступу : <http://goo.gl/7UdXI>.
11. Станіслав Ніколаєнко про використання мобільних телефонів в школі (25 травня 2008) [WWW документ]. URL http://www.loga.gov.ua/oda/about/depart/guon/news/2007/05/25/news_262.html?template=33 (10 березня 2008)
12. Дубова Н. E-Learning – Обучение с приставкой «е» / Н. Дубова // Открытые системы. – 2011. – № 11. – Режим доступу до журн. : <http://www.osp.ru/os/2004/11/184806>.
13. Мобільне навчання [Електронний ресурс]. – Режим доступу : <http://goo.gl/Yb9Pz>.
14. Мобильное обучение – второе рождение, но те же трудности [Електронний ресурс]. – Режим доступу : http://elearningtime.blogspot.com/2011/01/blog-post_17.html.
15. Доля Android на рынке смартфонов [Електронний ресурс]: Ferra.ru – Режим доступу: <http://www.ferra.ru/ru/techlife/news/2014/01/30/strategyanalytics-2013-smartphone/#.U6DXXyjiLt0>
16. Операційна система Google Android [Електронний ресурс]: ALLS.IN.UA – Режим доступу: <http://alls.in.ua/13729-operacijna-sistema-googleandroid.html>
17. Доля Android на рынке смартфонов [Електронний ресурс]: Ferra.ru – Режим доступу: <http://www.ferra.ru/ru/techlife/news/2014/01/30/strategyanalytics-2013-smartphone/#.U6DXXyjiLt0>

18. Новая среда разработки Android Studio [Электронный ресурс]: cnews.ru – Режим доступа: http://www.cnews.ru/top/2013/05/20/novaya_sreda_razrabotki_android_studio_sozdana_na_osnove_rossiyskogo_proekta529258
19. Android Studio – среда мобильной разработки на базе технологий JetBrains [Электронный ресурс]: soft.mail.ru – Режим доступа: http://soft.mail.ru/pressrl_page.php?id=51774
20. UX – это не UI [Электронный ресурс]: cmsmagazine.ru – Режим доступа: <http://www.cmsmagazine.ru/library/items/usability/ux-is-not-ui/>
21. Васильев А.Н. Java. Объектно-ориентированное программирование. Учебное пособие. – СПб.: Питер, 2011. – 400 с.
22. Роджерс Р., Ломбардо Д. Android. Разработка приложений [Текст] / Роджерс Р., Ломбардо Д. – М.: ЭКОМ Паблишерз, 2010. – 400 с.
23. Брайн Харді, Білл Філліпс Android. Програмування для професіоналів. - С-Пб .:Пітер, 2016. – 640 с.
24. Кармен Делессіо, Лорен Дерсі, Шейн Кондер Створення додатків для Android за 24 години – М .: Ескмо, 2015. – 528с.
25. Пол Дейтел, Харві Дейтел Android для розробників - С-Пб .: Пітер, 2016. – 512 с.
26. Меднікс З. Програмування під Android / 7. Меднікс З, Дорнін Л, Мік. Б, Накамура М. – Пітер, 2013. – 560 с.
27. Майер Р. Програмування додатків для планшетних комп'ютерів і смартфонів / Майер Р. – Ексмо, 2011. – 671с.
28. What is Mobile Learning? [Электронный ресурс]. – Режим доступа: <http://archive.excellencegateway.org.uk/page.aspx?o=135556>.
29. Adams J. E-learning offer myriad opportunities for rapid talent development [Electronic resource] / Jean Adams // T+D. – 2008. – March. – P. 69-73. – Mode of access : <http://yellowedge.files.wordpress.com/2008/03/adams.pdf>
30. Blended_learning [Electronic resource]. – Mode of access : http://en.wikipedia.org/wiki/Blended_learning.

31. E-learning [Electronic resource]. – Mode of access : <http://en.wikipedia.org/wiki/E-learning>.
32. M-learning [Electronic resource]. – Mode of access : <http://en.wikipedia.org/wiki/M-learning>.
33. Online Nation: Five Years of Growth in Online Learning [Electronic resource]/ I. Elaine Allen and Jeff Seaman. – Solan-C. – 2008. – 26 p. – Mode of access : http://sloanconsortium.org/sites/default/files/online_nation.pdf
34. Tobin L. Get a degree be 'blended learning' [Electronic resource] / Lucy Tobin // The Guardian. – 19 April 2011– Mode of access : <http://www.guardian.co.uk/education/2011/apr/19/distance-blended-learningdegrees>
35. Blended learning – очно-заочное обучение для "продвинутых" [Электронный ресурс] – Режим доступа : <http://www.dw.de/dw/article/0,,4310546,00.html>.
36. Collis B. Flexible learning in a digital world: experiences and expectations / Betty Collis, Jef Moonen. – London : Kogan Page Limited, 2011. – 231 p.
37. Defining eLearning / Performance, Learning, Leadership, & Knowledge Site. [Electronic resource] . – Mode of access : <http://www.nwlink.com/~donclark/hrd/elearning/define.html>.
38. Education World: Educators Battle Over Calculator Use [Electronic resource] // Education World. – Mode of access : http://www.educationworld.com/a_curr/curr072.shtml
39. e-Learning / E-Софт Девелопмент [Электронный ресурс]. – 2011. – Режим доступа : <http://www.web-learn.ru/>
40. Rosenberg M. Beyond E-Learning: New Approaches to Managing and Delivering Organizational Knowledge / Marc J. Rosenberg, Ph. D. // ASTD International Conference – June 3 – Atlanta, 2008.
41. Trenholm S. Long-Term Experiences in Mathematics E-Learning in Europe and the USA / Sven Trenholm, Angel A. Juan, Jorge Simosa, Amilcar Oliveira, Teresa Oliveira // Teaching Mathematics Online: Emergent Technologies and Methodologies – USA: Information Science Reference, 2012. – P.238-257.

42. Шляхтина С. Перспективы развития дистанционного обучения в мире и в России / Светлана Шляхтина [Электронный ресурс] // КомпьютерПресс. – 2006. – № 1. – Режим доступа к журналу : <http://www.compress.ru/article.aspx?id=14659&iid=695>.
43. Jonathan Peppers. Xamarin Cross-platform Application Development Second Edition / Peppers J. – Birmingham : Packt, 2015. – 459 с.
44. Nilanchala Panigrahy. Xamarin Mobile Application Development for Android Second Edition / Panigrahy N. – Birmingham : Packt, 2015. – 296 с.
45. Charles Petzold. Creating Mobile Apps with Xamarin.Forms First Edition / Petzold C. – Redmond : Microsoft Press, 2016. – 1187 с.
46. Best Hybrid Mobile App UI Frameworks: HTML5, CSS and JS – Режим доступа : <http://noeticforce.com/best-hybrid-mobile-app-ui-frameworks-html5-js-css/>. – Дата доступа : 29.03.2016.
47. RoboVM documentation – Режим доступа: <http://docs.robovm.com/>. – Дата доступа : 3.04.2016.
48. Native Mobile Platform Breakdown: A Guide to Xamarin, iOS, and Android – Режим доступа : <http://our.componentone.com/2016/04/12/native-mobileplatform-breakdown-a-guide-to-xamarin-ios-and-android/>. – Дата доступа : 16.04.2016.
49. Building Cross-Platform iOS/Android Apps with Xamarin, Visual Studio, and C# - Part 1 by Jim Wilson – Режим доступа : <https://app.pluralsight.com/library/courses/cross-platform-ios-android-visualstudio-csharp/table-of-contents/>. – Дата доступа : 25.03.2016.

ДОДАТКИ

Додаток А

SUMMARY

Solianyк K. V. Develop a mobile application to prepare for external independent evaluation. – Masters-level Qualification Thesis. Sumy State University, Sumy, 2018.

The article explores the essence of the concept of mobile learning, highlights the main advantages and disadvantages of mobile learning in preparing for external independent evaluation. A comparative analysis of mobile application development methods has been conducted. The main purpose of this study is to develop a mobile application for preparation for external independent evaluation.

Keywords: mobile learning, mobile application, external, android, google play, ionic framework.

Продовження додатку А

АНОТАЦІЯ

Соляник К. В. Розробка мобільного додатку для підготовки до зовнішнього незалежного оцінювання. – Кваліфікаційна магістерська робота. Сумський державний університет, Суми, 2018 р.

У роботі досліджено сутність поняття мобільне навчання, виділені основні переваги та недоліки мобільного навчання при підготовці до зовнішнього незалежного оцінювання. Проведений порівняльний аналіз методів розробки мобільних додатків. Основною метою даного дослідження є розробка мобільного додатку для підготовки до зовнішнього незалежного оцінювання.

Ключові слова: мобільне навчання, мобільний додаток, ЗНО, android, google play, Ionic Framework.

Додаток Б

Лістинг Б.1 - Роутинг додатку – файл Routes.js

```
.config(function ($stateProvider, $urlRouterProvider, $ionicConfigProvider) {
  $stateProvider
    .state('app', {
      url: '/app',
      templateUrl: 'templates/menu.html',
      controller: 'AppCtrl'
    })
    .state('app.main', {
      url: '/main',
      views: {
        'menuContent': {
          templateUrl: 'templates/main.html',
          controller: 'MainCtrl'
        }
      }
    })
    .state('app.subject', {
      url: '/subject',
      views: {
        'menuContent': {
          templateUrl: 'templates/subject.html',
          controller: 'MainCtrl'
        }
      }
    })
  }
}
```

Продовження додатку Б

```
    })
    .state('app.listTheory', {
      url: '/listTheory',
      views: {
        'menuContent': {

          templateUrl: 'templates/listTheory.html',
          controller: 'MainCtrl'
        }
      }
    })
    .state('app.theory', {
      url: '/theory/:theory_ID',
      views: {
        'menuContent': {
          templateUrl: 'templates/theory.html',
          controller: 'MainCtrl'
        }
      }
    })
    .state('app.subListTheory', {
      url: '/subListTheory/:subListTheory_ID',
      views: {
        'menuContent': {
          templateUrl: 'templates/subListTheory.html',
          controller: 'MainCtrl'
        }
      }
    })
    .state('app.subSubListTheory', {
```

Продовження додатку Б

```
url: '/subSubListTheory/:subSubListTheory_ID',
views: {
  'menuContent': {
    templateUrl: 'templates/subSubListTheory.html',
    controller: 'MainCtrl'
  }
}
}))
.state('app.aliases', {
  url: '/aliases',
  views: {
    'menuContent': {
      templateUrl: 'templates/Ukr-lit-aliases.html',
      controller: 'MainCtrl'
    }
  }
}))
.state('app.table_mno', {
  url: '/table_mno',
  views: {
    'menuContent': {
      templateUrl: 'templates/Math_Small_Info/tableMno.html',
      controller: 'MainCtrl'
    }
  }
}))
.state('app.table_step', {
  url: '/table_step',
  views: {
```

Продовження додатку Б

```
'menuContent': {
  templateUrl: 'templates/Math_Small_Info/table_Step.html',
  controller: 'MainCtrl'
}
})
.state('app.table_fact', {
  url: '/table_fact',
  views: {
    'menuContent': {
      templateUrl: 'templates/Math_Small_Info/table_fact.html',
      controller: 'MainCtrl'
    }
  }
})
.state('app.verbs', {
  url: '/verbs',
  views: {
    'menuContent': {
      templateUrl: 'templates/English_Templates_Info/verbs.html',
      controller: 'MainCtrl'
    }
  }
})
.state('app.datesOrPersonal', {
  url: '/datesOrPersonal',
  views: {
    'menuContent': {
      templateUrl: 'templates/datesOrPersonal.html',
```

Продовження додатку Б

```
        controller: 'MainCtrl'
    }
}
}))
.state('app.text_or_paragraph', {
    url: '/text_or_paragraph',
    views: {
        'menuContent': {
            templateUrl: 'templates/textOrParagraph.html',
            controller: 'MainCtrl'
        }
    }
})
.state('app.news', {
    url: '/news',
    views: {
        'menuContent': {
            templateUrl: 'templates/news.html',
            controller: 'NewsCtrl'
        }
    }
})
.state('app.OneNews', {
    url: '/OneNews',
    views: {
        'menuContent': {
            templateUrl: 'templates/OneNews.html',
            controller: 'NewsCtrl'
        }
    }
})
```

Продовження додатку Б

```
    }  
  })  
  .state('app.translate', {  
    url: '/translate',  
    views: {  
      'menuContent': {  
        templateUrl: 'templates/translate.html',  
        controller: 'TranslateCtrl'  
      }  
    }  
  })  
  $urlRouterProvider.otherwise('/app/main');  
});
```


Додаток В

Лістинг В.1 –Функція доступу до теоретичних матеріалів

```
$scope.goToTheory = function (theory) {  
  console.log(theory);  
  switch ($rootScope.titleSubject) {  
    case "Історія України":  
      $location.path('/app/theory/' + theory.number);  
      break;  
    case "Українська мова":  
      if (theory.subtheme == false) {  
        $location.path('/app/theory/' + theory.number);  
      } else {  
        $rootScope.subTheorys = theory.subtheme;  
        $location.path('/app/subListTheory/' + theory.number);  
      }  
      break;  
    case "Українська література":  
      $location.path('/app/theory/' + theory.number);  
      break;  
    case "Географія":  
      if (theory.subtheme == false) {  
        $location.path('/app/theory/' + theory.number);  
      } else {  
        $rootScope.subTheorys = theory.subtheme;  
        $location.path('/app/subListTheory/' + theory.number);  
      }  
      break;  
    case "Математика":
```

Продовження додатку В

```
if (theory.subtheme == false) {
    $location.path('/app/theory/' + theory.number);
} else {
    $rootScope.subTheorys = theory.subtheme;
    $location.path('/app/subListTheory/' + theory.number);
}
break;
case "Англійська":
if (theory.subtheme == false) {
    $location.path('/app/theory/' + theory.number);
} else {
    $rootScope.subTheorys = theory.subtheme;
    $location.path('/app/subListTheory/' + theory.number);
}
break;
case "Біологія":
if (theory.subtheme == false) {
    $location.path('/app/theory/' + theory.number);
} else {
    $rootScope.subTheorys = theory.subtheme;
    $location.path('/app/subListTheory/' + theory.number);
}
break;
case "Фізика":
if (theory.subtheme == false) {
    $location.path('/app/theory/' + theory.number);
} else {
    $rootScope.subTheorys = theory.subtheme;
```

Продовження додатку В

```

    $location.path('/app/subListTheory/' + theory.number);
  }
  break;
case "Хімія":
  if (theory.subtheme == false) {
    $location.path('/app/theory/' + theory.number);
  } else {
    $rootScope.subTheorys = theory.subtheme;
    $location.path('/app/subListTheory/' + theory.number);
  }
  break;
}
};

$scope.initTheoryHTML = function () {
  switch ($rootScope.titleSubject) {
    case "Історія України":
      $scope.link = 'templates/History_Theory/theory_' + $stateParams.theory_ID + '.html';
      break;
    case "Українська мова":
      $scope.link = 'templates/Ukr_Theory/theory_' + $stateParams.theory_ID + '.html';
      break;
    case "Українська література":
      if ($rootScope.titleTheory == "Теорія") {
        $scope.link = 'templates/Ukr_lit_Theory/theory_' + $stateParams.theory_ID + '.html';
      } else if ($rootScope.titleTheory == "Аналіз творів") {
        $scope.link = 'templates/Ukr_Lit_Analyze_Theory/theory_' + $stateParams.theory_ID +
        '.html';
      }
      break;
  }
};

```

Продовження додатку В

```
case "Географія":
    $scope.link = 'templates/Geography_Theory/theory_' + $stateParams.theory_ID + '.html';
    break;
case "Математика":
    if ($rootScope.titleTheory == "Формули") {
        $scope.link = 'templates/Math_Formula/theory_' + $stateParams.theory_ID + '.html';
    } else if ($rootScope.titleTheory == "Теорія алгебра") {
        $scope.link = 'templates/Algebra_Theory/theory_' + $stateParams.theory_ID + '.html';
    } else if ($rootScope.titleTheory == "Теорія геометрія") {
        $scope.link = 'templates/Geometry_Theory/theory_' + $stateParams.theory_ID + '.html';
    }
    break;
case "Англійська":
    if ($rootScope.titleTheory == "Теорія(тексти)") {
        $scope.link = 'templates/English_Theory/theory_' + $stateParams.theory_ID + '.html';
    } else if ($rootScope.titleTheory == "Пишемо правильно") {
        $scope.link = 'templates/English_Theory_Advice/theory_' + $stateParams.theory_ID + '.html';
    } else if ($rootScope.titleTheory == "Времена(рус)") {
        $scope.link = 'templates/English_Time_Theory/theory_' + $stateParams.theory_ID + '.html';
    }
    break;
case "Біологія":
    if ($rootScope.titleTheory == "Теорія") {
        $scope.link = 'templates/Biology_Theory/theory_' + $stateParams.theory_ID + '.html';
    } else if ($rootScope.titleTheory == "Учені України") {
        $scope.link = 'templates/Biologi_Personals_Theory/theory_' + $stateParams.theory_ID +
'.html';
    }
    break;
case "Фізика":
```

Продовження додатку В

```
if ($rootScope.titleTheory == "Теорія") {
    $scope.link = 'templates/Physics_Theory/theory_' + $stateParams.theory_ID + '.html';
} else if ($rootScope.titleTheory == "Учені України") {
    $scope.link = 'templates/Physics_Personals_Theory/theory_' + $stateParams.theory_ID +
'.html';
} else if ($rootScope.titleTheory == "Формули") {
    $scope.link = 'templates/Physics_Formula/theory_' + $stateParams.theory_ID + '.html';
}
break;
case "Хімія":
    if ($rootScope.titleTheory == "Теорія") {
        $scope.link = 'templates/Chemistry_Theory/theory_' + $stateParams.theory_ID + '.html';
    } else if ($rootScope.titleTheory == "Учені України") {
        $scope.link = 'templates/Chemistry_Personals_Theory/theory_' + $stateParams.theory_ID +
'.html';
    } else if ($rootScope.titleTheory == "Корисно") {
        $scope.link = 'templates/Chemistry_Useful_Theory/theory_' + $stateParams.theory_ID +
'.html';
    }
    break;
}
};
```

Додаток Г

Лістинг Г.1 - Контроллер роботи функції «Перекладач»

```

.controller('TranslateCtrl', function ($scope, $http, $rootScope) {
  $scope.textTransl = {};
  $scope.goToTransl = function (length) {
    $http({
      method: 'POST',
      url: 'https://translate.yandex.net/api/v1.5/tr.json/detect?hint=en,uk' +
'&key=trnsl.1.1.20170112T080008Z.fc7ddcca0e07f871.0dbe52c1f67f1593405d58b0fd3338cb6c48
1533&text=' +
      $scope.textTransl.whatLanguage,
      headers: {
        'Content-Type': 'application/x-www-form-urlencoded'
      }
    }).success(function (data) {
      $scope.textTransl.langFrom = data.lang;
      console.log($scope.textTransl.langFrom);
      $scope.textTransl.langIn = 'uk';
      if ($scope.textTransl.langFrom != 'en')
        $scope.textTransl.langIn = 'en'
    }).then(function () {
      $http({
        method: 'POST',
        url: 'https://translate.yandex.net/api/v1.5/tr.json/translate?lang=' + $scope.textTransl.langFrom
+ '-' +
        $scope.textTransl.langIn +
'&key=trnsl.1.1.20170112T080008Z.fc7ddcca0e07f871.0dbe52c1f67f1593405d58b0fd3338cb6c48
1533&text=' +

```

Продовження додатку Г

```
$scope.textTransl.whatLanguage,  
headers: {  
  'Content-Type': 'application/x-www-form-urlencoded'  
}  
}).success(function (data) {  
  $scope.textTransl.successTranslate = data;  
})  
});  
$rootScope.clickEvent("Длина слова:", length);  
};  
$rootScope.initEvent("Перекладач");  
})
```

Додаток Д

Лістинг Д.1 - Контроллер та сервіс взаємодії Firabase та функцією «Новини ЗНО»

```
.factory('news', function ($firebaseArray) {
  return {
    getData: function (url) {
      var messagesRef = new Firebase(url);
      return $firebaseArray(messagesRef);
    }
  };
});

.controller('NewsCtrl', function ($scope, news, $state, $timeout, $rootScope) {
  $rootScope.showLoading();
  $scope.newsArr = news.getData("https://zno-2017.firebaseio.com/");
  console.log($scope.newsArr);

  $scope.goToOneNews = function (news) {
    console.log(news);
    $rootScope.titleNews = news.$id;
    $rootScope.ContentNews = news.$value;
    $state.go("app.OneNews");
  };
  $rootScope.initEvent("Новини");
});
```