

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**Факультет комп'ютерних наук, фізики та математики**  
**Кафедра інформатики, програмної інженерії та економічної**  
**кібернетики**

**РОЗРОБЛЕННЯ МОБІЛЬНОГО НАВЧАЛЬНОГО ДОДАТКУ ДЛЯ**  
**ПІДГОТОВКИ ДО ВСТУПУ НА НАВЧАННЯ НА ДРУГОМУ**  
**(МАГІСТЕРСЬКОМУ) РІВНІ ВИЩОЇ ОСВІТИ**

**Кваліфікаційна робота (проект)**  
на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу 441 групи  
Спеціальності: 121 Інженерія  
програмного забезпечення  
Освітньо-професійної програми:  
Інженерія програмного забезпечення  
Валяєв Кирило Віталійович  
Керівник: кандидат педагогічних  
наук, доцент Вінник Максим  
Олександрович  
Співкерівник: доктор педагогічних  
наук, професор Співаковський  
Олександр Володимирович  
Рецензент: Азадова Елліна  
Валеріївна, DataArt, Senior QA

## ЗМІСТ

<b>ЗМІСТ</b> .....	2
<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ</b> .....	4
<b>ВСТУП</b> .....	5
<b>РОЗДІЛ 1 ПОНЯТТЯ ФРЕЙМВОРКУ. ПОРІВНЯННЯ ФРЕЙМВОРКІВ ДЛЯ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ</b> .....	8
1.1. Поняття фреймворку.....	8
1.2. Порівняння популярних фреймворків для розроблення мобільних додатків .....	10
1.3. Фреймворк React Native.....	13
<b>РОЗДІЛ 2 ПРОЄКТУВАННЯ ДОДАТКУ</b> .....	17
2.1. Розробка концепції мобільного додатку, його прототипу .....	17
2.2. Розробка концепції створення взаємодії між клієнтською та серверною частинами.....	20
<b>РОЗДІЛ 3 РОЗРОБЛЕННЯ ДОДАТКУ</b> .....	24
3.1. Розгортання серверної частини .....	24
3.2. Розгортання клієнтської частини.....	25
3.3. Адміністрування даних в CMS .....	25
<b>ВИСНОВКИ</b> .....	26
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	27
<b>ДОДАТКИ</b> .....	30
<b>ДОДАТОК А.</b> Прототип мобільного додатку .....	30
<b>ДОДАТОК Б.</b> Кодекс академічної доброчесності здобувача вищої освіти херсонського державного університету.....	31
<b>ДОДАТОК В.</b> Діаграми активності .....	32
<b>ДОДАТОК Д.</b> Порівняльна таблиця популярних фреймворків для розробки мобільних додатків .....	35
<b>ДОДАТОК Е.</b> Частка ринку смартфонів, настільних комп’ютерів та планшетів у всьому світі .....	36

<b>ДОДАТОК Ж.</b> Діаграма послідовності взаємодії із додатком.....	37
<b>ДОДАТОК З.</b> Діаграма діяльності .....	38
<b>ДОДАТОК К.</b> Діаграма використання.....	39

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ЗВО	Заклад вищої освіти
COVID	Corona Virus Disease
CMS	Content Management System
API	Application Programming Interface
ЄВІ	Єдиний вступний іспит
ЄФВВ	Єдине фахове вступне випробування
URL	Uniform Resource Locator
UX	User Experience
ОС	Операційна система
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
IT	Інформаційні технології
HTTP	HyperText Transfer Protocol
CSRF	Cross-Site Request Forgery
LTS	Long term support
APK	Android package kit
SQL	System query language

## ВСТУП

На сьогоднішній день навчання в Інтернеті користується неабияким попитом, оскільки таке навчання зручне та доступне. Ця тенденція ще більше прискорилася через зниження здатності організувати навчальний процес традиційним способом, наприклад у класі чи лабораторії, через триваючу глобальну пандемію на «COVID-19». Онлайн-навчання особливо зручне при використанні смартфонів, оскільки ці пристрої сприяють посиленню основних переваг онлайн-навчання - гнучкості та доступності. Користувачу просто потрібно мати доступ до смартфона з підключенням до Інтернету, щоб отримати знання, коли і де вони їм потрібні. Більше того, смартфони з кожним днем набувають все більшої популярності порівняно з планшетами та ноутбуками, задовольняючи більшість потреб сучасного суспільства, що робить їх чудовим вибором для реалізації цього проєкту **[Ошибка! Источник ссылки не найден.]** (див. Додаток Е).

Пристосовуючись до сучасних тенденцій, ЗВО Херсонський державний університет впровадив зміни у навчальний процес, здійснивши перехід від традиційної(стаціонарної) форми навчання на змішану(з елементами дистанційної освіти). В свою чергу це вплинуло і на спосіб проведення підготовчих курсів до вступу на навчання на другому (магістерському) рівні вищої освіти. Оскільки багато механізмів з викладацького арсеналу втратили свою актуальність, через відсутність прямого контакту з учнем, виникла необхідність у створенні допоміжного функціоналу. На прикладі мобільного додатку для вивчення іноземних мов ми надаємо викладачу інструмент забезпечення безперервності навчального процесу, адаптуючи зміст до потреб учнів.

Акцент було зроблено саме на іноземних мовах адже саме вони особливо важливі в наш час, оскільки дозволяють побудувати комунікаційний міст між носієм нових знань та їх одержувачем. Ось

чому такий додаток надає не тільки можливість інтерактивного навчання для користувачів відповідно до сучасних тенденцій, а й досить простий спосіб управління та збагачення бази знань додатку менеджерами вмісту за допомогою інтеграції CMS.

Додаток спрямований на підготовку до ЄВІ з іноземних мов, з можливістю розширити проект, включивши інші навчальні дисципліни для ЄФВВ в майбутньому. Зосередження уваги на мовах також має на меті популяризування ЗВО Херсонський державний університет серед студентів різного мовного походження.

Актуальність роботи обумовлена необхідністю розробки мобільного додатку, який буде допоміжним інструментом викладача, під час дистанційної форми навчання, що забезпечить неперервність освітнього процесу під час проведення підготовчих курсів у ЗВО Херсонському державному університеті.

Метою роботи є проектування, створення структури та розроблення мобільного додатку для навчання, доступного на телефонах Android, що складається зі сторони сервера, клієнта та API, який їх поєднує.

Об'єктом роботи є навчальний мобільний застосунок.

Предметом роботи є розробка навчального мобільного застосунку

Завдання роботи:

1. Проаналізувати існуючі фреймворки для розробки мобільних додатків та виділити ті, що відповідають вимогам проекту;
2. Розробити клієнтську частину мобільного додатку, яка відповідає сучасним стандартам та вимогам;
3. Інтегрувати клієнтську частину із серверною частиною мобільного додатку за допомогою API;
4. Реалізувати функціонал повторюваного навчання;
5. Реалізувати багатомовний інтерфейс користувача.

У процесі роботи використовувалися такі методи дослідження як аналіз, синтез та порівняння.

Структура роботи: робота складається з переліку умовних позначень, вступу, трьох розділів, висновків, списку використаних джерел та додатків.

# РОЗДІЛ 1

## ПОНЯТТЯ ФРЕЙМВОРКУ. ПОРІВНЯННЯ ФРЕЙМВОРКІВ ДЛЯ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

### 1.1. Поняття фреймворку

Фреймворк (Framework, структура, платформа) – це програмна платформа, що визначає будову програмної та допомагає полегшити розробку систем [2]. Такі платформи можуть включати в себе компілятори, бібліотеки з готовими рішеннями, інтерфейси створення додатків(API), які поєднують різні компоненти, що надає можливість на якісну та швидку розробку.

Головною з переваг фреймворку є те, що він підтримує стандартизований вигляд програми, а саме її структуру. Так об'єктно-орієнтовані фреймворки складаються з класів (визначених та абстрактних) та інтерфейсів [3]. Тому програмісту достатньо використати вже існуючу реалізацію певної задачі чи додати певний функціонал для вирішення проблеми за допомогою своїх власних класів. Більшість фреймворків націлені на розв'язання конкретних завдань. Фреймворки визначаються певними особливостями:

- розширюваність - розробник має можливість розширити структуру, наприклад, додавши певну функціональність до вже існуючих компонентів;
- кросплатформність (багатолатформність) – властивість програмного забезпечення працювати на декількох програмних чи апаратних платформах;
  - універсальний інтерфейс для програміста(API) та користувача;
  - повторно використовуваний код;
  - стійкість коду.

Багатолатформні фреймворки також можна поділити на дві групи:



- гібридно-нативну розробку;
- гібридну веб-розробку;

Гібридно-нативний підхід об'єднує фреймворки з нативним інтерфейсом користувача і загальним кодом; і фреймворки із загальною кодовою базою і нативним кодом. Серед них: кроссплатформені платформи розробки додатків Flutter, React Native, NativeScript і Xamarin.

Гібридна веб-розробка кроссплатформенних додатків здійснюється на базі фреймворків з веб-інтерфейсом і загальними компонентами та з єдиною кодовою базою, яка працює, де завгодно. Прикладом такого кроссплатформенного фреймворка є Ionic.

Основні переваги гібридного підходу:

- Багатоплатформність – створивши один додаток ви зможете експортувати його до будь-якої ОС;
- Швидкість розробки та масштабування;
- Низька вартість розробки та підтримки продукту(у порівнянні з нативним підходом);
- Використання однієї мови;
- Однаковий інтерфейс та UX;

Основні недоліки гібридного підходу:

- Більш низька продуктивність;
- Менше можливостей інтеграції з особливостями пристрою;
- Робота з уніфікованим стеком технологій не дасть тієї гнучкості налаштування і оптимізації, яка є у кожної ОС зі своїм власним стеком технологій;
- Можливі труднощі при публікації продукту до магазинів(Apple App Store та Google Play Store), через розбіжність у вимогах;
- Обмежена реалізація візуальних і графічних елементів в додатках, особливо анімації.

## 1.2. Порівняння популярних фреймворків для розроблення мобільних додатків

Згідно до статистики, опублікованій на веб-ресурсі Stackoverflow, найбільш запитуваними та найулюбленішими фреймворками серед розробників мобільних додатків є наступні: Flutter, React Native, Xamarin та Ionic [4,5](див. рис. 1.1.).

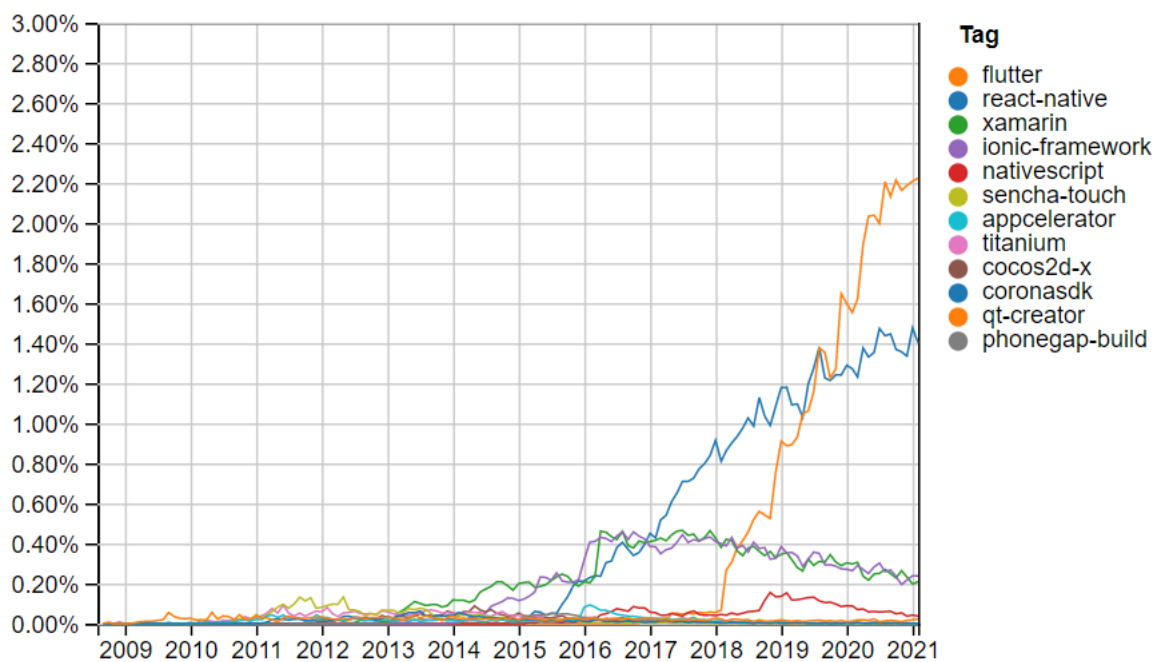


Рисунок 1.1 – Статистика запитів стосовно кожного фреймворка

Розглянемо більш детально декількох з них, а саме React Native, Flutter та Ionic(див. Додаток Д). Flutter пропонує безліч елементів інтерфейсу, які виглядають нативними для ОС, але за своєю сутністю не є такими. Цей фреймворк може підійти для створення уніфікованого UX, і він утримується іншого підходу, ніж React Native. Flutter не перетворює вихідний код в нативний, який виконується платформою. Фактично він малює вікно на екрані телефону і виводить всі елементи сам. Основними плюсами цього фреймворку є:

- Мова Dart - ця мова надає відмінні можливості для розробки кроссплатформених мобільних додатків. Крім того, Dart був створений Google як розширена версія JavaScript.
- Легкість навчання - є чітка документація і стандартні шаблони проектування.
- Висока швидкість - додатки Flutter компілюються в машинний код з використанням графіки і движків рендерингу, вбудованих в C / C ++. Додатки написані на ньому швидко і надійні. У сфері кроссплатформених технологій це дає очевидну перевагу Flutter перед конкурентами.

Проте є й мінуси:

- Нестабільний, «сирий» API - зміна специфікації може відбуватися досить часто і неочікувано.
- Не працюють деякі нативні речі, які повинні працювати з коробки.
- Віджети - повинні працювати в парі з основним додатком, а піднімати дартівський додаток в тлі для віджета - це відразу велика витрата пам'яті.

Ionic - це фреймворк типу Hybrid-Web, де створений ним додаток виконуються в спеціальній оболонці (UIWebView для iOS і WebView для Android), яка дозволяє показувати HTML і виконувати JavaScript на телефоні. Виходить, додаток працює немов би в веб-браузері. У Ionic велика вбудована бібліотека інструментів та велика кількість плагінів і модулів, які забезпечують доступ до нативних функцій. Переваги цього фреймворка:

- Використовується інструментарій, добре відомий всім розробникам.
- Розробка на фреймворку Ionic відбувається набагато швидше, ніж на інших. Він заснований на Angular, а це значить, що під

час розробки додаток можна запускати в браузері і дивитися, як він виглядає.

- Є готові шаблони додатків і вбудовані пресети, що допомагає швидко та без зусиль почати розробку.

З мінусів:

- за відгуками розробників тестування програми може бути досить складним. Протестувати вбудовані функції пристрою (наприклад, камера, вібрація, миготіння), які є в документації Ionic, не можна, адже ці функції просто не можуть бути виконані в браузері;
- різні нативні особливості складно поєднати між собою;
- додатки на Ionic працюватимуть погано на старих пристроях Android: щоб запуститися додаток на Ionic буде використовувати браузер, встановлений за замовчуванням на пристрої. Пристрої зі старими версіями Android (4.0-4.3) за замовчуванням використовують браузер Android, а не Google Chrome;
- не підходить для роботи зі складною графікою.

Ionic легко інтегрується з іншими бібліотеками, наприклад, Angular і React, вмiє працювати з Bluetooth, Health Kit і перевіркою достовірності за відбитком пальця. Також реалізовано інтерфейс командного рядка (CLI), що забезпечує необхідну продуктивність при використанні емуляторів мобільних пристроїв, перезавантаження додатки в реальному часі і ведення журналу. Щоб отримати доступ до камери, GPS або ліхтарика, Ionic використовує плагіни Cordova і Capacitor.

### 1.3. Фреймворк React Native

Основними вимогами до розробки такого додатку, враховуючи усі чинники, є висока продуктивність, багатоплатформність, стабільність, швидкість та легкість розробки.

Кожний з наведених фреймворків, що були описані в розділі 1.2, надають необхідний функціонал, для розробки, проте певні особливості не відповідають нашим критеріям. За результатами порівняння цих фреймворків (див. Додаток Д) можна відмітити саме React Native, який найбільше підходить для забезпечення неперервного освітнього процесу.

React Native використовує вбудовані компоненти інтерфейсу і API-інтерфейси. При роботі з цим фреймворком вам необхідно зробити збірку своїх власних елементів управління, побудувати ієрархію з урахуванням інтеграції і розробити для користувача інтерфейс на мові React Native. Додаток буде працювати на всіх платформах, оскільки майже весь код написаний на JavaScript, і цей код взаємодіє з нативними компонентами ОС. Кросплатформність, плагіни з відкритим вихідним кодом і простота розробки - все це спрощує завдання і скорочує час.

Додатки, створені за допомогою React Native, судячи з поведінки і зовнішнього вигляду близькі до нативних. Він ідеально підходить, у випадках коли потрібна швидкість нативного додатка, але не потрібна складність [7].

Розробник не повинен витратити зайвий час, щоб створити мобільний додаток, оскільки потрібно якнайменш розробити два додатки для підтримки iOS та Android. Оскільки різні платформи мають різний зовнішній вигляд, сприйняття та можливості, програма не може існувати однорідно для всіх операційних систем. Однак, оскільки графічний інтерфейс відрізняється, розробка може базуватися на одній мові, але може мати графіку, що відображатиметься по-різному залежно від цільової.

Основний принцип, закладений у React Native - це те, що додаток складається з окремих компонентів, які можна замінити або оновити самостійно, не оновлюючи всю сторінку. Це економить пропускну здатність. Зовнішні файли не завантажуються кожного разу при повторному завантаженні сторінки (SPA)[8] (див. рис.1.2).

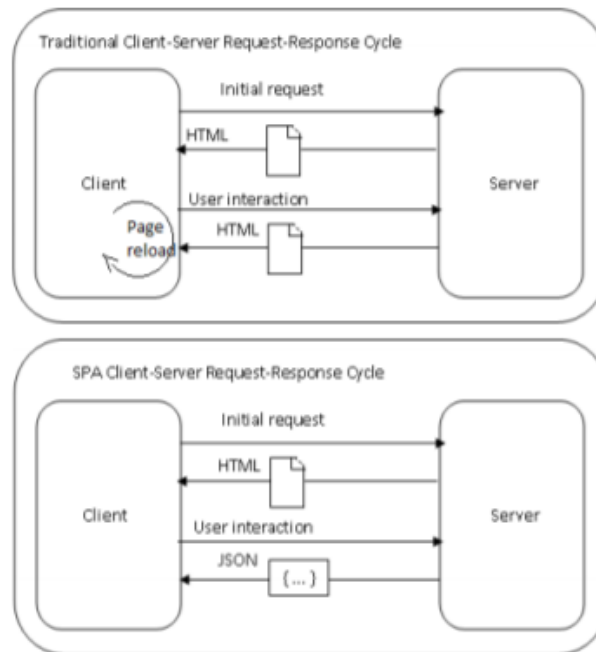


Fig. 1 Client-Server Request-Response Cycle

Рисунок 1.2 – Приклад «Цикл запит-відповідь SPA »

React Native працює у вбудованому екземплярі JavaScriptCore (iOS) або V8 (Android) всередині додатків і відображається на компонентах вищого рівня для платформи. Компоненти JavaScript оголошуються за допомогою набору вбудованих примітивів, що підтримуються компонентами iOS або Android.

React Native взаємодіє з екраном, впроваджуючи систему, яка обробляє дотики у складному вигляді з високимірнівними функціями. Оскільки розпізнавання жестів на мобільному пристрої є більш вдосконалим, ніж в Інтернеті, існує багато різних методів, які можна інтерпретувати на дотик, таких як скрол, натискання, ковзання тощо.

Більше того, одночасно можна виконувати кілька дотиків. Відчутна різниця між веб-програмою та нативною програмою, оскільки кожен дотик повинен відображати, що відбуватиметься після його виконання, і користувач повинен мати можливість скасувати дію, перетягуючи палець. React Native вирішив цю проблему, додавши абстрактну реалізацію `Touchable` та `TouchableHighlight`, яка належним чином асимілює списки та інші елементи, не вимагаючи додаткової конфігурації.

Крім того, подібно до React, React Native використовує JSX [9] та вбудований стиль для створення та стилювання компонентів без ускладнення структури або коду. У React стиль визначається як об'єкт і додається до елемента всередині тегу HTML(див. рис. 1.3).

```
var divStyle = {color: blue};

React.render(
  <h1 style={divStyle}>Hello, world!</h1>,
  document.getElementById('myDiv')
);
```

Рисунок 1.3 – Приклад додавання стилю до блоку розмітки

Однак React Native використовує веб-модель макета під назвою Flexbox. Flexbox спрощує процес побудови загальних макетів інтерфейсу користувача, а його абстракція `StyleSheet` пропонує можливість оголосити макет та стиль разом із вбудованим компонентом. Використання стилів для Flexbox видно на рисунку(див. рис 1.4).

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5FCFF',
  },
  nameText: {
    fontSize: 20,
    margin: 10,
  },
  bornText: {
    color: '#333333',
    marginBottom: 5,
  },
  image: {
    width: 53,
    height: 81,
  },
});
```

Рисунок 1.4 – Приклад «Об'єкт JSON, що містить стиль компонентів Flexbox»

React Native має підтримку загальних API платформи та нативних компонентів інтерфейсу, а також підтримку для створення власних власних модулів та подань, якщо це потрібно.

Тож, беручи до уваги основну характеристику та особливості фреймворку - React Native є вдалим вибором для використання.



## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ДОДАТКУ

#### 2.1. Розробка концепції мобільного додатку, його прототипу

Структура проекту повинна складатися з кількох взаємопов'язаних частин. Клієнтська сторона підключається до серверної через API. Клієнтська сторона надсилає запити на сервер, який їх обробляє. Після обробки запиту сервер надсилає відповідь клієнтській стороні. Серверна сторона використовує базу даних для зберігання вмісту, керованого в системі управління вмістом, та користувацького прогресу, створеного програмно(див. рис. 2.1).

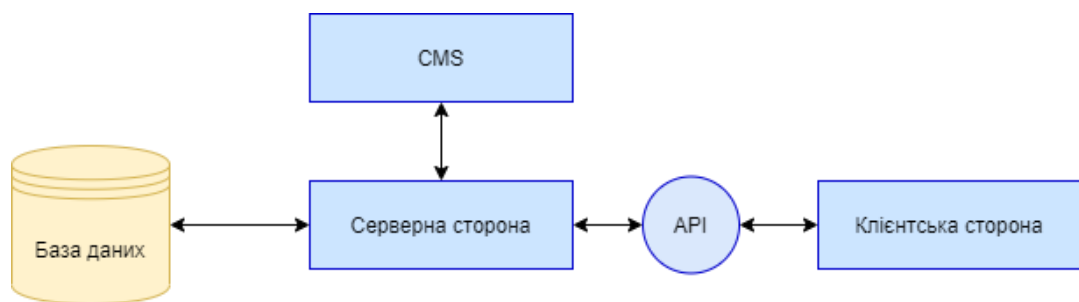


Рисунок 2.1 – Логічна структура додатку

Для менеджменту даних уповноважені особи – контент менеджери, мають бути зареєстровані та аутентифіковані в системі. Для управління усіма навчальними матеріалами є необхідним такий функціонал як:

- зручне додавання, редагування та усунення даних;
- система авторизації користувачів;

Для того, щоб створити конкурентну програму, яка задовольняла би потреби користувача, треба реалізувати наступний функціонал:

- авторизація користувача:
  - реєстрація, log in, log out;

- вправи на вивчення слів та фраз:
  - прочитати слово/фразу та вибрати переклад із списку;
  - прочитати переклад слова/фрази і написати його самостійно;
  - прослухайте слово/фразу та дайте відповідь, якщо воно правильно перекладено за свайпом вліво/вправо;
- профіль користувача;
- статистика користувача;
- перегляд для вивчених слів/фраз (терміни «слово» та «фраз» використовуються як взаємозамінні):
  - слова переглядаються після першого навчання;
  - кожен перегляд складається зі слів залежно від того, наскільки успішно ми взаємодіємо (взаємодія означає виконання вправи на тренінгу чи огляді) зі словами;
  - успішна взаємодія збільшує час, необхідний для того, щоб слово з'явилося в оглядах;
  - невдала взаємодія зменшує час, необхідний для появи слова в перегляді;
  - перегляди складаються з однієї вправи на слово; вправа визначається тим, яка з трьох вправ спричинила найбільш невдалі взаємодії; якщо кількість невдалих взаємодій однакова для двох або більше фраз, вправа вибирається випадковим чином;
- управління вмістом:
  - створення, редагування, видалення, приховування, публікація вмісту за допомогою веб-інтерфейсу;
  - імпорт фраз із файлів csv;
- багатомовний інтерфейс користувача;
- навігація між екранами на стороні клієнта додатку;

- реалізований onboarding для першої взаємодії з користувачем після встановлення;
- пристосованість компонентів програми до різної роздільної здатності екрану смартфона;

Інтерфейс, в даний час, є одним з найважливіших засобів комунікації між людиною - користувачем та програмно - апаратними компонентами комп'ютерної системи, що представляє собою сукупність засобів і методів, основним призначенням якої є забезпечення передачі інформації, як від системи до користувача, так і навпаки.

Важливою частиною при створенні клієнтської частини є розробка дизайн макетів, для того, щоб визначити набір елементів, загальний стиль додатку, вибрати або створити відповідні шрифти, набір іконок тощо (див. рис. 2.2, Додаток В). Враховуючи той факт, що користувач може проводити багато часу в додатку, було вирішено створити інтерфейс у темних кольорах, щоб зменшити навантаження на очі. Для поліпшення інтуїтивно зрозумілої візуальної частини та користувальницького досвіду була використана готова бібліотека значків `response-native-vector-icons`, яка складається з найпопулярніших наборів піктограм. Малюнок для головного екрану був виготовлений в додатку Adobe Illustrator, який забезпечує можливості створення високоякісних векторних зображень. Крім того, всі елементи екранів є адаптивними, що дозволяє користувачеві правильно і точно відобразити створений вигляд екранів на будь-яких смартфонах.

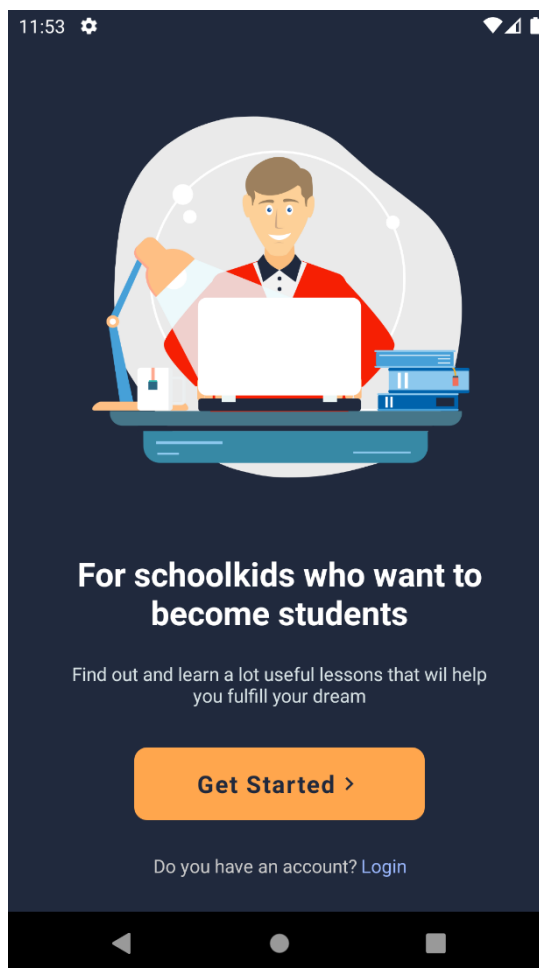


Рисунок 2.2 – Приклад «Прототип головного екрану мобільного додатку»

Макети екранів мобільного додатку були створені за допомогою інструментів веб-додатку [moqups.com](https://www.moqups.com).

## **2.2. Розробка концепції створення взаємодії між клієнтською та серверною частинами**

Використання мобільного додатку у дистанційній освіті має покращити та доповнити рівень підготовчих курсів під час змішаної або дистанційної форми навчання. Викладачу для формування матеріалів підготовчого курсу та обміну зі студентом достатньо додати матеріали до бази знань.

Веб-архітектура додатку забезпечить повну кросплатформеність системи, що у свою чергу забезпечить можливість використання додатку на різних пристроях, використовуючи тільки веб браузер або смартфон(в залежності від ролі особи) за умови доступу до мережі Інтернет.

Структура додатку складається з компоненту авторизації користувачів, надання їм прав до впровадження, редагування та усунування даних та власне надання доступу до функціоналу мобільного навчального додатку.

Мобільний додаток взаємодіє з серверною частиною за допомогою API, яка в свою чергу взаємодіє з CMS (див. Додаток Г).

Також, на етапі створення навчального мобільного додатку важливо відображати процес будування додатку, його тестування та створення можливості перегляду. У цьому нам допоможе Expo - фреймворк та платформа для універсальних додатків React [10]. Це набір інструментів та служб, побудованих навколо React Native та власних платформ, які допомагають розробляти, будувати, розгортати та швидко переглядати iOS, Android та веб-програми з тієї самої кодової бази JavaScript/TypeScript. У порівнянні з іншими інструментами, які можуть надати нам такі послуги(Android Studio, Xcode), за допомогою Expo можна відтворювати створений додаток не тільки на одному певному мобільному пристрої, зі своєю операційною системою(IOS або Android), але й на будь-якому, що має одну з цих операційних систем. Для початку роботи з цим фреймворком достатньо виконати команди в терміналі на рис. 2.3.

```
# Install the command line tools
$ npm install --global expo-cli

# Create a new project
$ expo init my-project
```

Рисунок 2.3 – Стартові команди для роботи з Expo

Не менш важливою частиною системи є база даних. База даних для управління вмістом надається eZ Platform and Doctrine. eZ Platform - це система управління вмістом з відкритим кодом та платформа цифрового досвіду, розроблена компанією Ibexa. eZplatform базується на фреймворці Symfony. Doctrine - об'єктно-реляційний картограф, що дозволяє писати запити до бази даних мовою запитів доктрини (DQL), об'єктно-орієнтованим діалектом SQL.

MySQL Workbench - інструмент візуального проектування баз даних; був використаний для тестування бази даних. Реляційне відображення об'єктів використовується для роботи з базою даних, тому замість терміна «таблиця» використовується «об'єкт».

Об'єктами у дереві вмісту є:

- Базова мова - мова, яку користувач використовуватиме для вивчення іншої,
- мова - мова, яку збирається вивчити користувач
- словник - групує слова за темою, рівнем тощо залежно від уподобань адміністратора.
- фраза - містить дані про слово чи словосполучення

Об'єктами, що використовуються для збереження прогресу користувача, є:

- Прогрес слова - містить інформацію про те, як користувач взаємодіє зі словами
- Streak - містить інформацію про щоденну частоту використання додатка користувачем

Жоден адміністратор не може побачити пароль користувача. Він захищений маркером CSRF.

Дані користувача, такі як обрана мова програми або вбудований ключ, зберігаються на клієнтській стороні в AsyncStorage - асинхронній

системі постійного зберігання ключів і значень. Це дозволяє раціонально спілкуватися між клієнтом та стороною сервера.

## РОЗДІЛ 3

### РОЗРОБЛЕННЯ ДОДАТКУ

#### 3.1. Розгортання серверної частини

Для того щоб ввести в експлуатацію серверну частину проєкту необхідна наявність декількох чинників. Оскільки архітектура додатку – веб, для неперервної та якісної роботи потрібне стабільне підключення до мережі Інтернет. Фундаментом для розгортання було обрано ОС Linux, а саме Ubuntu 18.4 LTS - чиста інсталяція. Після встановлення якої потрібно виконати наступні кроки:

1. Покласти архів з серверною частиною коду та 2 підготовлених інсталяційних скрипта до директорії /home (див. рис. 3.1).
2. Після цих кроків інформація про підключення з'явиться на вашому терміналі; використовуйте перше посилання з першого рядка «переадресації», щоб встановити зв'язок між клієнтом та стороною сервера.
3. Щоб отримати посилання, яке необхідно для комунікації між клієнтською та серверною частинами, як на кроці 3 після перезапуску Ubuntu, виконайте команди, вказані на рис. 3.2

```
chmod +x installation_script_1 && chmod +x  
installation_script_2;  
./installation_script_1;  
./installation_script_2;
```

Рисунок 3.1 – Команди в терміналі для перенесення файлів до директорії

```
cd app;  
ez start;  
./ngrok http 42080;
```

Рисунок 3.2 – Команди в терміналі для отримання URL



### 3.2. Розгортання клієнтської частини

Для того щоб ввести в експлуатацію клієнтську частину проекту потрібно виконати наступні кроки:

1. Завантажте та встановіть сгенерований APK файл додатку на свій смартфон.
2. Припустимо, що наш проект не має постійного домену, тому після отримання посилання в кінці внутрішньої установки(див. попередній розділ) ви повинні вставити це посилання в поле, яке з'явиться після 5 натискань на зображення на першому екрані.

### 3.3. Адміністрування даних в CMS

Для того, щоб створити об'єкт у дереві вмісту:

Перейдіть до /admin:

1. Використовуйте ці облікові дані для входу як адміністратор:  
login: admin; пароль: publish
2. Перейдіть до вмісту → структура вмісту → дерево вмісту
3. У дереві вмісту виберіть місце, в якому ви хочете створити об'єкт

Об'єкти повинні розміщуватися за такою схемою, припускаючи, що об'єкт праворуч містить найближчий об'єкт ліворуч:

1. Коренева папка → Базова мова → Мова → Словник → Фраза
2. Натисніть «створити» та виберіть у меню об'єкт, який потрібно створити

Введіть усі необхідні поля (об'єкт неможливо створити без введення обов'язкових полів; неможливо створити фразу без аудіо; якщо ви створите фразу без заповнення зображення, буде використано placeholder за замовчуванням) і натисніть «опублікувати».

## ВИСНОВКИ

На першому етапі даної роботи було досліджено та порівняно фреймворки для мобільної розробки та виділено критерії оцінювання їх придатності для забезпечення підготовчих курсів за умов змішаної або дистанційної форми освіти. Визначено такий перелік критеріїв, за допомогою яких можна досягнути необхідної функціональності, а саме:

- зручне додавання, редагування та усунення даних;
- багатомовність;
- багатоплатформність;
- інтерактивність курсів;
- система авторизації користувачів;

В процесі аналізу було відкинуто та відібрано різні варіанти, що дозволило обрати фреймворк React Native як основу для реалізації мобільного додатку.

З-поміж проаналізованих було обрано CMS, а саме eZ Platform, для менеджменту змісту, який дає змогу впроваджувати та редагувати базу знань мобільного додатку, що забезпечить можливість передачі даних від викладача до студента та проведення підготовчих курсів дистанційно.

На другому етапі було розроблено концепцію мобільного додатку, а також спроектовано структуру проєкту, прототипи екранів мобільного додатку та описано взаємодію між клієнтською частиною та серверною частиною.

У результаті роботи було створено повноцінний, конкурентноспроможний мобільний навчальний додаток разом з усіма невід'ємними його частинами, а саме клієнтською частиною, серверною частиною, API та впровадженою CMS.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. StatCounter Global Stats: Desktop vs Mobile vs Tablet Market Share Worldwide December 2018 – December 2020: Statcounter (2021) [Електронний ресурс]. URL: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#monthly-201812-202012>.
2. Review of framework terminology [Електронний ресурс]. URL: <https://riehle.org/computer-science/research/dissertation/chapter-2.html>
3. Фреймворки у веб-розробці. [Електронний ресурс]. URL: [https://web-creator.ru/articles/about\\_frameworks](https://web-creator.ru/articles/about_frameworks).
4. Stackoverflow. Stackoverflow Trends. [Електронний ресурс]. URL: <https://insights.stackoverflow.com/trends?tags=react-native%2Cflutter%2Camarin%2Cnativescript%2Cionic-framework%2Cunity3d%2Ccocos2d-x%2Ctitanium%2Cphonegap-build%2Csencha-touch%2Cappcelerator%2Ccordova%2Ccoronasdk%2Cqt-creator>
5. Stackoverflow. Most Loved, Dreaded, and Wanted Other Frameworks, Libraries, and Tools. [Електронний ресурс]. URL: [https://insights.stackoverflow.com/survey/2019#technology-\\_most-loved-dreaded-and-wanted-other-frameworks-libraries-and-tools](https://insights.stackoverflow.com/survey/2019#technology-_most-loved-dreaded-and-wanted-other-frameworks-libraries-and-tools)
6. Cross platform mobile application development: a comparison study of React Native Vs Flutter. [Електронний ресурс]. URL: <https://jyx.jyu.fi/handle/123456789/70969#>
7. React Native application development – A comparison between native Android and React Native [Електронний ресурс]. URL: <https://www.diva-portal.org/smash/get/diva2:998793/FULLTEXT02> .
8. Single Page Web Applications: JavaScript end-to-end. September 2013 ISBN: 978-1-61729-075-6

9. Introduction JSX [Электронный ресурс]. URL: <https://facebook.github.io/react/docs/introducing-jsx.html>
10. Framework Expo [Электронный ресурс]. URL: <https://docs.expo.io/>
11. Bonnie Eisenman. Learning React Native. O'Reilly Media, Inc., Dec. 2015. ISBN: 9781491929049.
12. Bonnie Eisenman. Writing Cross-Platform Apps with React Native [Электронный ресурс]. URL: <http://www.infoq.com/articles/react-native-introduction>.
13. React Native documentation. URL: <https://reactnative.dev/docs/intro-react>
14. eZ Platform (by Ibexa) documentation. URL: <https://doc.ibexa.co/en/latest/>
15. Symfony documentation. URL: <https://symfony.com/doc/current/index.html>
16. Content Management System (CMS). URL: <https://kinsta.com/knowledgebase/content-management-system/>
17. React Native application development – A comparison between native Android and React Native. URL: <https://www.diva-portal.org/smash/get/diva2:998793/FULLTEXT02>
18. Effects on performance and usability for cross-platform application development using React Native. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A946127&dswid=8947>
19. Learning React Native: Building Native Mobile Apps with JavaScript. URL: [https://books.google.com.ua/books?hl=ru&lr=&id=274fCwAAQBAJ&oi=fnd&pg=PR2&dq=react+native+&ots=tGqo8GkaiY&sig=xJW9-mGY5xjNaZ0ZHzeDzfO8c\\_g&redir\\_esc=y](https://books.google.com.ua/books?hl=ru&lr=&id=274fCwAAQBAJ&oi=fnd&pg=PR2&dq=react+native+&ots=tGqo8GkaiY&sig=xJW9-mGY5xjNaZ0ZHzeDzfO8c_g&redir_esc=y)
20. Learning JavaScript: JavaScript Essentials for Modern Application Development. URL: <https://books.google.com.ua/books?hl=ru&lr=&id=JUOWCwAAQBAJ&oi=fnd&pg=PP1&dq=Learn+javascript&ots=cJRS0wkVrh&sig=1k82cUlppskjDo>

[2AkdFdqHhWwT8&redir\\_esc=y#v=onepage&q=Learn%20javascript&f=false](#)



# ДОДАТОК Б. КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Валяев Кирило Віталійович, учасник освітнього процесу Херсонського державного університету, УСВІДОМЛЮЮ, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

## **ЗАВЛЯЮ, що у своїй освітній і науковій діяльності ЗОБОВ'ЯЗУЮСЯ:**

– дотримуватися:

- вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
- принципів та правил академічної доброчесності;
- нульової толерантності до академічного плагіату;
- моральних норм та правил етичної поведінки;
- толерантного ставлення до інших;
- дотримуватися високого рівня культури спілкування;

– надавати згоду на:

- безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
- оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
- використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;

– самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;

– надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;

– не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;

– своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;

– не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;

– підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;

– поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;

– не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;

– відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;

– запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;

– не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;

– не підроблювати документи;

– не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;

– не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;

– не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;

– не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;

– не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;

– не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;

– не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

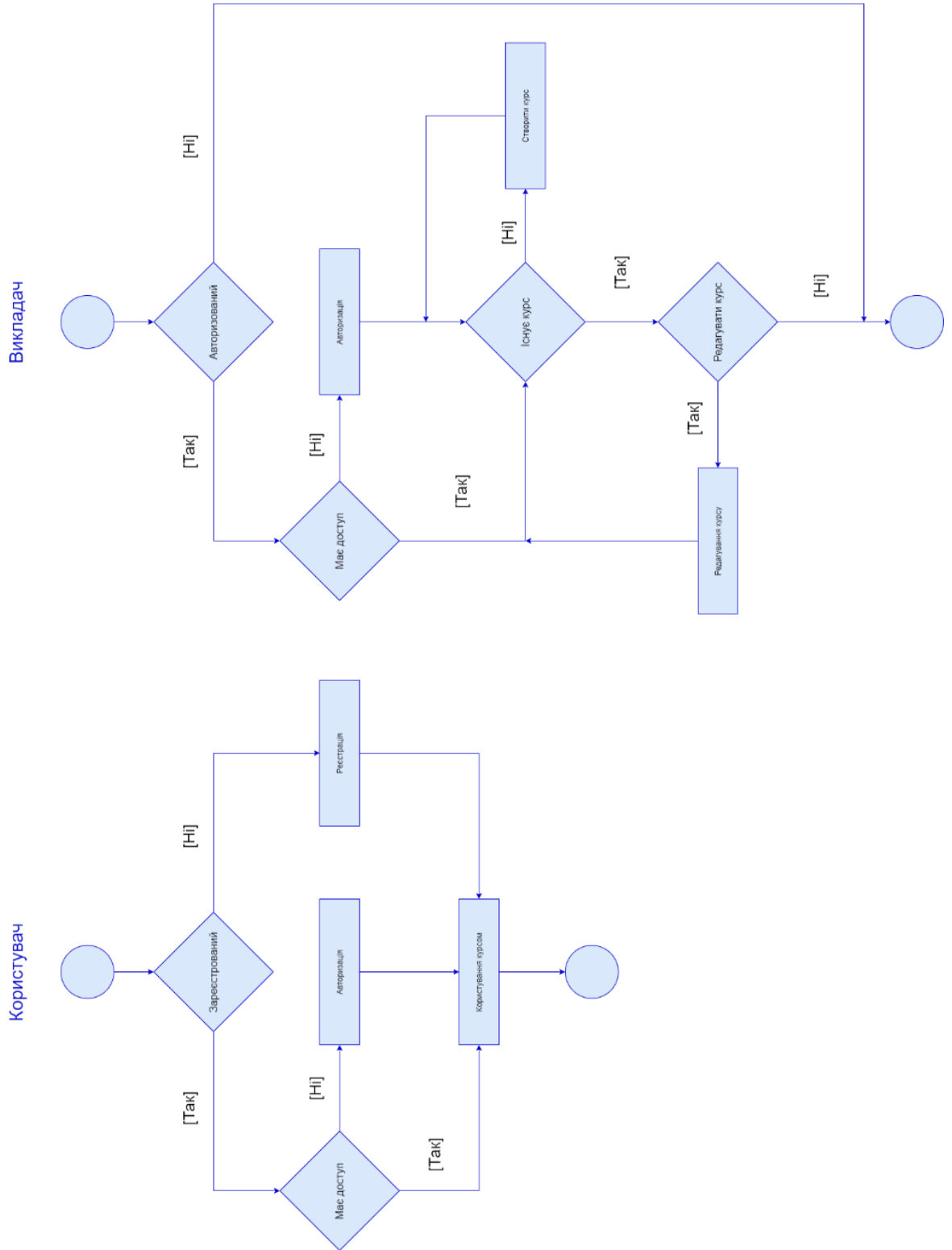
**УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.**

09.04.2021  
(дата)

  
(підпис)

Валяев Кирило  
(ім'я, прізвище)

## ДОДАТОК В. ДІАГРАМИ АКТИВНОСТІ





## Create user

**Endpoint:** /register\_user

**Headers:**

Content-Type: application/json

**Description:** creates a new User

**Example body:**

```
{
  "userName": "username19",
  "password": "Aa123412",
  "currentLanguage": 350
}
```

## Create session (login a user)

**Endpoint:** /api/ezp/v2/user/sessions

**Headers:**

Accept: application/vnd.ez.api.Session+json;

Content-Type: application/vnd.ez.api.SessionInput+json;

Cookie: string;

X-CSRF-Token: string;

**Description:** Performs a login for the user or checks if session exists and returns the session and session cookie. The client will need to remember both session name/ID and CSRF token as this is for security reasons not exposed via GET.

**Example body:**

```
{
  "SessionInput": {
    "login": "admin",
    "password": "secret"
  }
}
```

**Example response:**

```
{
  "Session": {
    "name": "eZSSID",
    "identifier": "go327ij2cirpo59pb6rrv2a4e12",
    "csrfToken": "231kneri34ijajedfw39orj3j93",
    "User": {
      "_href": "/user/users/14",
      "_media-type": "application/vnd.ez.api.User+json"
    }
  }
}
```

## Delete session (logout a user)

**Endpoint:** /api/ezp/v2/user/sessions/{sessionId}

**Description:** The user session is removed i.e. the user is logged out.

Headers:

```
Cookie: string
X-CSRF-Token: string
```

## Learning process

**Endpoint:** /dictionaries/training

Headers:

```
Cookie: string
```

Description: Lists up to five words for learning which belong to dictionaries of a language selected by user

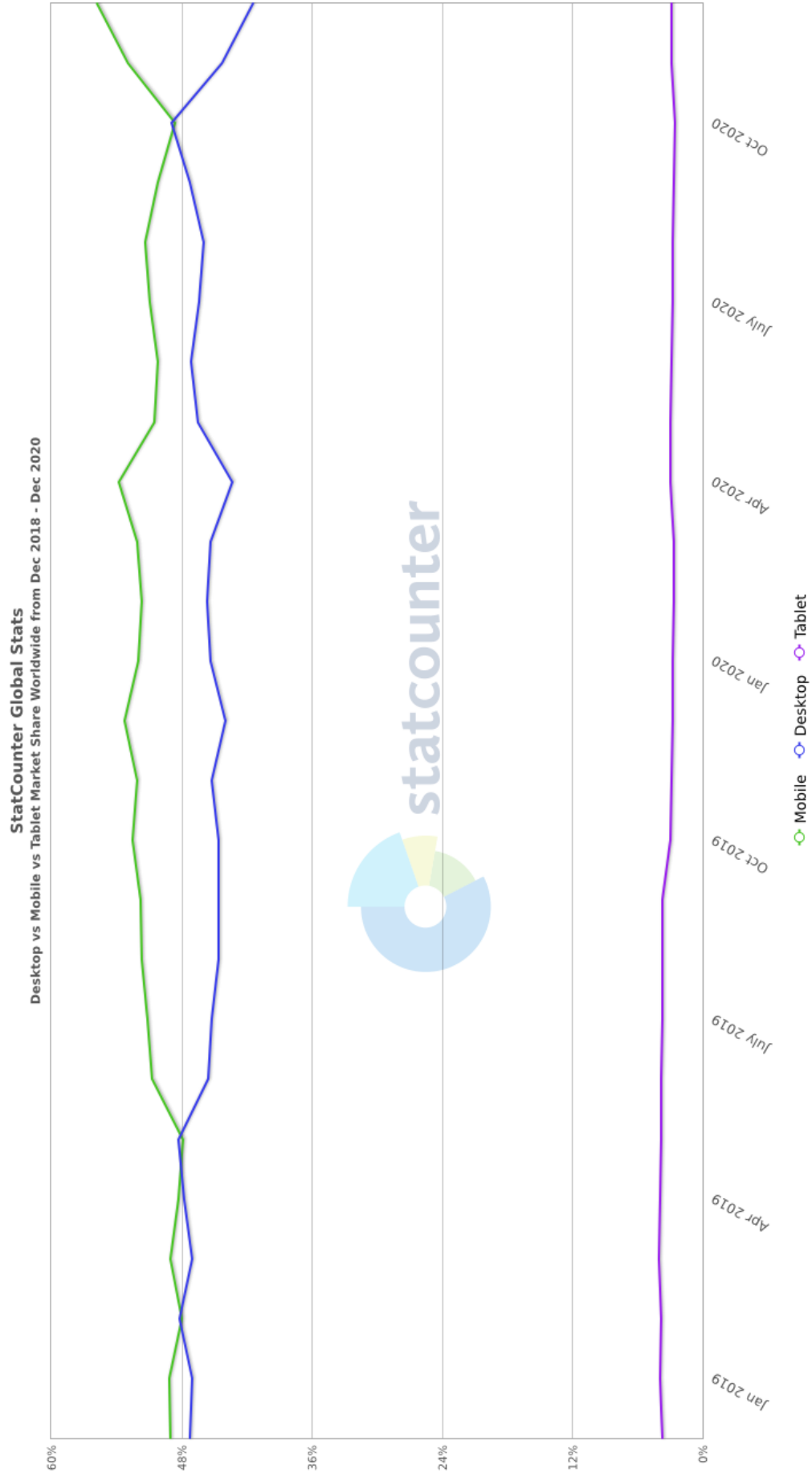
**Example response:**

```
[
  {
    "id": 485,
    "word": "Dom",
    "translation": "House",
    "fakeTranslations": [
      {
        "id": 63,
        "name": "Jędzyk"
      },
      {
        "id": 60,
        "name": "Dom"
      },
      {
        "id": 485,
        "name": "House"
      },
      {
        "id": 326,
        "name": "Goodbye!"
      }
    ],
    "audio":
    "localhost:42080/content/download/3609?version=1/audio/11.wav?
    inLanguage=eng-GB",
    "swipeFake": "Dom"
  }
]
```

**ДОДАТОК Д. ПОРІВНЯЛЬНА ТАБЛИЦЯ ПОПУЛЯРНИХ  
ФРЕЙМВОРКІВ ДЛЯ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ**

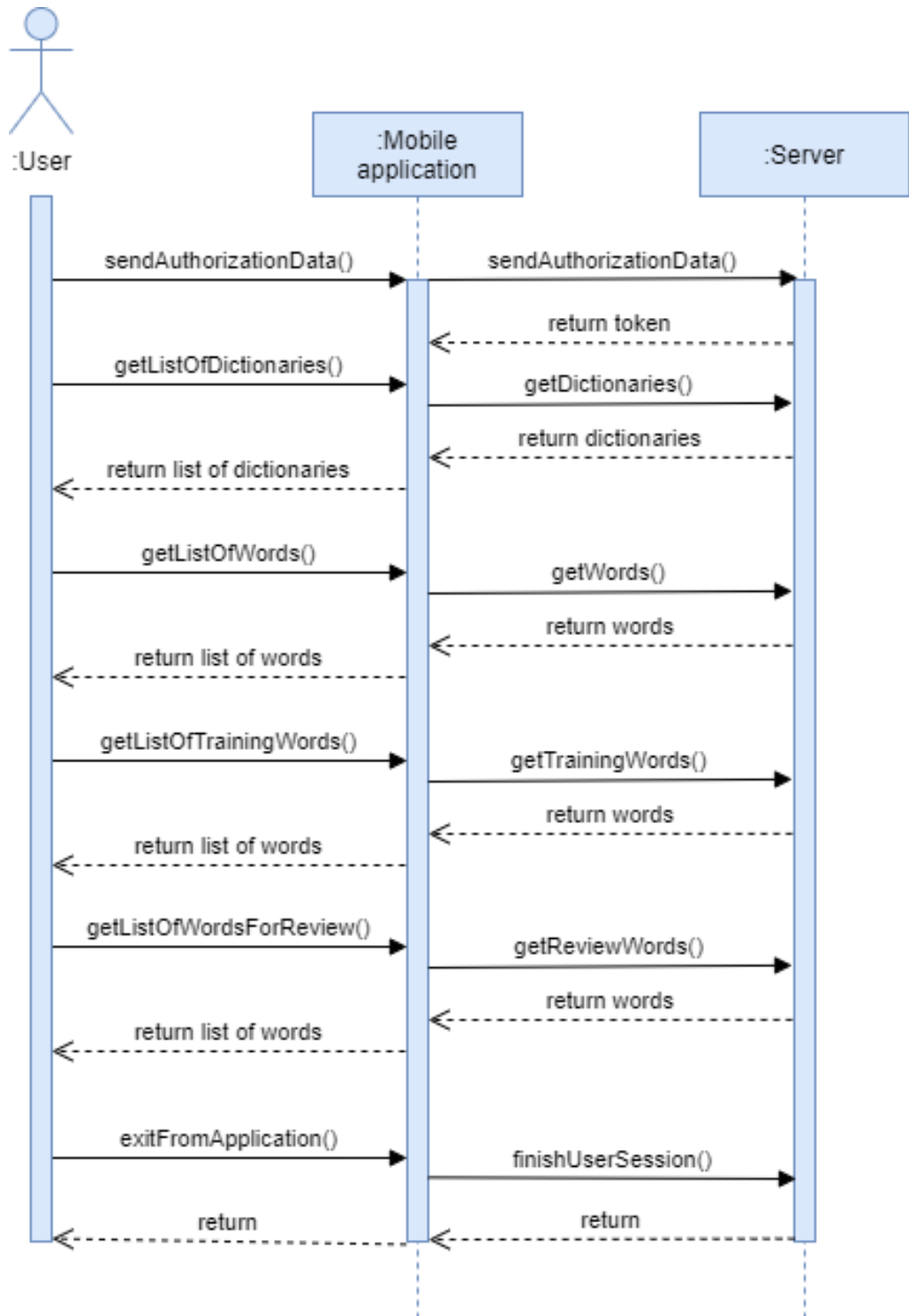
	<b>React Native</b>	<b>Flutter</b>	<b>Ionic</b>
<b>Мова</b>	JavaScript + React	Dart	JavaScript, HTML, CSS + Angular, React, Vue
<b>Тип фреймворку</b>	Кросплатформо вий, гібридо- нативна розробка	Кросплатформо вий, гібридо- нативна розробка	Кросплатформо вий, гібридна веб-розробка
<b>Перший реліз</b>	2015	2017	2013
<b>Розробник</b>	Facebook	Google	Drifty Co.
<b>Платформи</b>	Android, IOS, UWP	Android, IOS, Google, Fuchsia, Web, Desktop	Android, IOS, Web
<b>Відкритий ресурс</b>	так	так	Так + платні пакети
<b>Інструменти фронтенду</b>	Native + Declarative UI components	Built-in widgets	HTML, CSS + widgets
<b>Продуктивніс ть</b>	Висока, близька до нативної	Висока	Середня, через веб-технології

# ДОДАТОК Е. ЧАСТКА РИНКУ СМАРТФОНІВ, НАСТІЛЬНИХ

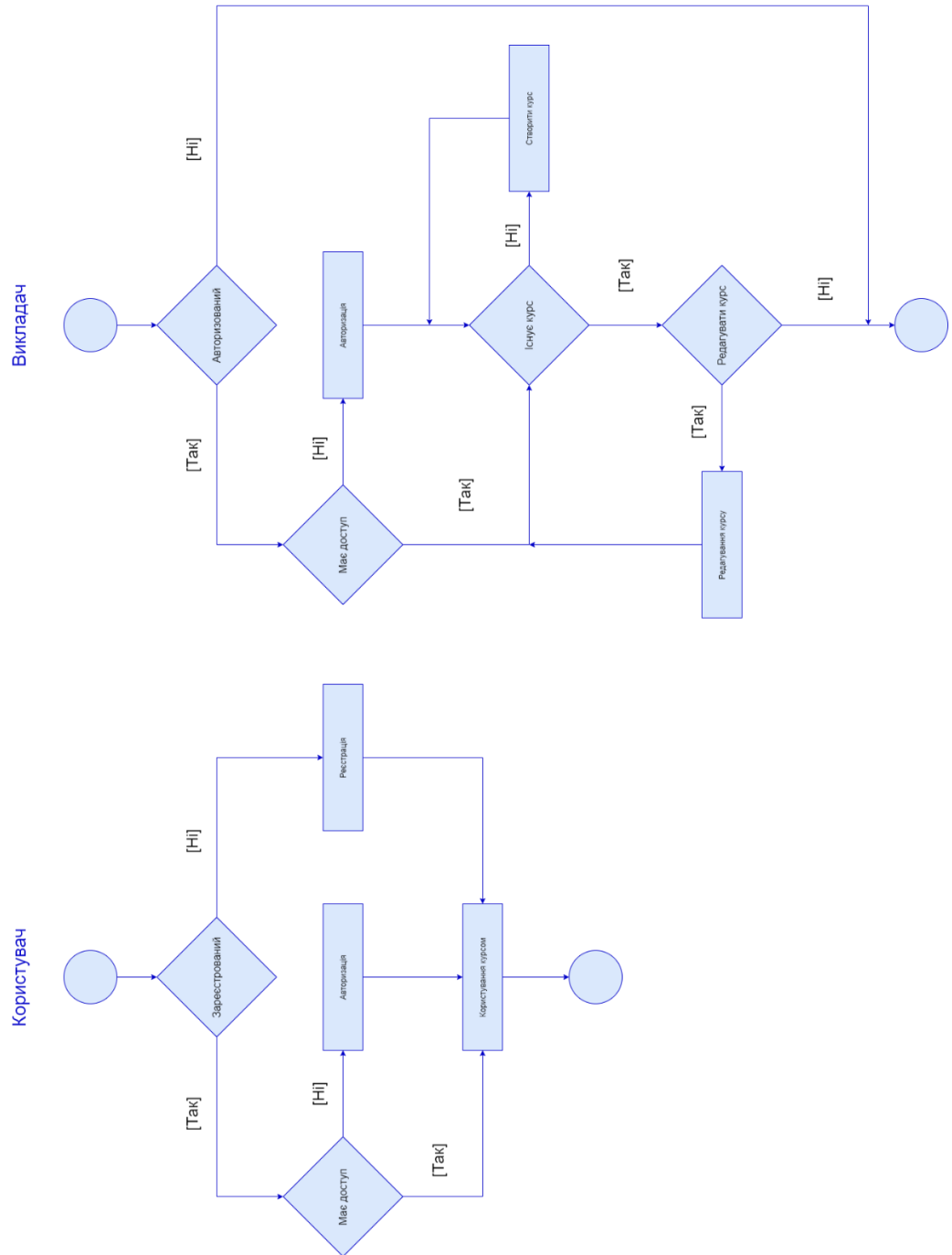


## КОМП'ЮТЕРІВ ТА ПЛАНШЕТІВ У ВСЬОМУ СВІТІ

## ДОДАТОК Ж. ДІАГРАМА ПОСЛІДОВНОСТІ ВЗАЄМОДІЇ ІЗ ДОДАТКОМ



## ДОДАТОК 3. ДІАГРАМА ДІЯЛЬНОСТІ



## ДОДАТОК К. ДІАГРАМА ВИКОРИСТАННЯ

