МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки

Кафедра Інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка мобільного додатку для навчання WEB-дизайну в ігровій формі»

Виконав студент 2 курсу групи МІС- 18
спеціальності 122 Комп'ютерні науки
Рижов Костянтин Сергійович

_____
Керівник к.геог.н., доц.
Кузніченко Світлана Дмитрівна
Рецензент к.т.н., доц.
Гнатовська Ганна Арнольдівна

Одеса 2019

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ

СКБД (Система керування базами даних) – комплекс програмного забезпечення, що надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

ADT (Android Development Tools) – універсальний засіб розробки мобільних додатків для операційної системи Android.

Android TV – версія операційної системи Android розроблена для телевізорів і мультимедійних приставок.

Android Wear – версія операційної системи Google Android, Створена для розумних годинників та других переносних пристроїв.

API (Application Programming Interface) – опис способів (набір класів, процедур, функцій, структур або констант), якими одна комп'ютерна програма може взаємодіяти з іншою програмою.

APK (Android Package Kit) – формат архівних виконуваних файлів-додатків для Android.

ASP.NET (Active Server Pages для .NET) – платформа розробки веб-додатків, до складу якої входить: веб-сервіси, програмна інфраструктура, модель програмування, від компанії Майкрософт.

ETEA (Educational Testing and Evaluation Agency) – агентство з оцінювання та навчального тестування.

GIT – розподілена система керування версіями.

IDE (Integrated Development Environment) – система програмних засобів, яка використовується програмістами для розробки програмного забезпечення.

IntelliJ IDEA – інтегроване середовище розробки програмного забезпечення для багатьох мов програмування, зокрема Java, JavaScript, Python, розроблена компанією JetBrains.

iOS (iPhone OS) – мобільна операційна система для смартфонів, електронних планшетів, програвачів і деяких інших пристроїв, що розробляється і випускається американською компанією Apple.

MongoDB – документоорієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць.

MSSQL – система керування реляційними базами даних, розроблена корпорацією Microsoft.

MySQL – вільна реляційна система управління базами даних.

NoSQL – термін, що позначає ряд підходів, спрямованих на реалізацію сховищ баз даних, що мають суттєві відмінності від моделей, що використовуються в традиційних реляційних СКБД з доступом до даних засобами мови SQL.

NTS (National Testing Service) – національна служба тестування.

OHA (Open Handset Alliance) – бізнес-альянс 84 компаній з розробки відкритих стандартів для мобільних пристроїв, що включає Google Inc., HTC, Intel, Motorola, Asus, Qualcomm, Samsung, LG Electronics, Huawei, T-Mobile, NVIDIA , Wind River Systems та інші компанії.

OS (Operating System) – комплекс взаємопов'язаних програм, призначених для управління ресурсами комп'ютера та організації взаємодії з користувачем.

PHP (PHP: Hypertext Препроцесор) – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків.

REST API (Representational State Transfer – «передача стану уявлення») – архітектурний стиль взаємодії компонентів розподіленого додатка в мережі.

SDK (Software Development Kit) – набір засобів розробки, який дозволяє фахівцям з програмного забезпечення створювати додатки для певного пакету програм, програмного забезпечення базових засобів розробки, апаратної платформи, комп'ютерної системи, ігрових консолей, операційних систем і інших платформ.

ВСТУП

Завдяки засобам нових інформаційних і комунікаційних технологій з'являються нові форми навчання (на додаток до традиційних очного та заочного навчання): дистанційне навчання і мобільне навчання (Mobile learning або M-Learning) з використанням мобільних телефонів, смартфонів, планшетів та веб долатків. У роботі розглядаються сучасні проблеми використання мобільних додатків в загальноосвітньому процесі. Аналізуються сучасні тенденції, що відбуваються в освітньому середовищі з використанням мобільних технологій, їх переваги і недоліки.

Впровадження мобільних технологій для допомоги навчального процесу стане дієвим, в разі якщо:

- Визначена сукупність умов для інтеграції мобільних технологій в систему допомоги навчального процесу;
- важливі і структурні активні властивості мобільної системи допомагають навчальному процесу, стане своєрідною частиною дистанційної освіти і застосовує нове покоління освітніх ресурсів і мобільних приладів і технологій;
- розроблені технологічні основи підтримки навчального процесу, орієнтовані на використання мобільних технологій учнями та викладачами.

У сучасному світі практично у кожного учня загальноосвітньої школи є мобільний пристрій. Причому мобільними пристроями школярі користуються не тільки для розваги або отримання різнопланової інформації, але і для вирішення різних навчальних питань. Поява спеціалізованих додатків для навчання останнім часом стала розглядатися як можливість використання таких мобільних додатків в загальноосвітньому процесі. Аналіз світових тенденцій демонструє життєву гостроту застосування в освітній діяльності мобільних додатків для вирішення різних педагогічних завдань, організації віддаленого доступу до

спеціалізованим ресурсів та сервісів навчальних закладів. Своєчасність застосування мобільних технологій в освітньому середовищі обумовлена наступними передумовами: високий рівень і динаміка поширення мобільних пристроїв (не рідкість, коли один користувач є власником двох і більше пристроїв), стійкий інтерес до їх застосування, можливістю перетворити в медіаконтент і супутнє зміст в інфраструктуру освітнього і науково-дослідного простору.

Метою магістерської роботи є теоретичне обґрунтування та практична інтеграція мобільних технологій в систему підтримки навчального процесу, а саме створення мобільного додатку, який дозволить студентам вивчати нові дисципліни та покращувати свої знання завдяки технологіям мобільних додатків. Також буде мати здатність мотивувати студентів до підготовки за рахунок використання сучасних технологій та мобільних пристроїв.

Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- виконати аналіз предметної області, аналіз подібних та існуючих систем дистанційного навчання;
- обґрунтувати вибір програмних засобів та технологій для створення мобільного застосування;
- розробити мобільний додаток, що міститиме наступні функціональні можливості: внесення і зберігання інформації про користувачів в базі даних; ведення статистики кожного користувача; проходження різноманітних навчальних рівнів; надання дисциплін і їх тем для підготовки до контролю знань; перевірити працездатність системи;
- розробити керівництво користувача.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

На сьогоднішній день існують дві основні концепції застосування мобільних пристроїв в освіті: BYOD (bring your own device) і GYOD (give your own device). BYOD - "принеси свій власний пристрій", концепція, в якій учні приносять свої власні пристрої. GYOD - "дай мені свій пристрій", учням видають мобільні пристрої.

Для BYOD були відзначені такі переваги:

– Зручність використання будь-який мобільний пристрій має крім технічних характеристик додатковими властивостями, які влаштовують користувача, в тому числі і зовнішній вигляд, тому складно, проводячи масову закупівлю, підібрати відповідні пристрої для всіх користувачів;
– старіння техніки ринок мобільні пристроїв розвивається дуже швидко і з'являються нові пристрої, що перевершують попереднє покоління. Загальноосвітні заклади не можуть дозволити собі оновлювати мобільні пристрої так швидко;
– використання пристроїв в особистих цілях - учні використовують мобільні пристрої для особистих цілей.

З недоліків слід зазначити те, що на даний момент не всі учні мають необхідні планшети і смартфони.

Для GYOD вся проблема полягає в бюджеті загальноосвітнього закладу, який доведеться збільшити для видачі мобільних пристроїв всім учням.

1.1 Мобільне навчання як нова форма організації освітнього процесу

Мобільні пристрої, такі як телефони і кишенькові комп'ютери мають набагато більш розумні ціни, ніж настільні комп'ютери, і, отже, є менш дорогий спосіб доступу в Інтернет (хоча вартість підключення може бути вище). Впро-

вадження планшетних ПК тепер дозволяє використовувати мобільний доступ в Інтернет з рівною, якщо не більшою, функціональністю, ніж у настільних комп'ютерів. Більшість мобільних пристроїв є корисними в галузі освіти, управління, організації та викладання для фахівців-практиків, а також технічними засобами підтримки навчання для учнів.

Можливості мобільних додатків для загальноосвітніх установ:

- Функція "мобільний щоденник";
- інтерактивне он-лайн розклад занять і уроків, контрольних робіт;
- сповіщення про терміни здачі та результати контрольних робіт;
- повідомлення про зміни в навчальному процесі;
- швидкий доступ до методичних і навчальних матеріалів;
- інтеграція модулів експрес вивчення іноземних мов;
- зручне використання методики аудіювання;
- зручний контроль знань учнів за допомогою он-лайн тестування;
- використання в процесі навчання додаткових інтерактивних модулів, створюваних для кожної дисципліни.

Навчальні програми з предметів шкільного курсу включають теоретичну інформацію і завдання для перевірки знань. Звичайно, кожне мобільний додаток для освітнього закладу створюється з урахуванням всіх особливостей навчального процесу та специфіки досліджуваних матеріалів в конкретній школі, гімназії або ліцеї.

Використання мобільних додатків в освітньому процесі зачіпає і іншу важливу проблему - діагностики і збереження здоров'я учнів.

Інтенсифікація мобільних технологій призвело до створення ефективних діагностичних програм, які можуть відстежувати фізичний і психологічний стан учнів. Мобільним системам для персоналізованої медицини слід приділяти особливу увагу. Їх потенціал спочатку був орієнтований на їх використання в аерокосмічній медицині. В даний час, персоналізовані медичні технології дос-

тупні для широкого використання, в тому числі і для учнів загальноосвітніх установ, взаємопов'язані з мобільними додатками в освітній сфері.

Переваги і недоліки використання мобільних додатків в загальноосвітньому процесі. Перевагами використання мобільних додатків в освітньому процесі наступні:

– В будь-який момент, в будь-якому місці можна вчитися;
– мобільне навчання часто робляться в ігровій формі;
– здійснюється контроль за відвідуванням;
– спрощення процедури проведення контрольних робіт або індивідуальних завдань;
– проводиться уніфікований контроль над рівнем знань учнів;
– прискорення обміну інформацією між усіма учасниками освітнього процесу.

Недоліками є:

– Можливу шкоду для зору при тривалій роботі з мобільним пристроєм;
– не завжди у дітей є сконфігуровані смартфони та планшети (при концепції BYOD);
– контент-фільтрація (або здійснення функції «батьківський контроль");
– учень може відволікатися на інші додатки, які носять розважальний характер.

Всі недоліки, крім відсутності в учня мобільного пристрою, можна дозволити різними способами, при слабкому зорі можна купувати спеціальні окуляри, фільтрацію контента можна і потрібно робити батькам дітей. Недоліків у використанні мобільних додатків в загальноосвітньому процесі набагато менше, ніж переваг. Існує значна кількість компаній, що займаються розробкою мобільних додатків, в тому числі такі великі фірми, як Whisper Arts, Agilie, Softreactor і інші. Однак, уніфікованих програм, що дозволяють повноцінно інтегруватися в освітній процес, практично відсутні.

В даний час використання мобільних додатків в освітньому процесі все ще перебуває на початковій стадії, але стрімко розвивається. За кордоном за підтримки великих брендів йде активне впровадження мобільних пристроїв в освітній процес. Така ж тенденція спостерігається і в Україні. Процес навчання за допомогою мобільних додатків перспективний, за ним майбутнє.

2 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ

2.1 Додаток Swift Playgrounds для вивчення мов програмування Swift

Компанія Apple представила додаток для iPad - Swift Playgrounds, с яким, зі слів розробників, всі бажаючі зможуть легко навчитися програмувати. Інтерактивний інтерфейс Swift Playgrounds [3][1] допомагає початківцям вивчити Swift - простий в освоєнні мову програмування Apple, на якому професійні розробники створюють додатки світового рівня. Swift Playgrounds включає в себе створені Apple уроки програмування, де учні пишуть код, щоб проводити екранних героїв по захоплюючому графічного світу, вирішуючи загадки, виконуючи завдання і освоюючи найважливіші навички програмування. Вбудовані в додаток шаблони допомагають учням втілювати свої творчі задуми і створювати реальні програми (рис. 2.1).

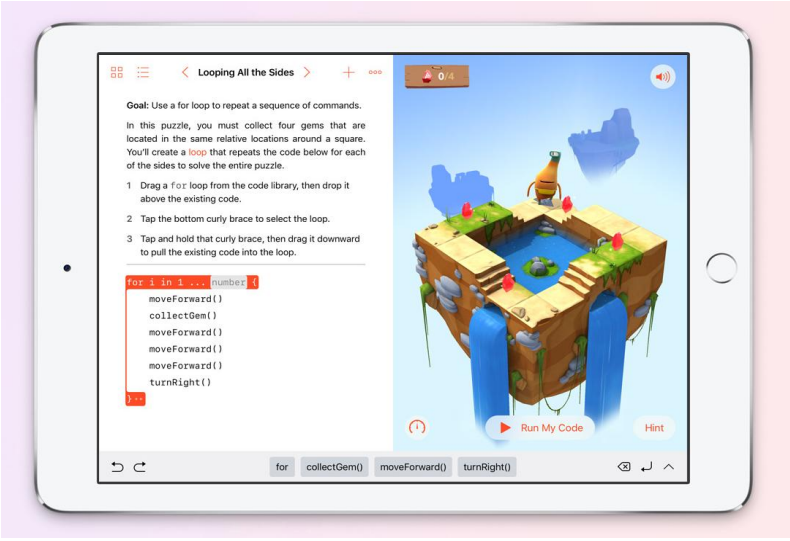

Рисунок 2.1 - Скріншот Swift Playgrounds

[1] [3] Swift Playgrounds. URL: https://www.apple.com/swift/playgrounds/ (Дата звернення 17.09.2019).

Завдяки урокам програмування, створеним Apple, учні зможуть вивчити такі основні поняття, як запуск команд, створення функцій, виконання циклів, застосування умовних виразів і змінних. Це, як вважають в Apple, допоможе їм поступово розвинути свої навички і набути впевненості. Apple буде регулярно випускати нові завдання, щоб учні могли відточувати свої вміння у міру зростання навичок і розширення інтересів. Викладачі та розробники можуть також створювати власні завдання, використовуючи Xcode.

Попередня версія Swift Playgrounds вже сьогодні доступна учасникам програми Apple Developer Program в складі попередньої версії iOS 10 для розробників, а в липні стане доступна в складі загальнодоступною бета-версії iOS 10. Остаточна версія Swift Playgrounds стане безкоштовно доступна в App Store цієї осені. Swift Playgrounds працює на iPad Air і iPad Pro, а також iPad mini 2 і новіше під управлінням iOS 10.

2.2 Додаток Lrn для вивчення мов програмування та гіпер-розмітки

Розробники новинки пропонують вивчати програмування в простій і доступній формі. За своєю концепцією Lrn нагадує Duolingo - сервіс, який перетворює вивчення мов в своєрідна розвага. Якщо ви хочете навчитися програмувати, але не знаєте, з чого почати - це додаток для вас (рис 2.2).

Lrn[4][1] зосереджений на вивченні JavaScript - незамінного в веб-програмуванні мови. З його допомогою створюються сценарії, які додають інтерактивність веб-сторінок. Якщо ви плануєте присвятити себе вивченню цієї гілки програмування, знання JavaScript - необхідний навик. Крім того, ця мова хороший в якості стартового, що тільки збільшує корисність Lrn.

[1)] [4] Lrn education. URL: https://lrn.com/education/ (Дата звернення 17.09.2019).

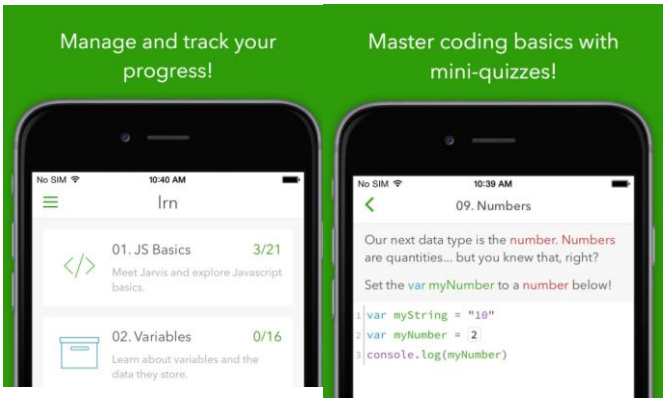

Рисунок 2.2 – Скріншот додатка Lrn

Курс складається з семи тематичних розділів з 15-20 уроками в кожному з них. Lrn орієнтований на новачків без будь-яких базових знань в програмуванні. За час навчання ви познайомитеся зі змінними, функціями, циклами, масивами, об'єктами і операторами вибору. Це кістяк більшості сучасних мов, який допоможе визначитися з планами щодо розвитку в цьому напрямку і спростить подальше навчання по спеціалізованій літературі.

2.3 Додаток Tynker для вивчення алгоритмізації та програмування

Tynker[5][1)] - це візуальний інструмент кодування (веб-сайт на базі HTML5 та мобільний додаток), який навчає дітей програмувати за допомогою блоків коду. Діти можуть розпочати кодування одразу в розділі Play Tynker. Безкоштовні шестигодинні уроки кодування та заняття Hour of Code пропонують досвід початківців за допомогою коротких головоломок кодування. Платні курси пропонують більш глибокий та творчий досвід через JavaScript та Python.

Діти можуть створювати власні проекти, співпрацювати з іншими людьми та ділитися з онлайн-спільнотою Tynker. Інформаційна панель адміністра-

[1)] [5] Tynker – coding to kids. URL: https://www.tynker.com (Дата звернення 17.09.2019).

тора дозволяє вчителям керувати списками та завданнями за допомогою одного входу в систему Google або Tynker. Розумна інтеграція з платформами типу Minecraft, Lego WeDo та Parrot Mambo безпілотників робить Tynker ще більш актуальним для дітей (рис 2.3).

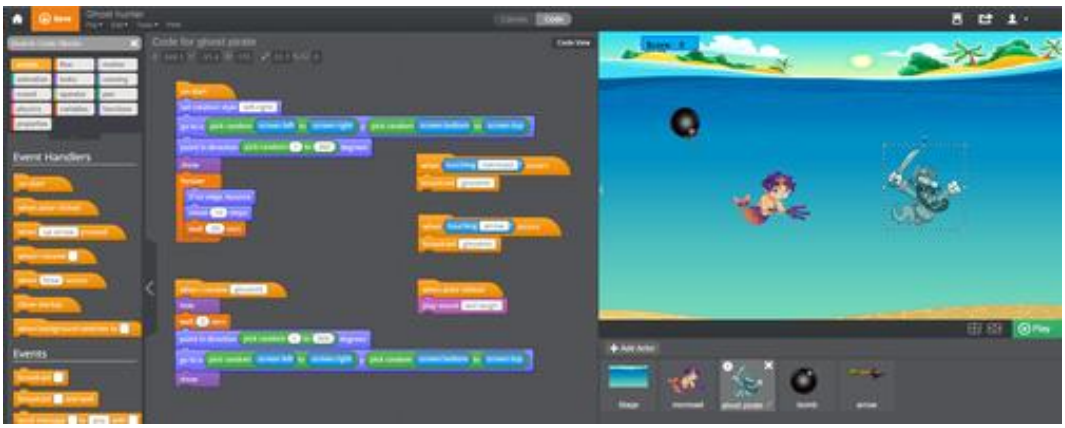

Рисунок 2.3 – Скріншот програми Tynker

Tynker – платформа освітнього програмування, спрямована на навчання дітей, як робити ігри та програми. Замість того, щоб вводити вихідний код, можна візуально перетягувати блоки коду. Візуальний дизайн та принципи засновані на безкоштовній Scratch, як і Hopscotch та Snap. На відміну від Scratch, Tynker не заснований на власних Adobe Flash, але HTML5 і JavaScript, і може використовуватися в браузері без плагінів, а також на планшетах і смартфонах.

2.4 Додаток Coursera для вивчення алгоритмізації та програмування

Coursera [6][1)] – Найбільш популярний портал з перерахованих вище. Нагадує онлайн-університет з безліччю дисциплін найширшої тематики. Зараз портал пропонує понад 600 курсів, які представлені на 13 мовах, включаючи

[1)] [6] Coursera. URL: https://ru.coursera.org (Дата звернення 17.09.2019).

французьку, японську і іврит, але майка лідера належить англійському (засновники ресурсу - професори Стенфорда). Вузи-партнери Coursera - такі гіганти освіти, як Університет Дюка, Стенфордський і Єльський університети (рис 2.4).

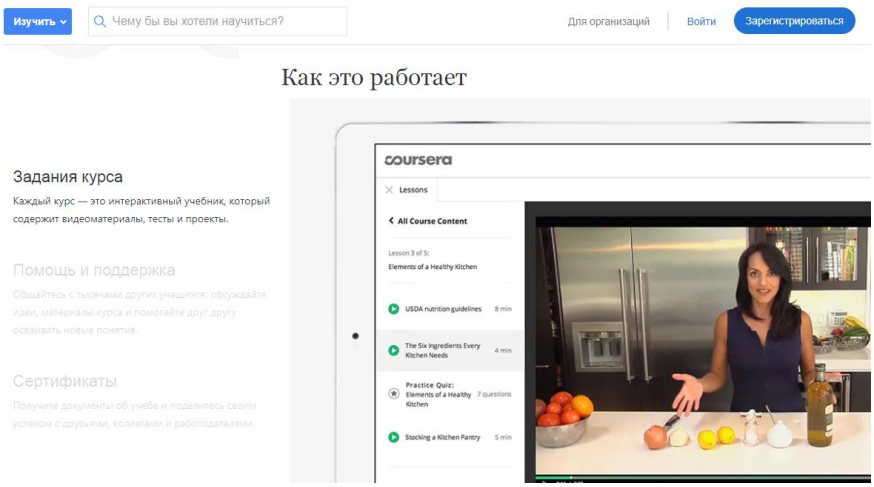

Рисунок 2.4 – Головна сторінка ресурсу Coursera

Багато курсів пропонують цікаву систему оцінювання «peer assessment»: колеги по курсу анонімно оцінюють роботи один одного. Бали виставляються дуже об'єктивно, а ваше завдання розглянуть, протестують і напишуть фідбек. У кожному курсі є форум, де всі студенти можуть спілкуватися, обговорювати завдання і ділитися своїми напрацюваннями. Недолік платформи - сувора прив'язка до часу. Деякі курси рідко повторюються, і деколи доводиться оплакувати втрачений шанс.

2.5 Ресурс Khan Academy для онлайн навчання

Khan Academy [7][1)], вона ж Академія Хана, названа по імені свого засновника - Салмана Хана. У 2006 році він створив цей ресурс як хобі: малював фор-

[1)] [7] Khan Academy. URL: https://ru.khanacademy.org (Дата звернення 17.09.2019).

мули, вирішував приклади і записував процес на відео, супроводжуючи досить неформальними коментарями свої дії. Спочатку сайт був розрахований тільки на учнів з мінімальними знаннями з технічних предметів, але зараз там є і більш складні лекції. Ця платформа ідеально підійде тим, хто звик рухатися в своєму темпі і тільки починає свій шлях в світі онлайн-освіти. Всі уроки на сайті, звичайно, безкоштовні (рис. 2.5).

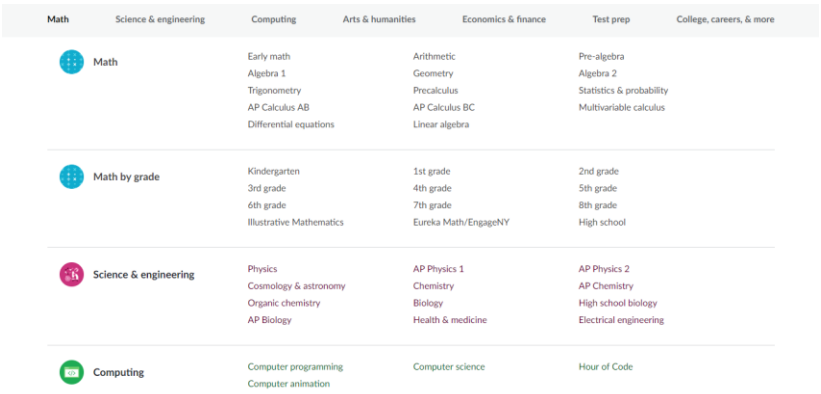

Рисунок 2.5 – Головна сторінка ресурсу Хан Академії

Салман Хан стверджує, що система освіти, представлена цією платформою, переверне світ і змусить стару систему посунутися в бік, звільняючи місце для нового покоління студентів: їх не будуть підганяти під один темп, вони зможуть самі вирішувати, чим хочуть займатися і з яким ступенем глибини вивчати ту чи іншу область. Вони зможуть самі стежити за прогресом, при необхідності консультуючись з друзями, які вже просунулися трохи далі, або звертаючись до професіоналів з найскладнішими питаннями. І вони будуть так само відкриті по відношенню до інших користувачів і будуть щедро ділитися з ними знаннями.

3 ПРОГРАМНІ ЗАСОБИ ДЛЯ РОЗРОБКИ ANDROID-ДОДАТКУ

3.1 Аналіз сучасних засобів розробки мобільних додатків

В даний час все більше користувачів вже перестають користуватися настільними комп'ютерами, віддаючи перевагу персональним мобільним засобам, які дозволяють бути на зв'язку з усім світом в режимі 24/7. На відміну від настільного комп'ютера, мобільний телефон або планшет - це більш особистісне пристрій. У них зберігаються фотографії користувача, номера його кредитних карт, з яких виконуються платежі, акаунти соціальних мереж. Деякі додатки вже здатні допомагати нам у плануванні дій: варто тільки запустити додаток, прийнявши попередньо угоду, і через деякий час воно визначить місце розташування вашого офісу і вдома; після чого буде підказувати скільки часу добиратися до роботи і за скільки необхідно прокинутися, щоб не пропустити важливу нараду або зустріч.

Додатки, доступні для скачування, сьогодні мають найрізноманітніші категорії: соціальні, новинні, освітні та ін. Число додатків з кожним днем стрімко зростає. Однак, щоб створювати додатки, необхідні інструменти для розробки. Сучасні засоби розробки мають наступні параметри:

- Можливість використання об'єктно-орієнтованого стилю програмування;
- можливість використання візуальних компонентів для більш наочного і потужного проектування інтерфейсу;
- можливість використання баз даних;
- можливість використання алгоритмів реляційної алгебри для здійснення керування базами даних;
- можливість синхронізації складових частин проекту (Надається при реалізації великих програмних комплексів).

Нижче переліченими властивостями володіють такі мови програмування: PHP, ASP, Java та інші.

Для вибору засобів розробки використовується метод варіантних обґрунтувань. Цей метод передбачає проставлення значення кожним критерієм із запропонованих, а потім проставлення балів кожному із значень, виходячи з значень порівнюваних об'єктів. Він складається з наступних етапів:

– визначення критеріїв, за якими проводиться порівняння і ступенем їх важливості;
– кожен варіант оцінюється по рівню наявності критерію;
– проставлення балів кожному із критеріїв;
– знаходження суми балів.

Для вирішення поставленого завдання використовувався перелік характеристик, наведений вище (табл. 3.1).

Таблиця 3.1 – Вибір засобу розробки мобільного додатку[7] [1)]

Характеристика	Java	PHP	Perl	ASP.NET
Швидкість роботи додатків	висока	середньо-висока	середньо-висока	висока
Кросплатформеність	Так	Так	Так	Ні
База даних	Oracle	MySQL	PostgreeSQL	MSSQL
Надійність	Висока	Висока	Висока	Висока
Безпека	Висока	Середня	Висока	Середня
Готові бібліотеки	Багато	Багато	Багато	Небагато
Вартість	Безкоштовно	Безкоштовно	Безкоштовно	Середня

[1)] [7] Аналітичний огляд і порівняння можливість операційних систем для мобільних пристроїв URL: https://www.fundamental-research.ru/ru/article/view?id=39079 / (Дата звернення 24.09.2019)

Супровід і розширення функціональності (рівнем нижче ніж у розробників системи)	Низький поріг входження	Низький поріг входження	Високий поріг входження	Середній поріг входження

Продовження табл. 3.1

Характеристика	Java	PHP	Perl	ASP.NET
Додатково	Всю систему краще від самого початку писати з її використанням	Найбільш популярне у використанні середовище розробки	Відмінні можливості по роботі з текстами.	Повільно, але стабільно набирає популярність
Власний досвід	Середній	Великий	Малий	Малий

Необхідно розставити бали по кожному з критеріїв. Далі визначаємо суму критеріїв (табл. 3.2).

У таблиці балів за критеріями «0» бал визначається як найменший показник за обраною характеристикою, тобто це або відсутність критерію, або найменший показник за обраною характеристикою. «5» балів – це найвища оцінка за обраною характеристикою. Якщо характеристика має однакові показники, або не має ступеня порівняння, то їй проставляється бал «1».

Таблиця 3.2 – Таблиця балів за критеріями

Критерії	Java	PHP	Perl	ASP.NET
Швидкість роботи додатків	5	4	4	5
Кросплатформеність	1	1	1	0
База даних	1	1	1	1

Надійність	1	1	1	1
Безпека	4	3	4	3
Готові бібліотеки	1	1	1	0
Вартість	1	1	1	0

Продовження табл. 3.2

Критерії	Java	PHP	Perl	ASP.NET
Супровід і розширення функціональності (рівнем нижче ніж у розробників системи)	3	3	0	2
Додатково	3	4	3	2
Власний досвід	4	5	2	2
Всього	24	24	22	16

В результаті проведеного аналізу інструментальних засобів визначилося, що в якості засобу розробки можна використовувати на вибір мови Java або PHP, як найбільш оптимальні засоби реалізації поставленого завдання з точки зору розробника. Обраною мовою програмування буде Java, так як існує велика кількість сучасних засобів і ця мова більше оптимізована для розробки мобільних додатків.

3.2 Опис операційної системи Android

Існує кілька найпоширеніших операційних систем для смартфонів, такі як, IOS, Android, WindowsPhone, BlackBerry, Simbian.

Для створення мобільного додатку була обрана операційна система Android, тому що:

– Android – операційна система з відкритим вихідним кодом;
– поширеність ОС Android (рис. 3.1);
– доступ до розробки будь-якому користувачеві;
– абсолютно безкоштовна для розробки.

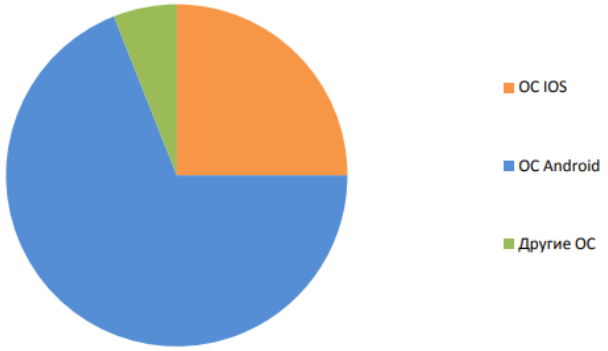

Рисунок 3.1 – Частка пристроїв на різних ОС

Android [9][1)] – це одна з найпопулярніших і найбільш перспективних операційних систем для різних мобільних пристроїв. Система пропонує дуже зручний інструментарій і максимальну гнучкість налаштувань, що дозволяє кожному користувачеві смартфона або планшета на Андроїд налаштувати його повністю під свої потреби. Розробка даної операційної системи стартувала в далекому 2003-му році, але по-справжньому відомою вона стала лише через 2 роки - після придбання компанією Google. Переломний момент в історії Android стався восени 2008-го року. Тоді компанія Google продемонструвала T-Mobile G1, який став першим смартфоном під керуванням Андроїд. Саме в той момент багато світових виробників звернули увагу на перспективну операційну систему. Вже згадана ОС завжди позиціонувалася своїми розробниками як система з відкритим кодом. Це дозволяє будь-якому охочому створювати свої програми, ігри та інші додатки для розширення можливостей Андроїд-гаджетів. Розробни-

[1)] [9] Android. URL: https://ru.wikipedia.org/wiki/Android (Дата звернення 24.09.2019).

ки спочатку продумали все так, щоб операційна система працювала максимально швидко навіть на самому «бюджетному» залозі. Це є безсумнівним плюсом, тому що тепер навіть люди з самими скромними фінансовими можливостями можуть насолоджуватися усіма основними перевагами сучасних смартфонів. Відкритістю системи охоче користуються і виробники мобільної електроніки, випускаючи власні призначені для користувача інтерфейси, наприклад, Sense від компанії HTC. Це робить гаджети на Андроїд від різних виробників несхожими один на одного. Любителям класичного і «чистого» Android слід звернути свою увагу на пристрої Nexus. Саме вони традиційно першими отримують оновлення.

Терміни поновлення інших пристроїв зазвичай затягуються через необхідність доопрацювання фірмових оболонок відповідно до особливостей нових версій ОС.

Головним джерелом ігри та програми є Play Маркет. У каталогах даного магазину доступний величезний вибір безкоштовного і платного контенту, асортимент якого активно розширюється. Якість «андроїдних» ігри та програми постійно зростає. Періодично різні додатки випускає і сам Google. В цілому ж пристрої на Android мають всі функції, якими за негласними вимогам повинні володіти сучасні смартфони і планшетники. Крім цього, функціонал з легкістю розширюється за допомогою додатків, віджетів або сторонніх прошивок.

Бажаючим придбати смартфон на Android необхідно пам'ятати, що багато функцій і програми орієнтовані на роботу з інтернетом. При відсутності доступу до Wi-Fi рекомендується підключити вигідний тариф для інтернет-користувачів або ж відключити деякі функції, що вимагають виходу в мережу.Також не можна не відзначити обмежену автономність пристроїв на Андроїд, особливо представників попередніх поколінь. При активному використанні смартфон або планшет доведеться заряджати щодня, а в деяких випадках і по 2 рази на добу. Виробники активно працюють над виправленням цього недоліку і

останнім часом на ринку стали з'являтися цікаві пристрої з помітно збільшеною автономністю. Не залишаються осторонь і розробники. З кожною новою версією операційна система «їсть» все менше заряду.

3.3 Загальна схема роботи програми Android

Додатки для Android пишуться на мові програмування Java. Інструменти Android SDK (Software Development Kit - комплект розробки програмного забезпечення) компілюють написаний вами код - і всі необхідні файли даних і ресурсів - в файл APK - програмний пакет Android, який представляє собою файл архіву з розширенням .apk. У файлі APK знаходиться все, що потрібно для роботи Android-додатки, і він дозволяє встановити додаток на будь-якому пристрої під управлінням системи Android.

Кожне унікальне вікно має назву Activity, являє собою один екран з призначеним для користувача інтерфейсом. Наприклад, в додатку для роботи з електронною поштою одна операція може служити для відображення списку нових повідомлень, інша - для складання повідомлення і третя операція - для читання повідомлень. Незважаючи на те що операції спільно формують чіткий взаємодія користувача з додатком по роботі з електронною поштою, кожна з них не залежить від інших операцій. Будь-які з цих операцій можуть бути запущені іншим додатком (якщо це дозволяє додаток по роботі з електронною поштою).

3.4 Середовище розробки «Android Studio»

Для створення програми на операційній системі Android, можуть бути обрані різні середовища розробки, такі як, Eclipse, Embarcadero JBuilder, JDeveloper тощо, але для даної магістерської роботи обрана AndroidStudio від

компанії Google. Причиною вибору стали велика кількість доступних технологій та засобів розробки, зручність інтерфейсу і швидкодія програми.

Android Studio[10][1)] – інтегроване середовище розробки виробництва Google, за допомогою якої розробникам стають доступні інструменти для створення додатків на платформі Android OS. Android Studio можна встановити на Windows, Mac і Linux. Android Studio створювалася на базі IntelliJ IDEA.

IDE перебувала у вільному доступі починаючи з версії 0.1, опублікованій в травні 2013, а потім перейшла в стадію бета-тестування, починаючи з версії 0.8, яка була випущена в червні 2014 року. Перша стабільна версія 1.0 була випущена в грудні 2014 року, тоді ж припинилася підтримка плагіна Android Development Tools (ADT) для Eclipse.

IDE можна завантажити і користуватися безкоштовно. У ній присутні макети для створення UI, з чого зазвичай починається робота над додатком. В Studio містяться інструменти для розробки рішень для смартфонів і планшетів, а також нові технологічні рішення для Android TV, Android Wear, Android Auto, Glass і додаткові контекстуальні модулі.

Середа Android Studio призначена як для невеликих команд розробників мобільних додатків (навіть в кількості однієї людини), або ж великих міжнародних організацій з GIT або іншими подібними системами управління версіями. Досвідчені розробники зможуть вибрати інструменти, які більше підходять для масштабних проектів. Рішення для Android розробляються в Android Studio з використанням Java або C++. В основі робочого процесу Android Studio закладений концепт безперервної інтеграції, що дозволяє відразу ж виявляти наявні проблеми. Тривала перевірка коду забезпечує можливість ефективного зворотного зв'язку з розробниками. Така опція дозволяє швидко опублікувати версію

[1)] [10] Android Studio 1.0: первая стабильная IDE от Google. URL: https://habr.com/ru/company/vcstart/blog/364117/ (Дата звернення 26.09.2019).

мобільного застосування в Google Play App Store. Для цього є також підтримка інструментів LINT, Pro-Guard і App Signing.

За допомогою засобів оцінки продуктивності визначається стан файлу з пакетом прикладних програм. Візуалізація графіки дає можливість дізнатися, чи відповідає додаток орієнтиру Google в 16 мілісекунд. За допомогою інструменту для візуалізації пам'яті розробник дізнається, коли його застосування буде використовувати занадто багато оперативної пам'яті і коли відбудеться «прибирання сміття». Інструменти для аналізу батареї показують, яке навантаження припадає на пристрій.

Android Studio сумісна з платформою Google App Engine для швидкої інтеграції в хмарі нових API і функцій. У середовищі розробки ви знайдете різні API, такі як Google Play, Android Pay і Health. Є підтримка всіх платформ Android, починаючи з Android 1.6. Є варіанти Android, які істотно відрізняються від версії Google Android. Найпопулярніша з них – це Amazon Fire OS. В Android Studio можна створювати APK для цієї ОС. Підтримка Android Studio обмежується онлайн-форумами.

Додатки для Android пишуться на мові програмування Java. Інструменти Android SDK (Software Development Kit – комплект розробки програмного забезпечення) компілюють написаний вами код – і всі необхідні файли даних і ресурсів – в файл APK – програмний пакет Android, який представляє собою файл архіву з розширенням .apk. У файлі APK знаходиться все, що потрібно для роботи Android-додатку, і він дозволяє встановити додаток на будь-якому пристрої під управлінням системи Android.

Кожен додаток Android, встановлений на пристрої, працює у власній "пісочниці" (ізольованому програмному середовищі):

– Операційна система Android представляє собою розраховану на багато користувачів систему Linux, в якій кожен додаток є користувачем;

– за замовчуванням система призначає кожному з додатком унікальний ідентифікатор користувача Linux (цей ідентифікатор використовується тільки системою і невідомий з додатком); система встановлює повноваження для всіх файлів в додатку, з тим щоб доступ до них був дозволений тільки користувачеві з ідентифікатором, призначеним цьому додатку;
– у кожного процесу є власна віртуальна машина (ВМ), так що код програми виконується ізольовано від інших додатків;
– за замовчуванням кожен додаток виконується у власному процесі Linux. Android запускає процес, коли потрібно виконати будь-який компонент додатка, а потім завершує процес, коли він більше не потрібен або коли системі потрібно звільнити пам'ять для інших додатків.

Таким чином система Android реалізує принцип мінімальними правами. Тобто кожен додаток за замовчуванням має доступ тільки до тих компонентів, які йому необхідні для роботи, і ні до яких інших. Завдяки цьому формується виключно безпечне середовище, в якій додаток не має доступу до недозволеним областям системи.

Однак у додатків є варіанти надання своїх даних іншим програмам і доступу до системних служб:

– додаток може запросити дозвіл на доступ до даних пристрою, наприклад до контактів користувача, SMS-повідомленнями, яка підключається карті пам'яті (SD-карті), камері, Bluetooth і ін. Всі дозволи повинні надаватися з додатком при його установці;
– двом додаткам можна призначити один ідентифікатор користувача Linux. У цьому випадку кожен з них може звертатися до файлів іншої програми. Для економії ресурсів системи також можна зробити так, щоб програми з однаковим ідентифікатором користувача виконувалися

в одному процесі Linux і використовували одну ВМ (додатки також повинні бути підписані одним сертифікатом).

Це основні відомості про те, яким чином додаток Android існує в системі.

3.5 Емулятор системи Android

Genymotion [12][1)] – один з найкращих емуляторів системи Android, що дозволяє запускати всі програми для цієї мобільної операційної системи на ПК під управлінням Windows. Як кажуть розробники, мета продукту - замінити емулятор Android від Google не тільки розробникам Android, а й людям, які роблять демонстрації додатків Android.

Genymotion доступний для Linux, Windows і Mac OS X і вимагає VirtualBox. Код віртуальний машин відкритий, але софт, який працює на хості, безкоштовний для використання, але його вихідний код закритий. В майбутньому Genymotion матиме безкоштовну версію з безліччю можливостей, але також будуть доступні платні версії, в основному для великих компаній, яким потрібна спільна робота над Genymotion. В каталозі цієї платформи міститься безліч віртуальних пристроїв, які можна використовувати для тестування програм, експериментів з налаштуваннями. На рис. 3.2 приведений зовнішній вигляд емулятору платформи Android «Genymotion». Для роботи цього емулятора необхідна віртуальне середовище VirtualBox.

Віртуальний пристрій емулює всі необхідні вбудовані інструменти Android як в звичайному (реальному) пристрої: функція дзвінків і SMS, робота з мережею і інтернетом, індикація батареї, GPS-позиціювання, управління дозволом екрану, камера і т.д.

[1)] [12] Genymotion. URL: https://android-emulator.net/genymotion (Дата звернення 05.10.2019).

Рисунок 3.2 – Зовнішній вигляд емулятору платформи Android «Genymotion»

Основні можливості Genymotion:

– Величезний список доступних віртуальних пристроїв;
– налаштування координат GPS і рівня заряду батареї;
– режим повноекранного перегляду;
– емуляція передньої і задньої камери;
– загальний буфер обміну віртуального пристрою і комп'ютера;
– можливість редагування IMEI / MEID віртуального пристрою;
– можливість скидання віртуального пристрою до заводських налаштувань;
– редагування кількості процесорів і оперативної пам'яті;
– муляція роботи інтернету через GPRS, Edge, 3G, 4G і багато іншого.

4 ОГЛЯД ОСНОВНИХ МЕТОДОЛОГІЧНИХ ПІДХОДІВ НАВЧАННЯ

4.1 Системний підхід

Системний підхід - це методологічна орієнтація пізнання об'єктивної дійсності і практики управління складними системами. Системний підхід у педагогіці дозволяє відокремити і ретельно вивчити кожен елемент системи окремо, проаналізувати і зіставити їх один з одним, об'єднавши в цілісну структуру. При цьому виявляються всі їх подібності та відмінності, протиріччя і зв'язуючі характеристики, пріоритет одних елементів по відношенню до інших, динаміка розвитку кожного елемента і всієї системи в цілому. Системний підхід у педагогіці не слід плутати з системою педагогічних наук, кожна з яких може бути розглянута з точки зору системного підходу.

Поряд з поняттями «система» і «виховна система» в термінологічну складову системного підходу входять поняття: системність (наявність інтегральних властивостей), компонент (частина системи), елемент (мінімальна одиниця системи), структура (спосіб встановлення зв'язків і відносини), зв'язок (наявність взаємної залежності).

4.2 Особистісний підхід

Сутність:визнає особистість як продукт суспільно-історичного розвитку і носія культури, і не допускає зведення особистості до натури.

Особистість як мета, суб'єкт, результат і головний критерій ефективності педагогічного процесу.

Унікальність особистості – її інтелектуальна моральна свобода, право на повагу. Завдання вихователя: створення умов для саморозвитку задатків і творчого потенціалу особистості.

Особистісний підхід вимагає визнання унікальності особистості, її інтелектуальної та моральної свободи, права на повагу. Він передбачає опору в вихованні на природний процес саморозвитку задатків і творчого потенціалу особистості, створення для цього відповідних умов.

4.3 Діяльнісний підхід

Сутність: діяльність – основа, засіб і умова розвитку особистості, це доцільне перетворення моделі навколишньої дійсності.

Завдання вихователя – вибір і організація діяльності дитини з позиції суб'єкта пізнання праці і спілкування (активність самого). Це передбачає: усвідомлення, цілепокладання, планування діяльності, її організація, оцінка результатів і самоаналіз (рефлексія).

А.Н. Леонтьєв. «Для оволодіння досягненнями людської культури, – писав він, – кожне нове покоління повинно здійснити діяльність, аналогічну (хоча і не тотожну) тієї, яка стоїть за цими досягненнями». Ось чому, щоб підготувати вихованців до самостійного життя і різнобічної діяльності, необхідно в міру можливостей залучити їх в ці види діяльності, тобто організувати повноцінну в соціальному і моральному відношенні життєдіяльність.

Діяльнісний підхід вимагає перекладу дитини в позицію суб'єкта пізнання, праці і спілкування. Важливим при цьому є те, що активність особистості, її потреби в самовдосконаленні розглядаються неізольовані. Вони розвиваються лише в умовах взаємин з іншими людьми, побудованих за принципом діалогу. Діалогічний підхід в єдності з особистісним і діяльнісних становлять сутність методології гуманістичної педагогіки.

4.4 Індивідуальний підхід

Це психолого-педагогічний принцип, в якому постулюється важливість для навчання і виховання обліку індивідуальних особливостей кожної дитини. У педагогічній роботі вихователь (учитель) організовує свою діяльність з урахуванням особливостей кожного вихованця т. К. Все діти різні, а значить і здатності у них теж різні.

4.2 Види сприйняття інформації

Сприйняття як безпосереднє відображення світу класифікується за різними підставами.

Традиційно виділяють п'ять видів сприйняття відповідно до провідним аналізатором, які беруть участь в побудові перцептивного образу (по модальності сприйняття):

- Візуальне;
- аудиальное;
- дотикове (тактильну);
- смакове;
- нюхові.

Розрізняють також види сприйняття в залежності від об'єкта сприйняття, наприклад, сприйняття простору, часу, руху, швидкості; творів живопису, музики; основних явищ соціального життя людини (іншої людини, подій суспільного життя) і т.п.

Візуальне сприйняття грає велику роль в нашому житті. Воно допомагає пізнавати навколишню дійсність і взаємодіяти з іншими людьми. Для людини бачити зовнішній світ настільки природно, що багато людей навіть не замислюються про величезне значення цього каналу сприйняття. Розуміння того, як

візуал сприймає інформацію, відкриває великі можливості для більш тісної взаємодії з покупцями.

Психолог Річард Грегорі стверджує, що візуальне сприйняття людини йде «зверху вниз», тобто від загального до конкретного. Візуали формують свою картину сприйняття, рухаючись від великих деталей до більш дрібних елементів. Це дозволяє їм робити вірні припущення про те, що вони бачать. Іншими словами, будь-який візуал в магазині робить розрахункові припущення. Якщо товар сподобався йому з першого погляду, він постарається розглянути його у всіх подробицях, щоб переконатися, що очі його не обманули і продукт відповідає його очікуванням і рівню минулого досвіду.

Працювати з візуалами дуже легко. Неважливо, яким буде сегмент бізнесу знання того, як візуал сприймає дійсність, допоможе уявити йому товар або послугу в найбільш вигідному світлі. Красиво оформлена вітрина, правильно розставлені товар на полицях, елегантний інтер'єр офісу - і візуал покірний. Красива картинка виходить для нього дуже багато.

Але не варто обманювати визуалов. Навіть невеликий недолік не залишиться без уваги. Будь-який товар для людей із зоровим типом сприйняття повинен відрізнятися першокласним дизайном і бездоганним зовнішнім виглядом.

Домінуюче почуття типового аудиала - слух. Така людина сприймає навколишній світ через звуки. Більшість аудиалов люблять музику, їм подобається слухати радіо, і навіть дивлячись фільм по телевізору, вони набагато більше уваги звертають на те, що говорять персонажі, ніж на те, як вони виглядають.

Розпізнати аудиала дуже легко. У нього не загоряються очі при вигляді чогось яскравого і незвичайного, вітрини і полиці з товарами його якщо й цікавлять, то не дуже сильно. Аудіали - справжня знахідка для амбітних продавців і менеджерів з продажу. Людина зі слуховим типом сприйняття завжди уважно слухає, що йому говорять. При цьому він часто стоїть до співрозмовника боком так йому легше засвоїти отриману інформацію.

Потрібно пам'ятати, що у аудиала вибіркове сприйняття. Такі люди схильні аналізувати отриману інформацію, особливо, коли їм здається, що співрозмовник схильний до маніпуляцій. У спілкуванні з аудіалом потрібно говорити спокійним впевненим тоном і бути максимально переконливим.

Не завадить і трохи гумору: жарт чи кумедний афоризм - це невелика зачіпка в пам'яті людини зі слуховим сприйняттям. І чим більше таких зачіпок, тим більша ймовірність, що потенційний клієнт перетвориться в постійного покупця.

До кинестетикам відносяться люди, які більшою мірою покладаються на тактильні або смакові відчуття. Нюх теж грає роль, але не занадто велику, якщо мова йде про ритейлі або ділових відносинах. Кінестетик отримує максимум інформації при першому тактильному контакті з продуктом, про що ніколи не можна забувати. Наприклад, відвідавши незнайому йому кафе, кінестетик навряд чи в найдрібніших подробицях опише інтер'єр закладу або зможе сказати, яка музика звучала в залі. Але те, що їжа була несмачною, а кава - трохи теплим, він запам'ятає назавжди. Як і аудиалов, у кинестетиков дуже розвинене вибіркове сприйняття і цю особливість можна використовувати для створення ефективного комунікаційного каналу.

Підприємства громадського харчування, парфумерні магазини, косметичні салони і бутіки - це місця, де основний акцент потрібно робити на кинестетиков. Саме ця категорія людей формує основну громадську думку про бренд або роздрібної точці. Саме кинестетики скоріше за все розкажуть, чи смачно годують у ресторані, наскільки цікавий вибір ароматів у відділі парфумерії та якої якості матеріал, з якого пошито одяг, що претендує на звання дизайнерської. Щоб збільшити продажі за рахунок кинестетиков, буде потрібно не тільки показати товар лицем - потрібно буде дати його помацати. Приміряти, пом'яти в руках, спробувати або навіть понюхати - все залежить від виду товару.

В моїй роботі я задію два види спийняття візуальне та тектильне. Вони допожуть великой кількості людей сприймати та заучувати інформацію, а візуальний простий інтерфейс полігшує цей процес.

4.5 Візуальне сприйняття

При розробці якісного User Experience дуже важливо розуміти, як ми, homo sapiens, сприймаємо інформацію, яка надходить із зовнішнього світу. І, звичайно ж, робимо ми це завдяки почуттям

Візуально людина може захопити не так багато пространства.От близькою до далекої периферії ми можемо розрізняти лише контраст, світлі і темні ділянки, чорне і біле.Окреслені ділянки в центрі можуть сприймати детальну інформацію, таку як розпізнавання осіб, читання тексту і подібні речі.

Процес послідовної фіксації очей походить від того як ми читаємо текст. Така фіксація відбувається по тому, що ми вивчили як повинен йти текст, в нашому випадку зліва направо.Ряд досліджень показав, що за тим же шаблоном обробляється інформація на веб-сторінках. Люди не сприймають все частині сторінки однаково.

Завдяки дослідженням Якоба Нільсона (фахівець з юзабіліті) стало відомо, що користувачі схильні слідувати інформації на веб-сторінці за F-образному шаблоном. Нижче представлена теплова карта (рис. 4.1), яка складена за технологією відстеження очей. Червоним кольором позначені місця - точки уваги, на які дивилося найбільша кількість користувачів, жовті - менше, синій колір - невелика кількість і сірий - ніхто не звертав увагу. Справа на картинці зображено сторінка пошуку Google і то, як люди сприймають інформацію на ній.

Користувачі дивляться зліва направо і побіжно переглядають зверху вниз в пошуках інформації, що цікавить. Таким чином вийдуть наш шаблон F.

Рисунок 4.1 – Теплова карта зорового сприйняття

З огляду на те, про що ми тільки що говорили: фіксації очей, про те, що люди не завжди бачать все на сторінці зробимо висновок для правильного проектування інформації на сторінці.

4.6 Дотикове (тактильне) сприйняття

Тактильне сприйняття - це пізнання (обстеження, розпізнавання) об'єктів при безпосередньому торканні. Цей вид сприйняття ще називають дотикальним. Його суть полягає в відображенні характеристик, особливостей поверхні і структури того, до чого торкаються.

Механізм тактильного сприйняття заснований на дії шкірних рецепторів. Рецептори є складним пристроєм, що сприймає сенсорні сигнали і зчитує інформацію. Ці пристрої розташовані по поверхні всього тіла, забезпечуючи шкірну чутливість.

Самі сприйнятливі до різноманітної інформації рецептори сконцентровані на пальцях рук. Обмацуючи предмети, людина може отримати так багато інформації, як ніби він дивиться на них. Тому дотик вважають, в першу чергу, пізнавальною функцією руки людини.

Процес сприйняття надає дитині можливість формувати образи, робити пов'язані з ними висновки. Чуттєве відображення і тактильні відчуття дають такий же багатий матеріал для розумової переробки, як і зорові.

Значна кількість інформації від оператора через візуальні та слухові канали викликає серйозні перевантаження. Крім того, через наслідки певних перешкод, сигнал цієї форми може сильно спотворюватися та викликати помилки при розпізнаванні інформації. У зв'язку з цим останнім часом відбувається пошук можливості передачі інформації про інші канали розпізнавання інформації оператором. Найбільш перспективним є використання тактильного аналізатора.

Експериментальні дослідження показали, що дотичні зображення формуються на основі синтезу значної кількості тактильних та м'язових сигналів. Відомо, що шкіра людини визнає термічні, хімічні, механічні та електричні подразники. Якщо перші два способи передачі інформації ще не доступні, є певні досягнення щодо останніх двох (47 100).

Механічні подразники передаються з допомогою вібратора і виявляються різними способами різними частинами шкіри тіла. Абсолютна чутливість вимірюється як мінімальний тиск, необхідний для відчуття. Найчутливіша область (губи, язик) – 1 ... 50 мг / мм2, найбільш чутлива область (спина, верхня) – 10 г / мм2.

Диференціальний диференціальний поріг становить близько 7% від початкового тиску.

Космічна чутливість також залежить не тільки від характеристик подразника, але і від характеристик конкретної області людського тіла. Диференціальний пороговий простір становить принаймні від 1 до 2,5 мм від губ і пальців, а максимум – 60 мм від спини та плечей. Найвища чутливість спостерігається при частоті вібрації 100 ... 300 Гц.

Відомий цікавий спосіб передачі інформації через вібратор. Людська мова записується у фільмі і відтворюється повільніше, ніж звичайний темп. Отрима-

ний низькочастотний електричний сигнал змінюється у механічну вібрацію пластини, торкаючись поверхні шкіри людини. Після деякого тренування оператор може визначити стандартний звук мови. Цей метод може використовуватися для передачі сигналів у середовищах із значним шумом, коли аналізатор слуху є неефективним.

У випадку людей із вадами зору (слабозори та зору) роль тактильного аналізатора широко використовується, оскільки інформація про навколишнє середовище є єдиним каналом, який передається в мозок людини.

Характеристики тактильно-вібраційної чутливості людини розглядаються при побудові оригінального пристрою для людей з вадами зору "Optocon", в яких оптичний сигнал перетворюється в тактильну вібрацію, надаючи сліпим можливість прочитати існуючу книгу. Швидкість – 40-50 слів за хвилину.

Кожне досягнення в цьому напрямку має деякі недоліки, які обмежують використання механічних методів передачі інформації. По-перше, це неповнота самого вібратора, тобто об'єм та інерція. У зв'язку з цим розвивається використання стимулів електричного стимулювання для передачі інформації з імпульсами прямокутного струму.

Використання електричного стимулятора вимагає попередньої підготовки, після чого абсолютний поріг зменшується, а інший поріг підвищується, але тоді час майже не змінюється і не залежить від області стимулу. Старі люди вищі, ніж молодші.

Важливою перевагою електричного подразнення шкіри в порівнянні з механічною шкірою є потреба в силі сигналу (100 разів) та можливість використання дрібних електродів. Це дуже важливо при розробці пристроїв для передачі інформації через тактильний аналізатор. У Кореї та за кордоном розроблені тактильні стимулятори та тактильні кодекси, які покращують доставку інформації водіям.

Різні канали передачі інформації використовують різні аналізатори, але вони не функціонують окремо, але вони працюють в одній дуже складній системі. У цьому випадку вплив подразника на конкретний аналізатор впливає не тільки на пряму відповідь, але і на функцію всіх інших аналізаторів. Наприклад, чутливість центрального поля залежить від впливу гучного звуку, а збільшення запаху, солодкого смаку, комфортного положення тіла, атмосферного тиску або подразнення шкіри зменшує чутливість навколишнього поля.

4.7 Тестування

Тест – це теоретично і практично обґрунтована система висловлювань, завдань, що дозволяє отримати вимірювання відповідних психологічних властивостей.

Відомо, що термін «тест» походить від англійського слова «test» - випробування, проба. Історія зберегла ім'я людини, який першим використовував слово «тест» в якості наукового терміна, що означає психолого-діагностичну методику. Це заслуга американського психолога, першого в світі професора психології Дж.М.Кеттелла (1860-1944). У 1890 році в одному з наукових журналів їм була надрукована стаття «Mental Tests and Measurements». Вчений розробив серію тестів (50) і використовував їх у створеній ним лабораторії (1891) при Колумбійському університеті для оцінки рівня інтелектуального розвитку.

У шкільній практиці тести використовувалися набагато раніше. У ряді джерел можна знайти згадку про те, що тести були вперше застосовані в 1862 році у Великобританії. З метою перевірки знань учнів учитель Грінвічській школи Дж. Фішер створив градуированную книгу, яка містить питання-завдання і відповіді на них (учні повинні були вибрати правильну відповідь). Книга Фішера охоплювала такі предмети як граматика, математика, французьку мову, святе письмо, загальна історія, малювання та ін.

Теорія тестування розвивалася в роботах зарубіжних психологів XX століття А. Біне, В.Штерна, Ч.Спирмен і ін. Основоположником тестології вважається англійський психолог Ф. Гальтон (1822-1911), який вніс істотний внесок в розробку теорії тестування, тестових завдань і їх застосування для педагогічних цілей.

В даний час під тестами розуміються стандартизовані завдання, призначені для перевірки знань, умінь, професійної підготовки в усіх областях.

Існує кілька підходів до класифікації тестів. По предмету дослідження виділяють три групи тестів:

1) Загальні тести. З їх допомогою вивчають психічні властивості особистості як цілісне утворення.

2) Особистісні тести. Призначені для діагностики тієї чи іншої риси, характеристики, властивості (інтелектуальних і спеціальних здібностей, рівня загальної відповідальності, самоконтролю, властивостей особистості). Особливо широке застосування ця група тестів знайшла при вимірюванні психічних процесів (пам'ять, увага, мислення та ін.)

3) Групові тести. Застосовуються при діагностиці групових психічних процесів - рівня згуртованості груп і колективів, особливостей групового психологічного клімату, міжособистісних взаємин, сили «тиску» групи на неї членів та ін.

4.8 Методи генерації тестових завдань

Метою завдання є аналіз існуючих методів тестування та розробка моделі, яка буде являти собою автоматизований тест з наступними вимогами:

– Мобільні пристрої або планшетні ПК і електронні книги легше і займають менше місця, ніж файли, паперу та підручники, і навіть ноутбуки;

Розпізнавання за допомогою стилуса або сенсорного екрану стає більш наочним, ніж при використанні клавіатури і миші;

– легкість створення бази знань, що буде основою створення тестових робіт;
– можливість налаштування різних типів завдань тестування;
– ясність словника;
– гнучкі можливості тестування на основі важливих сфер навчального вмісту, цілей та стратегій тестування;
– можливість використання результатів тестування для адаптації індивідуальних навчальних процесів;
– модель може бути інтегрована з мережевою ієрархічною мережевою моделлю навчальних матеріалі.

Щоб удосконалити та спростити процес підготовки дистанційних курсів за допомогою тест-блоків, вчені запропонували різні підходи до автоматизації створення тестових завдань. Існують методи перевірки параметрів, які відносно легко реалізувати з вашою аудиторією. Ключ до цього методу – забезпечити шаблон для декількох студентів, які можуть бути різними для певних параметрів, які автоматично генеруються. Відповідь вводиться на клавіатурі. Тому, згідно з певною формулою або алгоритмом, система замінює параметри, так що кожен учень отримує мету особи та студент може отримати правильну відповідь, щоб підтвердити відповідь. Недоліком цього підходу є вузьке націлювання. Тому параметри моніторингу та тестування підходять для програми точних наукових і практичних методів для тестування знань з теоретичних знань та функцій управління.

Підхід до використання семантичних мереж для тестування автоматизації був отриманий з вивчення автоматизації управління знаннями. Основна семантична мережа акордів. Є кілька ключових моментів так званого 1 природного небінарного "процесу" і "програми". У цьому випадку ви можете встановити

зв'язок між типами "часткового типу". Тоді ви розумієте, що суть "процедури" є "частиною" сутності "програми". Тестова робота полягає в тому, щоб пропустити одну з тріадних посилань і запитати, чи немає посилань. Перевага такого підходу полягає у спроможності системи виходити з знань області. Недоліком є висока вартість розробки повної семантичної мережі, яка точно відображає предметну область дослідження. Іншим недоліком такого підходу є неоднозначність мови, а іноді і неможливість формованої роботи. Таким чином, на основі семантичної мережі часто виникає питання про об'єкт функціональної області, який не має вивчення індивіда в контексті освітнього значення. Цей недолік виникає з проблем, які характеризують класичну модель знання штучного інтелекту, яку можна назвати проблемою всіх знань. Оскільки дані опитування відбуваються у певних ситуаціях, поведінковий тест для поширених запитань, таких як це сталося, через вимоги до мережі, успішність студентів та недоліки навчання, значення повного відображення формату домену не відповідає. Поінформованість чи комплексне формулювання вимог потребують значних зусиль для створення правильної моделі без належної підготовленості у навчальних цілях та тестуванні.

Ці та інші питання притаманні здатності застосовувати класичну модель ІІ до навчальних завдань. Незважаючи на те, що дві моделі можуть забезпечити дводоменну систему вирішення, перша мета другого випадку, симулювання "знаю", а другий – мета моделювання, але знання штучного інтелекту та освіти. Система моделювання дає можливість "вчити" людські знання про певну область, яка може мати суттєві відмінності тут. Суттєвою відмінністю у створенні цього завдання є необхідність розробки спеціалізованих моделей формалізації знань для навчання та організації на основі автоматизованого тестування. Основним засобом доставки наукової інформації студентам в дистанційну освіту є форма тексту, яку можна зрозуміти у світлі цього навчання. З огляду на це, до-

цільно зосередитись на форматі моделі, яка автоматизує конфігурацію тестових операцій на основі тексту.

Формалізація уроку тексту. Тезисова концептуальна модель (ПТМ) розроблена на перетині багатьох наукових дисциплін, включаючи наступні тексти: знання напрямку та інженерії штучного інтелекту; Освіта, тобто уроки, що розкривають правила навчання - тлумачення (тлумачення), вивчення правил інтерпретації текстів; Лінгвістика, її семантика, регулярність природної мови, зміст лексичної єдності, сенс і тлумачення. Інструменти впровадження - це технології, що розробляють веб-системи.

У програмі можна виділити чотири компоненти, такі як зв'язок між учасниками програми, передачу навчальної інформації, такої як управління класом, знання та соціальні компоненти. Перша місія дистанційної освіти полягає у спрощенні та оптимізації поширення навчальної інформації, знань студенту. Історично склалося так, що найбільш рання форма передачі інформації в традиційних формах навчання є усною: наприклад, читачі читають лекції. Натомість основним форматом передачі дистанційної навчальної інформації від віддаленого вчителя є текстовий або, швидше, гіпертекстовий та мультимедійний вміст. Іншими словами, основним засобом передачі знань, або "протоколом", є текст, що містить навчальну інформацію. Це є основою концептуальної моделі знань тезауруса.

Основою структури ПТМ є така ж сама, як і в деяких областях знання концепцій, дискусійних тем та навчальних матеріалів. Ця концепція обговорюється і вказує на певний предмет і надається студентам. Наприклад, в курсі «Алгоритмизація та програмування» можна виділити такі поняття, як «процедура», «цикл», «програма», «змінна», «життєвий цикл програми». У процесі "Програмування в середовищі Java" можна виділити такі поняття як "об'єкт", "подія", "клас", "форма", "компонент JButton". Набір системних понять відображається наступним чином:

$$C = \left(C_{1,} \dots C_{n1}\right) \tag{4.1}$$

Включає резюме, спеціальні елементи структури для подання знань концепцій ПТМ. Цей документ є концептуальним твердженням. Її можна порівняти з символами, атрибутами чи твердженнями про цю концепцію. Це поняття відноситься лише до тем, що розглядаються в тексті, але цей документ є знаннями або технічним. Повнота цих наборів пропозицій залежить від повноти бази знань та здатності системи освіти генерувати ефективні тестові роботи. Фактично цей білий документ містить одне або декілька речень, які безпосередньо описують конкретну концепцію, але саме поняття не відображається у словнику. З точки зору лінгвістики ця стаття в принципі виключена. Ось кілька прикладів: Документ про поняття "процедури" – "Програми можна розділити на підпрограми"; "Клас" – "Тези про поняття методів, тобто функції та процедури", а також поля-атрибути можуть бути включені в структуру. Підсумок резюме системи виглядає наступним чином.

$$\mathrm{T} = \left(\mathrm{T}_{1,} \dots \mathrm{T}_{n2}\right) \tag{4.2}$$

Поняття та резюме називаються РТ елементами. Кожна стаття пов'язана з концепцією. Це з'єднання забезпечується наступним співвідношенням:

$$\mathrm{CT}: \mathrm{T} \rightarrow \mathrm{C} \tag{4.3}$$

У свою чергу, кожне поняття може мати будь-яку кількість абстракцій, описаних відносинами:

$$\mathrm{TC}: \mathrm{C} \rightarrow 2^{\mathrm{T}} \tag{4.4}$$

Як зазначалося вище, основним вектором знань є навчальний матеріал. Таким чином, вибір конкретної семантичної одиниці тісно пов'язаний з процесом маніпулювання підручниками. Підготувавши систематичні інгредієнти вуличного курсу, всі інгредієнти в підсумку поділяються на дрібні шматки, також називають пострілами. Тому на кожному етапі навчальної інформації студенти отримують невелику частину, яка дозволяє їм краще зосередити увагу та визнати поданий матеріал. Концепція рамки дуже важлива в ПТМ. Це забезпечує зв'язок між значущими елементами та масою. У цій статті ми не обмежуємось конкретною реалізацією загальної структури збереження та подання тексту. ПТМ також дозволяє іншим реалізаціям цієї структури інтегрувати ПТМ з існуючими системами дистанційного навчання. ПТМ є одним з компонентів системи освіти з іншими моделями. Включає конструювання навчальних матеріалів з використанням спеціальних моделей вмісту. Позначимо частину чи сторінку навчального вмісту.

$$V = \{V_{1,} \dots V_{n3}\} \tag{4.5}$$

Семантичні елементи ПТМ призначаються безпосередньо з тексту фрагмента навчання. Сам процес формування концептуального тезауруса насправді читає текст значущим чином і вимагає простих маніпуляцій. Тому при вивченні підручника вчитель призначає атрибути безпосередньо до тексту та додає важливу концепцію навчання та статтю до БЗ. В результаті кожний фрагмент Vi може бути джерелом певної кількості зелених tj's за відбиттям.

$$\mathrm{T}V: V \rightarrow 2^{\mathrm{T}} \tag{4.6}$$

Кожен tj складається з короткої шматки у свою чергу. Vi:

$$VT: T \rightarrow V \tag{4.7}$$

Оскільки дисертація належить до єдиного навчального поля, поняття та поняття, що застосовуються для багатьох освітніх полів, і зв'язки між підручниками, надаються опосередковано через підручники з концептуальних дисциплін. Поняття, пов'язані з цією сферою освіти, визначаються оператором.

Як результат, навчальний матеріал, що випливає з цієї концепції, визначається оператором.

Схематична схема площі ПТМ для тексту показана на рис. 4.2.

Поняття і тези можуть бути віднесені до певних класів. Ця класифікація служить для збереження в БЗ інформації про смисловий або лексичний характер того чи іншого поняття чи тези. Класи понять і тез використовується в алгоритмах побудови тестових завдань.

Рисунок 4.2. Ділянка ПТМ у співвідношенні з навчальним матеріалом

ПТМ забезпечує дистанційне тестування генератора системи навчання, завдяки унікальності кожного тесту на конфігурацію, що дозволяє звести до мінімуму проблему великих банківських проблем, неможливо несправедливого тестування, статичного тестування або інспекції. PST – це достатньо, щоб розвинути мережевий домен, оскільки він коштує менше, ніж достатньо, щоб створити великі банківські та невеликі витрати на оплату праці одночасно з витратами на оплату праці. На відміну від тестів, сгенерованих в семантичній мережі, лексичний запас на основі ММТ активно впливає на контроль якості. Точна передача семантичних даних у сферу навчальних матеріалів дозволяє проводити різноманітні стратегії тестування форматної оцінки. Таким чином, ПТМ дозволяє створювати тести в певних областях навчального вмісту, а також у результатах тестування, ви можете точно сприймати освітні концепції, які потребують оновлення студентів та тренування фрагментації.

Фактична реалізація підходу ПТМ та автоматизоване тестування виконуються в рамках Порталу дистанційного навчання. Інструменти впровадження програмного забезпечення Включає асинхронний веб-інтерфейс та модуль створення і аналіз тестів на базі даних РТ-додатків для ММТ.

Більше досліджень має бути спрямовано на підвищення якості тестової роботи та вдосконалення методів та алгоритмів для цього покоління. Що є перспективним – це розширити базову класифікацію тез і концепцій та класифікувати нові шаблони для тестування. Іншим перспективним напрямком є створення моделі шаблонів для відкритого тестування.

4.9 Drag-and-drop

Інтерфейси стають все більш об'єктно-орієнтованими, більш графічними і візуальними, тому перевага все частіше віддається прямому маніпулюванню. Однак проблема методу Drag and Drop («перетягнув і кинув») полягає в тому,

що відсутня візуальне вказівку на те, що об'єкти можуть або повинні бути переміщені і скинуті на інші об'єкти. Якщо користувачі не вивчать, як і коли використовувати даний метод, то потенційні переваги не будуть реалізовані. Розробники повинні побудувати метафори інтерфейсу і запропонувати інтуїтивно зрозумілі схеми, що закликають і запрошують користувачів до прямого маніпулювання об'єктами в інтерфейсі. Джаред спула (Jared Spool) пропонує чотириступінчасту підказку, яка допоможе ефективно користуватися методом «перетягнув і кинув» .Пользовательскій інтерфейс повинен бути наочним, щоб користувачам не доводилося міркувати, як взаємодіяти з тим, що вони бачать на екрані. У інтерфейсах з прямим маніпулюванням не потрібно забувати про масштабованості. Переміщення одного або декількох об'єктів може бути простим і зручним, що не завжди буває, коли працюєш з сотнями об'єктів.

Не потрібно змушувати користувачів діяти прямим маніпулюванням, коли інші методи більш ефективні.

Базовими діями і найпростішими прикладами drag-and-drop дій є: переміщення об'єкта, переміщення об'єкта з панелі в панель, хоча в сучасних операційних системах drag-and-drop набув широкого застосування і є одним з головних способів взаємодії з комп'ютером в графічному інтерфейсі користувача (рис. 4.3).

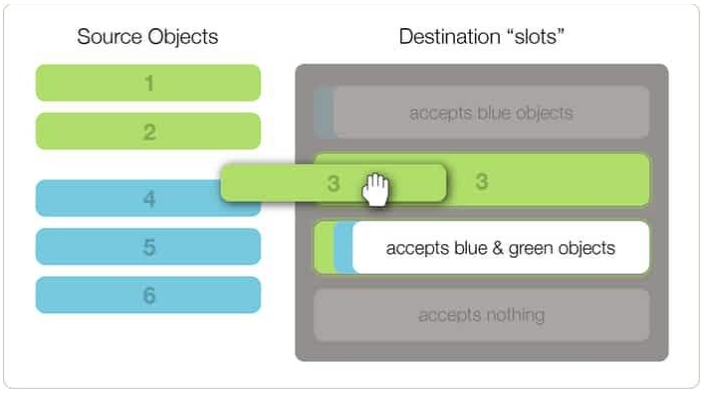

Рисунок 4.3 – Принцип роботи Drag-and-drop технології

Об'єктами для переміщення можуть бути наступні елементи інтерфейсу: значки (іконки) Робочого столу, плаваючі панелі інструментів, ярлики програм в панелі завдань (починаючи з Win XP), елементи TreeView, текстовий рядок, комірка DataGridView., Також елементи OLE. Переміщатися об'єкти можуть як в межах деякої певної області, в межах одного вікна, між панелями одного вікна, так і між різними вікнами.

Основи Drag і Drop використання функціональності drag і drop має на увазі виконання наступних кроків:

Визначити стерпний елемент. Присвоїти true атрибуту draggable елемента, який ми хочемо перенести. Для детальної інформації дивіться The Draggable Attribute.

Визначити дані, які можуть бути переміщені, вони можуть бути різного формату. Наприклад, текстові дані, що містять рядок тексту який може бути переміщений. Для детальної інформації дивіться Drag Data.

Визначити зображення яке буде поруч з покажчиком миші протягом всієї операції перетягування. Якщо для користувача зображення не буде визначено, буде згенеровано картинка за замовчуванням, в залежності від елемента, на якому була затиснута кнопка миші (що означатиме, що елемент переносять). Ознайомитися детально з установкою зображення перетягування можна за посиланням Setting the Drag Feedback Image.

Визначити можливі ефекти переносу. Можливі три таких ефекту: copy показує, що переміщуються дані копіюються з колишнього місця розташування в нове, move показує, що переміщуються дані повністю переносяться на нове місце, і link показує, що створюється якась форма взаємодії або зв'язку між вихідною точкою і точкою призначення. Протягом операції переміщення, картинка яка слідує за курсором миші може змінюватися в залежності від того, чи може елемент бути переміщений в область під курсором. Якщо перенесення дозволе-

ний, переміщення може бути вироблено. Дивіться Drag Effects для детальної інформації.

Визначити область призначення. За замовчуванням браузер не дозволяє переміщати що-небудь на HTML елемент. Однак, щоб зробити елемент активним для переміщення інших елементів на нього, просто скасуйте дію за замовчуванням. Тобто, підпишіться на події "ondragenter" або "ondragover". Для детальної інформації дивіться Specifying Drop Targets.

Обробити завершення перенесення. Ви можете отримати дані з переносного елемента і зробити над ними необхідні операції. Для детальної інформації, будь ласка, дивіться Performing a Drop.

Для того, щоб ознайомитися із загальним списком даних підтримуваних операцією drag and drop дивіться Recommended Drag Types.

Також доступні приклади з кращою практикою використання операції drag and drop для переміщення даних різних типів:

- Text;
- links;
- HTML and XML;
- files;
- images;
- Document Nodes.

Ряд подій спрацьовують протягом всієї процедури drag and drop. Drag-події спрацьовують протягом операції переміщення; події миші, такі як mousemove - немає. Також запам'ятайте, що події dragstart і dragend не спрацьовують при спробі перенести файл з операційної системи в браузер.

Властивість dataTransfer всіх подій переміщення містить дані про всі drag і drop операції.

Dragstart спрацьовує коли елемент почав переміщатися. У момент спрацьовування події dragstart користувач починає перетягування елемента. Оброб-

лювач цієї події може бути використаний для збереження інформації про переміщуваний об'єкті, а також для зміни зображення, яке буде асоційоване з переміщенням. Данное подія не спрацьовує, коли деякий файл буде переноситися з операційної системи в браузер. Для детальної інформації Starting a Drag Operation.

Dragenter спрацьовує, коли переміщуваний елемент потрапляє на елемент-призначення. Оброблювач цієї події показує, що елемент знаходиться над об'єктом на який він може бути перенесений. Якщо ж обробника немає, або він не робить ніяких дій переміщення за замовчуванням заборонено. Ця подія також використовується для того, щоб підсвітити або промаркувати об'єкт над яким відбувається переміщення в разі, якщо переміщення на даний елемент дозволено. Для детальної інформації дивіться Specifying Drop Targets.

Dragover – дана подія спрацьовує кожні кілька сотень мілісекунд, коли переміщуваний елемент виявляється над зоною, яка приймає перетягуються елементи. Для детальної інформації дивіться Specifying Drop Targets.

Drag опускається при переміщенні елемента або виділеного тексту.С цією технологію можливо зробити велику кількість тестів для мобільного додатку:

- Перетягування блоків з правильною відповіддю;
- перетягування блоків з правильною відповіддю в потрібному порядку;
- переятягування слів в приготовленний текс на місця з пропусками.

4.10 Sandbox – рантайм компілятор коду

Інтерпретація – порядковий аналіз, обробка та виконання вихідного коду програми або запиту (на відміну від компіляції, де весь текст програми, перед запуском, аналізується і транслюється в машинний або байт-код, без її виконання).

Можна нескінченно дивитися на вогонь, воду і активність програм, ізольованих в пісочниці. Завдяки віртуалізації ти одним кліком можеш подивитись результат набраного кода.

Втім, віртуалізація застосовується і в дослідницьких цілях: наприклад, захотілося тобі проконтролювати вплив знов скомпільованної програми на систему або запустити дві різні версії додатка одночасно. Або створити автономне додаток, яке не буде залишати слідів в системі. Варіантів застосування пісочниці - безліч. Чи не програма диктує свої умови в системі, а ти їй вказуєш дорогу і розподіляєш ресурси.

Пісочниця – спеціально виділена середовище для безпечного виконання комп'ютерних програм. Зазвичай являє собою жорстко контрольований набір ресурсів для виконання гостьової програми - наприклад, місце на диску або в пам'яті. Доступ до мережі, можливість спілкуватися з головною операційною системою або зчитувати інформацію з пристроїв введення зазвичай або частково емулюють, або сильно обмежують. Пісочниці представляють собою приклад віртуалізації.

Простий інтерпретатор аналізує і тут же виконує (власне інтерпретація) програму покомандно (або через підрядник), у міру надходження її вихідного коду на вхід інтерпретатора. Перевагою такого підходу є миттєва реакція. Недолік - такий інтерпретатор виявляє помилки в тексті програми тільки при спробі виконання команди (або рядка) з помилкою.

Інтерпретатор компілюючего типу – це система з компілятора, який переводить вихідний код програми в проміжне представлення, наприклад, в байт-код або р-код, і власне інтерпретатора, який виконує отриманий проміжний код (так звана віртуальна машина). Перевагою таких систем є більшу швидкодію виконання програм (за рахунок виносу аналізу вихідного коду в окремий, разовий прохід, і мінімізації цього аналізу в інтерпретаторі). Недоліки – більша вимога до ресурсів і вимога на коректність вихідного коду. Застосовується в таких

мовах, як Java, PHP, Tcl, Perl, REXX (зберігається результат парсинга вихідного коду), а також в різних СКБД.

За рівної кількості інтерпретатора компілює типу на компоненти виходять компілятор мови і простий інтерпретатор з мінімізованих аналізом вихідного коду. Причому вихідний код для такого інтерпретатора не обов'язково повинен мати текстовий формат або бути байт-кодом, який розуміє тільки даний інтерпретатор, це може бути машинний код якийсь існуючої апаратної платформи. Наприклад, віртуальні машини на зразок QEMU, Bochs, VMware включають в себе інтерпретатори машинного коду процесорів сімейства x86.

Деякі інтерпретатори (наприклад, для мов Лисп, Scheme, Python, Бейсік і інших) можуть працювати в режимі діалогу або так званого циклу читання-обчислення-друку (англ. Read-eval-print loop, REPL). У такому режимі інтерпретатор зчитує закінчену конструкцію мови (наприклад, s-expression в мові Лісп), виконує її, друкує результати, після чого переходить до очікування введення користувачем наступної конструкції.

5 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Проектування діаграми прецедентів

Прецеденти [13][1)] – це текстовий документ, що описує послідовність подій, пов'язаних з використанням системи. UML діаграма класів знаходиться у додатку В.

Діаграма прецедентів – в UML, діаграма, на якій зображений відношення між акторами і прецедентами в системі.

Актор – це набір ролей, який виконує користувач в ході взаємодії з деякою сутністю (система, підсистема, клас).

На діаграмі прецедентів ілюструється набір прецедентів системи та виконавці, а також взаємозв'язку між ними. Прецеденти визначають, як виконавці взаємодіють з програмною системою. У процесі цієї взаємодії виконавцем генеруються події, які передаються системі, які представляють собою запити на виконання деякої операції. Як правило, окремі кроки або види діяльності у вигляді прецеденту не представляються.

При створенні діаграм використання спочатку визначаються виконавці (ролі, користувачі). Виконавець (actor) є зовнішнім по відношенню до системи поняттям, яке певним чином бере участь у процесі, описуваному прецедентом. Актор є якоюсь ідеалізацією намірів користувача, а не самого користувача. Один реальний користувач може використовуватися декількома акторам, а один актор може представляти один і той же намір відразу декількох користувачів.

[1)] [13] Диаграмма прецедентов (вариантов использования) UML. URL: https://planerka.info/item/diagramma-precedentov-variantov-ispolzovaniya-uml/ (Дата звернення 05.11.2019).

Поведінка, що розробляється (тобто функціональність, що забезпечується системою) описується за допомогою моделі прецедентів, яка відображає системні прецеденти (use cases), системне оточення (дійових осіб або акторів – actors) і зв'язку між прецедентами і акторами (діаграми прецедентів – use cases diagrams). Головною метою даної моделі є реалізація собою єдиного засобу для взаємодії розробника з замовником та кінцевим користувачем для обговорення функціональної складової і поведінки системи.

На основі цього типу діаграм проводиться ітераційний цикл загальної постановки задачі разом з замовником. Оскільки замовник ніколи не буде точно знати чого він хоче, то діаграми прецедентів якраз і є основою для досягнення взаєморозуміння між програмістами-професіоналами, які розробляють проект, і "бізнесменами"-замовниками проекту. Ці діаграми описують функціональність ІС, яка буде видна користувачам системи, основні функції, які повинні бути включені в систему (use case), їх оточення (actors). "Кожна функціональність" зображується у вигляді "прецедентів використання" (use case) або просто прецедентів. Прецедент – це типова взаємодія користувача з системою, при цьому:

− Описує видиму користувачем функцію;

− може представляти різні рівні деталізації;

− забезпечує досягнення конкретної мети, важливої для користувача.

Актори не є частиною системи, вони використовують систему (або використовуються системою) в даному прецеденті.

Актори можуть:

− Тільки постачати інформацією систему;

− тільки отримувати інформацію з системи;

− забезпечувати інформацією і отримувати інформацію з системи.

Актор, який представляє людини-користувача, характеризується роллю в даному прецеденті. На діаграмі зображується тільки один актор, однак, реальних користувачів, які виступають в цій ролі по відношенню до ІС, може бу-ти

багато. Актори визначаються на основі складеного завдання або у процесі обговорення з замовником.

Спроектована діаграма прецедентів згідно додатку, що розроблюється, зображена на рис. 5.1.

У даному випадку у системі присутній один рівень доступу для користувачів оскільки недоцільно розмежовувати дану інформацію, адже вона знаходиться у громадському доступі. Кожен користувач зможе скористатися усіма можливими функціями додатку.

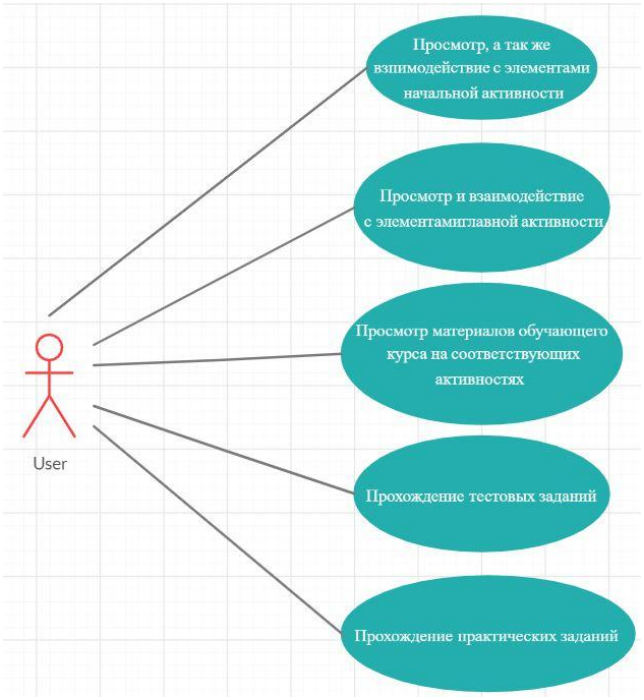

Рисунок 5.1 – Діаграма прецедентів

5.2 Функціональна схема додатку

Компоненти програми можна віднести до одного з чотирьох типів. Компоненти кожного типу призначені для певної мети, вони мають власний життєвий цикл, який визначає спосіб створення і припинення існування компонента.

Існують чотири стандартні блоки додатку Android:

− Activity (активність);

− Intent Receiver;

− Service (служба);

− Content Provider;

Не кожен додатку повинен містити всі чотири блоки, але він буде використовувати деяку їх комбінацію.

Коли розробник вирішує які компоненти необхідно створювати для додатку, їх необхідно записати у файлі на ім'я AndroidManifest.xml. Це – файл XML, де оголошуються елементи додатку, а також їхні можливості і вимоги.

Activity (активність) – основний з вище перелічених блоків Андроїд. Активність являє собою один повноцінний екран додатку, зазвичай – єдиний. Кожна активність здійснена як єдиний клас, який розширює базовий клас Activity. Клас відображає призначений для користувача інтерфейс, складений з Views на її макеті, і відповідає на події ініційовані користувачем. Більшість додатків складається з декількох екранів. Переміщення в інший екран досяга-ється стартом нової активності. Коли новий екран відкривається, попередній екран припиняється і поміщається у стек хронології. Користувач може перей-ти на попередню активність перемістившись назад через раніше відкриті акти-вності в хронології. Андроїд зберігає стеки хронології для кожної програми.

Незважаючи на те що Activity спільно формують чітку взаємодію користувача з додатком, кожна з них не залежить від інших операцій. Будь-які з цих операцій можуть бути запущені іншим додатком

Для організації переходу від одного екрану до іншого Андроїд використовує спеціальний клас, що називається Intent. Він описує те, що додаток збирається зробити. Дві найважливіших частині структури Intent – дія і дані до дії. Intent може містити наступні значення для дії: MAIN (головний екран до-датка), VIEW, PICK, EDIT, і т.д. Дані виражені у вигляді Uniform Resource Indicator

(URI) – послідовність символів, що ідентифікує логічний чи фізич-ний ресурс. Приклади ресурсів включають електронні документи, датчики дверей ліфту, простір імен XML, веб-сторінки та ідентифікаційні мікрочипи для домашніх тварин.

Існує пов'язаний клас, під назвою IntentFilter. Якщо Intent являє собою запит на якусь дію, то IntentFilter є описом того, що Intent Activity, здатний обробити. Активність, яка в змозі відобразити інформацію для користувача, видає IntentFilter, який повідомляє, що може зробити VIEW і містить інструк-ції для обробки. IntentFilters задається в файлі AndroidManifest.xml до відпо-відної активності.

Переміщення між екранами додатку досягається за допомогою Intent. Щоб перемістити вперед, активність викликає startActivity(myIntent). У та-кому випадку система звертається до IntentFilter всіх встановлених додатків і вибирає активність, Intent якої відповідає myIntent. Викликаній активності повідомляється про Intent, який змусив її початися. Процес виконання Intent відбувається, в той момент коли викликається метод startActivity.

Користувальницькі інтерфейси в Андроїд можуть бути створені двома шляхами, через XML-код або в java-коді. Створення структури графічного інтерфейсу користувача в XML переважно, тому що за принципом зразкового управління засобами перегляду, користувальницький інтерфейс повинен завжди бути відокремлений від логіки програми.

Основний функціональний модуль додатки Android – Activity – об'єкт класу android.app.activity. Activity відповідає за функціональну складову додатку і не має окремої присутності на екрані. Щоб дати Activity присутність на екрані і проектувати його користувальницький інтерфейс, необхідно працюва-ти з Views і Viewgroups – основними одиницями графічного представлення візуальних компонентів користувальницького інтерфейсу на платформі Анд-роїд.

View – об'єкт, який розширює базовий клас android.view.view. View є структурою даних і його властивості знаходяться у макеті. View безпосеред-ньо відповідає за інформаційне наповнення для певного контейнеру макету. Об'єкт View обробляє вимір, його схему розміщення, малюнок, зміни центру, прокрутку, і клавіші / знаки для області екрану, яку він представляє. Клас View слугує базовим класом для всіх графічних фрагментів – ряд підкласів, які малюють інтерактивні елементи екрану. Серед дочірніх графічних об’єктів класу View знаходяться такі як: текстові представлення у вигляді статичного тексту TextView та рядку для введення тексту EditText, звичайна кнопка Button, кнопка-перемикач, яка дозволяє користувачеві обрати один з пунктів (опцій) з визначеного набору, RadioButton, кнопка-прапорець Checkbox, ScrollView і т.д.

Viewgroup – об'єкт класу android.view.viewgroup. Viewgroup – спеціальний тип об'єкту View, функція якого – утримувати набор View і Viewgroup і керувати ними. Viewgroups дозволяють додавати структуру до користувальницького інтерфейсу і створювати складні елементи екрану, до яких можна звернутися як до єдиного об'єкту. Клас Viewgroup слугує базовим класом для Layouts – ряду повністю здійснених підкласів, що забезпечує загальні типи Layouts екрану. Layouts дають можливість вбудувати структуру для ряду View.

На платформі Андроїд визначається користувальницький інтерфейс Activity використання дерева View і Viewgroup вузлів, як показано на рис. 5.2. Дерево може бути настільки ж простим або складним, як зробить розробник. Можна його побудувати, використовуючи набори зумовлених графічних фрагментів і Layouts Андроїда, або замовних типів View, які створюються самостійно.

Щоб прикріпити дерево до екрану і візуалізувати його, Activity викликає свій метод setContentView() і передає інформацію на кореневий об'єкт вузла. Як тільки система Андроїд отримує інформацію на кореневий об'єкт вузла, вона починає працювати безпосередньо з вузлом, щоб виміряти, і окреслити дерево.

Коли Activity стає активним і отримує пріоритет, система реєструє його і звертається до кореневого вузла задля виміру і окреслення дерева. Тоді кореневий вузол надає запит, щоб його дочірні вершини окреслили себе – у свою чергу, кожен Viewgroup вузол в дереві відповідальний за окреслення його прямих дочірніх вузлів.

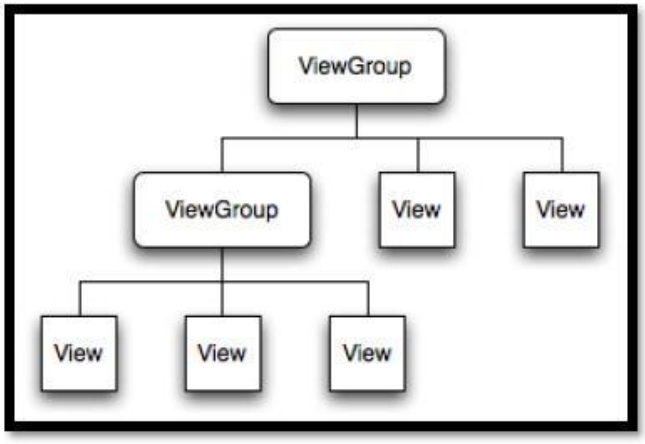

Рисунок 5.2 – Деревоподібна структура користувальницького інтерфейсу операційної системи «Android»

Як згадано раніше, у кожної групи View є відповідальність вимірювання її доступного простору, розташування її дочірніх вузлів, і виклик draw() на кожному дочірньому вузлі, щоб дозволити їм візуалізувати себе. Дочірні вузли можуть запрошувати розмір і місце розташування в батьківського, але у батьківського об'єкта є кінцеве рішення, де і наскільки великий кожен спадкоємець може бути.

Даний додаток використовує 5 активності:

– «MainActivity» –активність, яка містить стартову сторінку, яка має кнопку переходу до головної активності.

– «MainMenu» – головна активність, що містить у собі мапу з темами завдань. При натисканні на окрему тему буде відкриватися активність з теоретичною частиною навчання;
– «Practice» –активність, що містить у собі фрагмент, на якому будуть знаходитись практичні завдання;
– «Content_scroller» – активність, яка має на собі мінливі фрагменти, на яких є різноманітні блоки с теоретичною частиною;
– «TestActivity» – активність, яка має на собі мінливі фрагменти, на якіх є різноманітні блоки с тестовими завданнями;

Крім того, додаток містить відповідні макети:

– «activity_main» – макет, що відповідає за формування головної сторінки;
– «main_menu» – макет, що відповідає за формування сторінки «MainMenu»;
– «content_scroller_layout» – макет, що відповідає за формування сторінки «Content_scroller»;
– «practice_layout» – макет, що відповідає за формування сторінки «Practice»;
– «test_activity_layout» – макет, що відповідає за формування сторінки «TestActivity».

Також додаток містить 3 фрагмента, які поміщуються до вкладок активностей «Content_scroller», «Practice», «TestActivity»:

– «Test_fragment» – фрагмент, який оброблює тестову частину навчання;
– «Scroll_fragment» – фрагмент, який оброблює теоретичну частину навчання;
– «Practice_fragment» – фрагмент, який оброблює практичну частину навчання;

А також відповідне кількість макетів, та 4 класа адаптерів.

5.2 Алгоритм роботи додатку

При запуску додатку відкривається активність, яка містить кнопку завдяки якою починається робота додатка. Після чого відкривається головна активність додатку, на якій розташовується мапа з різноманітними темами, за якими відкривається блок з навчальним матеріалом. При натисканні на окремі частини мапи відкривається наступна активність на якої розташовується окремий фрагмент, індивідуальній для кожної теми з теоретичним блоком. Після того, як теоретична частина буде оброблена, наприкінці цієї розташовується кнопка, при натисканні на яку відкривається наступна активність. Кожна тема має як декілька тестових завдань, так і практичне завдання. Активність, яка має тестові завдання зроблена на подобі вікторини, де э питання, та 2 варіанта відповіді, один з яких вірний. Зокрема тестових завдань, активність має кнопку «Назад», при натисканні на яку можливо перейти на попередню активність з теоретичним блоком. Також на цієї активності є лічильник питань, вірні відповіді мають зелений колір, а не вірні – червоний. Відповіді мають собою дві картки з зображенням, та підписом під кожної з них. При натисканні на відповідь, якщо вона вірна, з’являється характерне зображення, так само, якщо б була обрана не вірна відповідь.

Наступна активність з практичним завданням має текст завдання, зображення з завданням, там блок, у який потрібно внести зміни. Після внесення змін треба натиснути на кнопку у низу вікна «Проверить», після чого введенні данні практичного завдання пройдуть обробку та в блоку з завданням з’являться зміни та вікно Dialog яке попередить, якщо завдання виконане вірно, чи ні. Клас Dialog є базовим для всіх класів діалогових вікон. Оскільки ProgressDialog,

TimePickerDialog I DatePickerDialog - розширення класу AlertDialog, вони також можуть мати командні кнопки [14][1)].

Кожне діалогове вікно повинно бути визначено всередині активності, в якій буде використовуватися. Діалогове вікно можна відкрити один раз або кілька разів.

5.3 Розробка зовнішнього вигляду додатку

Розробка зовнішнього вигляду та дизайну додатку відбувалась виключно дотримуючись стандартів та канонів цієї області. В першу чергу орієнтованість направлена на Material design [15][2)]. Це мова візуальних образів, який створила корпорація Google для уніфікації інтерфейсів її продуктів і сервісів.

Material design грунтується на принципах друкованого дизайну. І не тільки для краси, а й для розстановки акцентів і фокусування уваги користувача на потрібному елементі, для спрощення навігації серед ієрархії конструкцій інтерфейсу, для інтуїтивної передачі їх сенсу. На діях користувача сфокусовано основну увагу. Взаємодією з дизайном керує призначений для користувача досвід, а не навпаки. Всі дії відбуваються в одному оточенні, інтерактивні об'єкти без переривання переходять з однієї середи в іншу.

Крім того додаток розроблювався відповідно вимогам зручності та використовуємості користувальницького інтерфейсу, базуючись на UX-дизайні та естетиці зовнішнього вигляду.

Нижче приведені рис. 5.3 та 5.8, на яких зображена візуальна частина додатку.

[1)] [14] Android developers. URL: https://developer.android.com/?hl=ru (Дата звернення 09.11.2019).

[2)] [15] Material Design for Android. URL: https://developer.android.com/guide/topics/ui/look-and-feel (Дата звернення 12.11.2019).

На рис. 5.3 зображено головну активність додатку, яка являє собою активність з боковим меню, якому присвоєна роль навігації по додатку. Ця мапа являє собою рівневу структуру, яка починається з простих тем та чим нижче спускатися – теми будуть складніші. Весь дизайн додатку виконан у авторському стилі

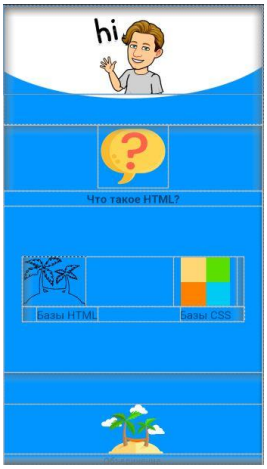

Рисунок 5.3 – Головна активність додатку: мапа рівнів

На рис. 5.4 зображено головну активність, яка вітається з користувачем та пропонує перейти на наступну активність. Також присутня кнопка та зображення.

Рисунок 5.4 – Головна активність додатку: фрагмент з завданнями

На рис. 5.5 зображена активність, яка вміщую в себе фрагменти, на яких розміщується теоретична частина обраної теми. Також знизу є кнопка переходу до наступної активності.

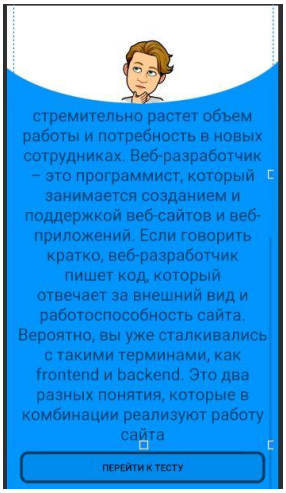

Рисунок 5.5 – Активність з теоретичною частиною

На рис. 5.6 зображено активність з тестовим завданням, на якій розташован фрагмент з індивідуальним до кожної теми завданням. Тестове завдання - це частина складного (складеного) тесту, по якій випробуваний в ході виконання тесту робить окрему дію, а його результат реєструються в первинному протоколі відповіді.

Рисунок 5.6 – Активність з тестовими завданнями

На рис. 5.7 одно з вікон Dialog. На ньому зображено вітальне сповіщення. У деяких випадках потрібно показати діалогове вікно, де користувачу потрібно зробити якийсь вибір або показати повідомлення про помилку. Безумовно, можна створити власне вікно, розташувати в ньому потрібні кнопки і обробляти їх натискання. Але, в Android вже є власні вбудовані діалогові вікна, які гнучко налаштовуються під завдання. Використання діалогових вікон для простих завдань дозволяє скоротити число класів Activity в додатку, економлячи ресурси пам'яті. Адже вам не доведеться реєструвати активність в маніфесті, думати над компонуванням елементів на екрані і так далі. Діалогові вікна в Android є напівпрозорі «плаваючі» активності, частково перекривають батьківський екран, з якого їх викликали.

Рисунок 5.7 – Діалогове вікно

На рис. 5.8 зображено активність, яка розміщує на собі фрагмент з практичним завданням. Практичне заняття – це форма організації навчального процесу, що передбачає дотримання учнями за завданням і під керівництвом викладача однієї або декількох практичних робіт.

Рисунок 5.8 – Активність з практичним завданням

5.6 Розробка структури додатку

Головною активністю проекту на якій знаходиться навігаційну мапу і виконується основний функціонал, є активність MenuActivity. Вона базується на використанні RelativeLayout. ReiativeLayout (відносна розмітка) знаходиться в розділі Layouts і дозволяє дочірнім компонентів визначати свою позицію щодо батьківського компонента або щодо сусідніх дочірніх елементів (ідентифікатора елемента). У RelativeLayout дочірні елементи розташовані так, що якщо перший елемент розташований по центру екрана, інші елементи, вирівняні щодо першого елемента, будуть вирівняні відносно центру екрана. При такому розташуванні, при оголошенні розмітки в XML-файлі, елемент, на який будуть посилатися для позиціонування інші об'єкти уявлення, повинен бути оголошений раніше, ніж інші елементи, які звертаються до нього за його ідентифікатором. Окрім цього, на активності є декілька LinearLayout, які виступають в ролі столбців, та строк. У студії макет LinearLayout представлений двома варіантами - Horizontal

і Vertical. Макет LinearLayout вирівнює всі дочірні об'єкти в одному напрямку - вертикально або горизонтально. Напрямок задається за допомогою атрибута орієнтації android: orientation. Також, щоб уся мапа була доступна на екрані, потрібен макет ScrollView. При великій кількості інформації, яку потрібно помістити на екрані доводиться використовувати смуги прокрутки. В Android існують спеціальні компоненти ScrollView і HorizontalScrollView, які є контейнерними елементами і успадковуються від ViewGroup.

ВИСНОВКИ

Аналіз літератури, присвяченої застосуванню мобільних додатків в навчанні, переконливо показує, що в цій ідеї укладено великі можливості для модернізації навчання, тобто для вдосконалення процесу навчання шляхом виконання нових вимог, що пред'являються в даний час до вузів.

Таким чином, активне і планомірне використання мобільної технології у ВНЗ це природний процес соціалізації і в той же час близьке студентам інформаційне середовище. При впровадженні мобільних технологій в навчанні різко збільшується потік затребуваною інформації, активізуються можливості студентів, підвищується ефективність самостійної роботи. Все це дозволить підвищити якість професійної освіти.

У ході створення мобільного додатку для навчання Веб-дизайну в ігровій формі були виконані наступні дії:

- Розглянуті, проаналізовані та обрані програмні засоби для реалізації додатку;
- проаналізована та описана предметна область;
- розглянуті програмні аналоги, що існують на ринку;
- проаналізована та описана предметна область;
- підтверджена актуальність додатку;
- проаналізовано основних методологічних підходів навчання;
- встановлено мету;
- змодельована робота системи через UML діаграму прецедентів;
- розроблена оригінальна візуальна частина додатку;
- розроблена функціональна частина додатку;
- протестовано роботу додатку;

Для реалізації програмного забезпечення було проведено об’ємну роботу по взаємодії між елементами середовища Android Studio. Для цього будо вико-

ристана допоміжна література. За допомогою інструментів, що надаються Android Studio, був разроблен дизайн, та структура додатку.

Перелік джерел посилання

1. Мобільний навчання: плюси і мінуси. URL: https://kogio.ru/blog/m-learning/ (Дата звернення 12.09.2019).

2. Kumari Madhuri, Vikram Singh, Mobile Learning: An Emerging Learning Trend - HiTech Whitepaper ,11,2009 (Дата звернення 12.09.2019).

3. Swift Playgrounds. URL: https://www.apple.com/swift/playgrounds/ (Дата звернення 17.09.2019).

4. Code Academy. URL: https://www.codecademy.com (Дата звернення 17.09.2019).

5. Lrn education. URL: https://lrn.com/education/ (Дата звернення 17.09.2019).

6. Tynker – coding to kids. URL: https://www.tynker.com (Дата звернення 17.09.2019).

7. Khan Academy. URL: https://ru.khanacademy.org (Дата звернення 17.09.2019).

8. Аналітичний огляд і порівняння можливість операційних систем для мобільних пристроїв URL: https://www.fundamentalresearch.ru/ru/article/ view?id=39079/ (Дата звернення 24.09.2019)

9. Android – Википедия. URL: https://ru.wikipedia.org/wiki/Android (Дата звернення 24.09.2019).

10. Архитектура Android приложений. URL: https://habr.com/ru/post/278815/ (Дата звернення 24.09.2019).

11. Android Studio 1.0: первая стабильная IDE от Google URL: https://habr.com/ru/company/vcstart/blog/364117/ (Дата звернення 26.09.2019).

12. Genymotion. URL: https://android-emulator.net/genymotion (Дата звернення 05.10.2019).

13. Диаграмма прецедентов (вариантов использования) UML. URL: https://planerka.info/item/diagramma-precedentov-variantov-ispolzovaniya-uml/ (Дата звернення 05.11.2019).

14. Android developers URL: https://developer.android.com/?hl=ru (дата звернення 09.11.2019).

15. Material Design for Android URL: https://developer.android.com/guide /topics/ui/look-and-feel (дата звернення 12.11.2019).

Д О Д А Т К И

ДОДАТОК А
ЖИТТЄВИЙ ЦИКЛ ANDROID ДОДАТКУ

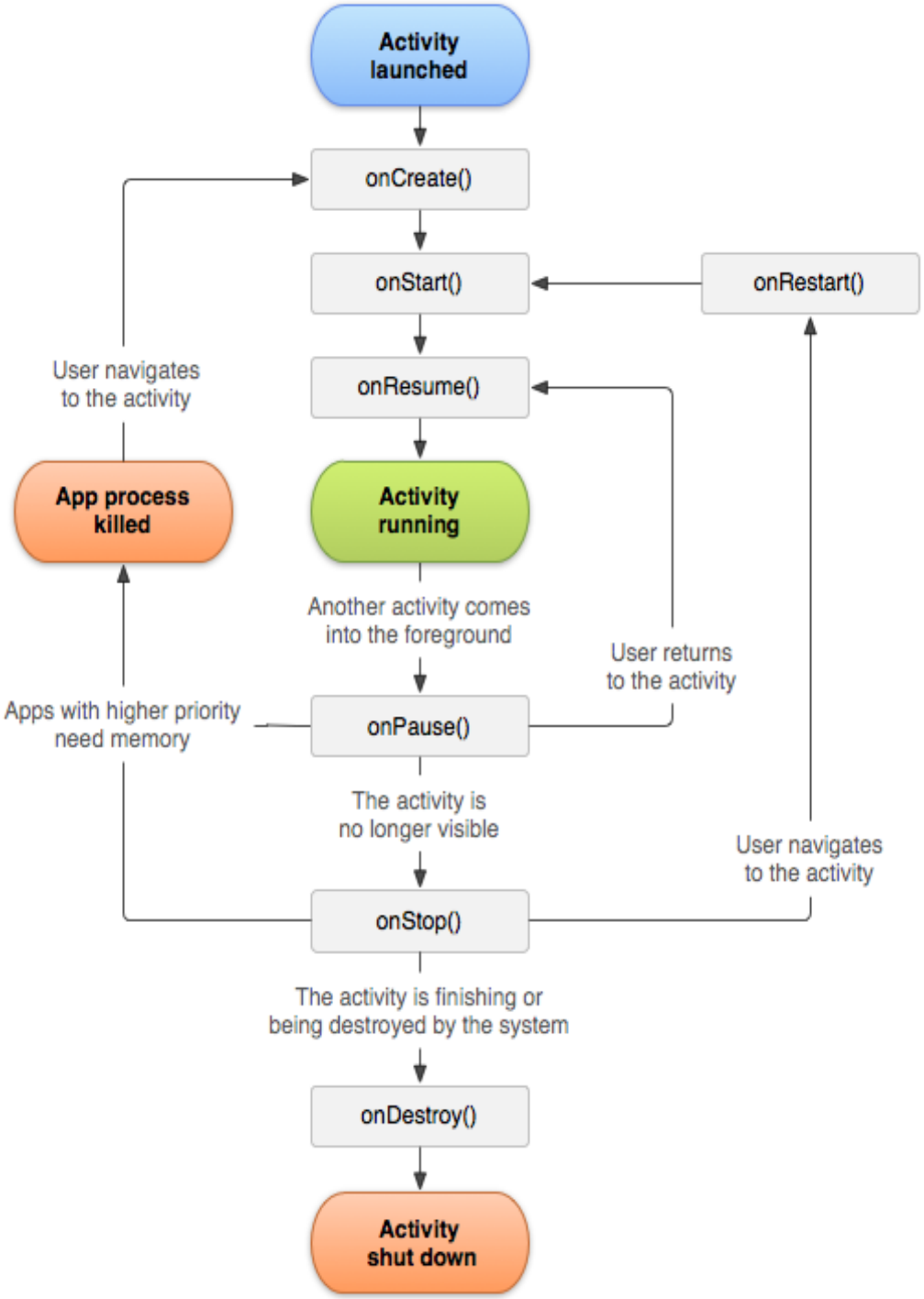

Рисунок А.1 – Життєвий цикл роботи android додатку у мобільної ОС Android

ДОДАТОК Б
UML ДІАГРАМА КЛАСІВ ANDROID ДОДАТКУ

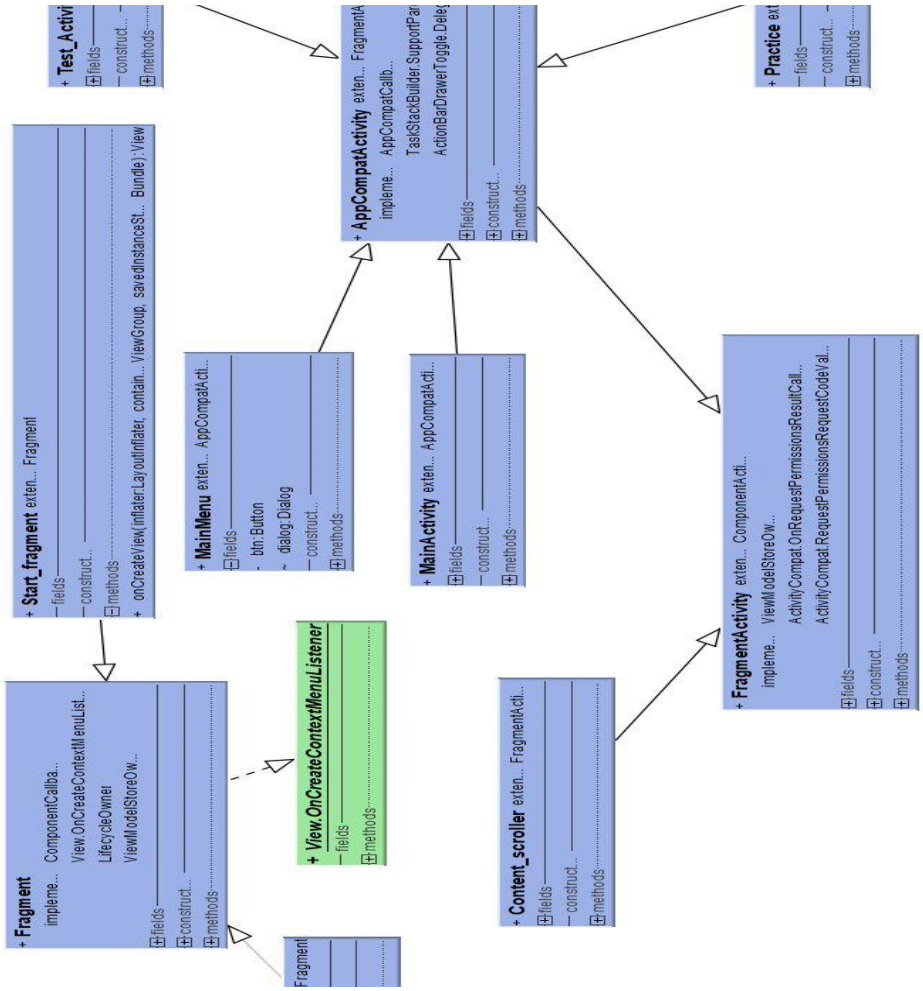

Рисунок Б.1 – UML диаграма класів додатку з Android Studio

ДОДАТОК В
UML ДІАГРАМА ГЕНЕРАЦІЇ ТЕСТОВОГО ЗАВДАННЯ

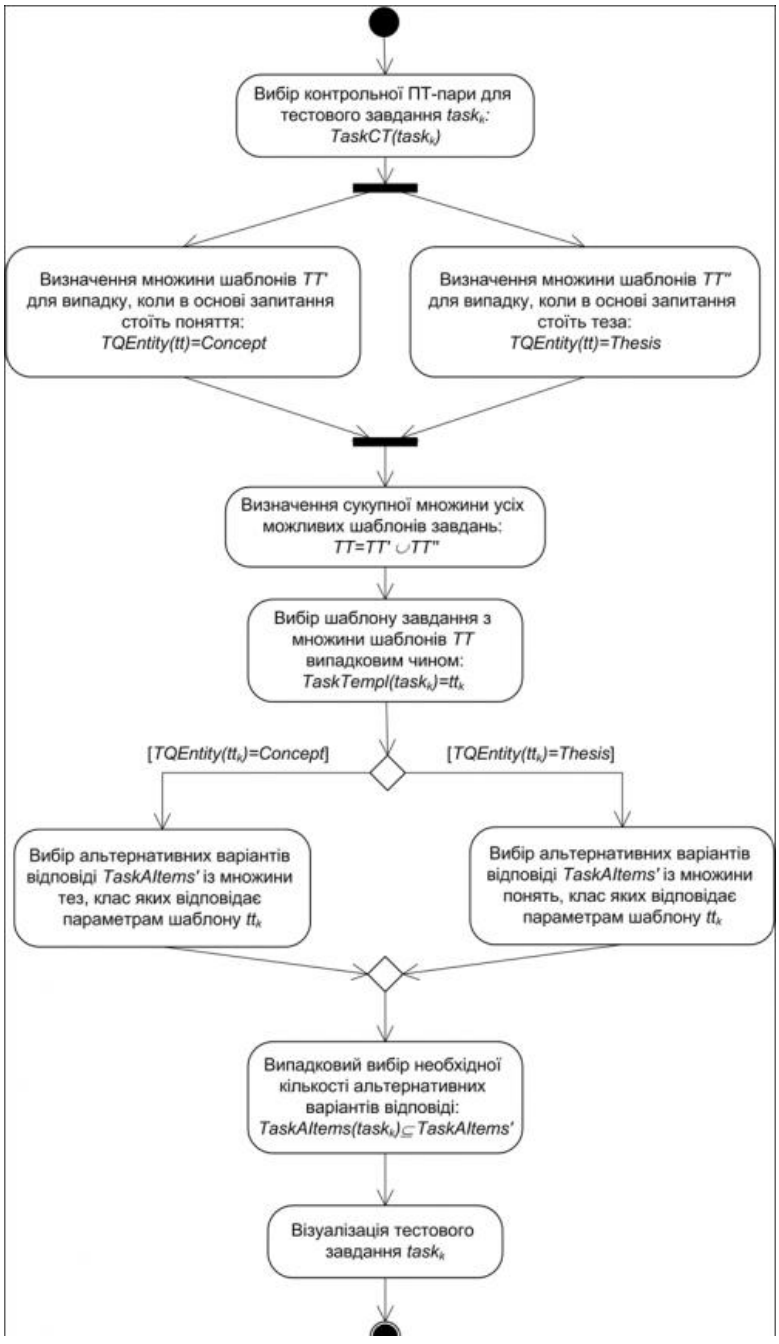

Рисунок В.1 – Генерація тестового завдання. Діаграма діяльності у нотації UML