МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської та аспірантської підготовки
Кафедра Комп'ютерних наук

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інтеграція мобільних технологій в систему підтримки навчального процесу»

Виконав студент 2 курсу групи МК-2
спеціальності 122 Комп'ютерні науки
Грибова Ольга Олегівна

Керівник к.геогр.н.,доцент Кузніченко Світлана Дмитрівна

Консультант

Рецензент к.т.н.,доцент Гнатовська Ганна Арнольдівна

Одеса 2018

АНОТАЦІЯ

на магістерську роботу

«Інтеграція мобільних технологій в систему підтримки навчального процесу»

Об’єкт дослідження – мобільні інформаційні технологій і процес їх інтеграції в систему підтримки навчального процесу.

Предмет дослідження – методи проектування і розробки мобільних застосувань для підготовки студентів до контролюючих заходів.

Методи дослідження: UML моделювання, об’єктно-орієнтоване моделювання, концептуальне моделювання.

Метою магістерської роботи є теоретичне обґрунтування ефективності інтеграції мобільних технологій в систему підтримки навчального процесу та розробка мобільного застосування для самостійної роботи та підготовки студентів до контролюючих заходів.

Актуальність дослідження обумовлена необхідністю вирішення проблем підвищення якості освіти шляхом вдосконалення підходу підготовки студентів до різного роду контролюючих заходів з використанням мобільних технологій.

Результатом виконання магістерської роботи є розроблене мобільне застосування для підготовки студентів до іспитів, залікових, контрольних і модульних робіт. Функціональність застосування передбачає можливість проходження різних типів тестів, перегляду статистики результатів тестування, додавання термінів в особистий словник, спілкування з одногрупниками і викладачами тощо.

Структура магістерської роботи складається з анотації, вступу, чотирьох розділів, висновків, переліку посилань на 18 найменувань. Повний обсяг роботи становить 59 сторінок і містить 24 рисунки.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ТЕСТУВАННЯ, ДИСТАНЦІЙНЕ НАВЧАННЯ, ПЛАТФОРМА ANDROID.

ANNOTATION

For master's research

" Integration of Mobile Technologies into the Educational Support System"

The object of research - the integration of mobile information technologies into the educational process support system.

The subject of research - methods of designing and developing mobile applications that prepare students for control activities.

Research Methods: UML Modeling, Object-Oriented Modeling, Conceptual Modeling.

The purpose of this master's work is the theoretical substantiation of the efficiency of integration of mobile technologies in support of the educational process and the development of mobile applications for independent work while preparing students for supervisory activities.

The urgency of this study is due to the need to address problems to increase the quality of education by improving the approach of preparing students for various kinds of controlled activities using mobile technologies.

The result of this master's work is the development of a mobile application to prepare students for examinations, tests, as well as controlled and modular work. The functionality of the application includes the possibility of passing various types of tests and reviewing statistics based on the test results, adding terms to a personal dictionary, communicating with classmates and teachers, etc.

The structure of the master's thesis consists of an abstract, an introduction, four sections, conclusions, a list of references for 18 titles. The full work volume is 59 pages long and it contains 24 drawings.

Keywords: INFORMATION SYSTEM, TESTING, DISTANCE LEARNING, ANDROID PLATFORM.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ 10
ВСТУП..... 12
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ 14
1.1 Мобільне навчання як нова форма організації освітнього процесу 14
1.2 Переваги і недоліки використання мобільних додатків в освіті 18
1.3 Аналіз подібних та існуючих систем підтримки навчального процесу . 20
2 ПРОГРАМНІ ЗАСОБИ ДЛЯ РОЗРОБКИ ANDROID-ДОДАТКУ 27
2.1 Опис операційної системи Android 27
2.2 Загальна схема роботи програми Android 29
2.3 Аналіз сучасних засобів розробки мобільних додатків 30
2.4 Вибір засобів реалізації мобільної системи підтримки навчального процесу 33
3 ПРОЕКТУВАННЯ СИСТЕМИ 36
3.1 Середа розробки Android Studio 36
3.2 Порівняння реляційних та нереляційних БД 38
3.3 Система керування базами даних Firebase 43
3.4 Проектування БД..... 45
3.5 Архітектура системи підтримки навчального процесу..... 47
3.6 UML-діаграма класів 51
3.7 Діаграма варіантів використання 52
4 РОБОТА ДОДАТКУ 55
4.1 Опис функціональності 55
ВИСНОВКИ..... 65
ПЕРЕЛІК ПОСИЛАНЬ 66

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.

MCQs (Multiple Choice Questions) – запитання на множинний вибір.

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 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 і КПК. Технологія мобільного навчання є наступною стадією розвитку технології електронного навчання E-Learning, яка передбачає наявність системи дистанційного навчання, що включає в себе підсистему доступу до навчальних матеріалів і сервісів з різних мобільних пристроїв, а також наявність web-доступу.

Використання мобільних технологій для підтримки навчального процесу буде ефективним, якщо:

– виявлена сукупність умов інтеграції мобільних технологій в систему підтримки навчального процесу;

– досліджені сутнісні та структурно-функціональні характеристики системи мобільного підтримки навчального процесу, яка буде розглянута як певна частина відкритого дистанційної освіти та використовує освітні ресурси нового покоління та мобільні пристрої та технології;

– розроблені технологічні основи підтримки навчального процесу, орієнтовані на використання мобільних технологій учнями та викладачами.

Актуальність дослідження обумовлена необхідністю вирішення проблем підвищення якості освіти, що визначається сукупністю показників, які характеризують різні аспекти діяльності вищих навчальних закладів. Вдосконалення підходу підготовки студентів до різного роду контролю знань.

Метою магістерської роботи є теоретичне обґрунтування та практична інтеграція мобільних технологій в систему підтримки навчального процесу, а саме створення мобільного додатку, який дозволятиме студентам підготовлюватися до складання екзаменів, заліків та будь-яких контрольних робіт. 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 підтримки навчального процесу;

– обґрунтувати вибір програмних засобів та технологій для створення веб-застосування;

– виконати проектування системи за допомогою засобів UML;

– розробити мобільний додаток, що міститиме наступні функціональні можливості: внесення і зберігання інформації про користувачів в базі даних; ведення статистики кожного користувача; можливість здійснення переписки між користувачами; надання дисциплін і їх тем для підготовки до контролю знань; додавання нових термінів до словнику;

– перевірити працездатність системи;

– розробити керівництво користувача.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Мобільне навчання як нова форма організації освітнього процесу

Термін «мобільне навчання» з'явився більш десяти років тому, і більшість його трактувань грунтувалося на технологічних особливостях переносних пристроїв і дидактичних можливостях їх використання в навчанні.

Згідно поточним тенденціям, наступна фаза розвитку електронного навчання буде зосереджена на мобільних тренуваннях. Мобільне навчання – це та точка, в якій перетинаються мобільні комп'ютерні середовища і електронне навчання, утворюючи в результаті практику навчання в будь-який час, в будь-якому місці.

З відходом в мобільну сферу освіта адаптується до умов цього середовища, стаючи більш компактною, вузьконаправленою і інтерактивною. У мобільному інтернеті важливо, щоб інформація могла бути засвоєна невеликими шматками, максимально чітко відповідала ситуації і, при цьому, щоб процес взаємодії з продуктом приносив задоволення. Відповідно, змінюються звички користувачів: учні все більше хочуть, щоб освіта була простою та ефективною.

Можливість впровадження мобільного навчання в освітній навчальний процес обгрунтовано тим, що воно задовольняє основним принципам навчання, таким як:

– принцип науковості та об'єктивності здійснюється за рахунок доступу до всесвітньої мережі і постійно поповнюється, оновлюється науковими даними відповідних сучасному стану наук і ґрунтується на положеннях, відповідних фактів;

– принцип зв'язку теорії з практикою здійснюється за рахунок застосування вбудованих датчиків і сенсора в портативних пристроях при проходженні навчання в польових умовах, в лабораторних роботах; за рахунок мобільного доступу та інтерактивності до необхідної літератури;

– принцип послідовності і систематичності здійснюється за рахунок можливості автоматизувати збір і аналіз інформації про успішність учнів за допомогою мобільних технологій, що надає педагогу додаткові дані і вільний час для подальшого планування процесу навчання і аналізі знань учнів. Даний принцип вимагає створення системи і логічній послідовності навчання;

– принцип доступності здійснюється за рахунок надання викладачем широкого вибору дидактичного матеріалу для індивідуального освоєння в

залежності від їх рівня освоєння предмета, з огляду на інтереси того, хто навчається, при цьому доступ до даних і програми мобільного навчання можуть допомогти в організації індивідуального підходу в навчанні, а так само підвищити мотивацію учнів (студентів) у вивченні предмета;

– принцип наочності здійснюється за рахунок активного використання мультимедійного контенту (тексти, зображення, відео/аудіо), завдяки чому навчання проходить більш якісно і ефективно засвоюється зміст навчального предмета.

– принцип активності учнів здійснюється за рахунок можливості швидкого зворотного зв'язку вчителя/викладача і учня/студента, можливості контролю і підвищення мотивації учнів за допомогою нових методів навчання.

– принцип міцності засвоєння знань здійснюється за рахунок форми презентативного надання дидактичної інформації, проведення проміжних тестувань навіть за один урок, проведення підсумкових контрольних завдань і тестувань [1].

Більшість сучасних студентів технічно і психологічно готові до використання мобільних технологій в навчальному процесі, і необхідно розглядати нові можливості для більш ефективного використання потенціалу мобільних пристроїв і технологій. Вирішення цього завдання вимагає організаційних зусиль з боку педагогів, дослідної та методичної роботи викладачів щодо впровадження стратегій, форм і методів мобільного навчання в навчальний процес вищих навчальних закладів. Педагоги вже не можуть не звертати увагу на очевидний факт – популярність мобільних засобів зв'язку серед молоді; саме тому слід проаналізувати, як дані види зв'язку можуть сприяти оптимізації процесу навчання в вузі.

Існує ряд способів застосування мобільних пристроїв в навчальному процесі:

– для відтворення мультимедійних навчальних веб-ресурсів (аудіофайли, відеофайли, подкасти, графіка, карти, зображення);

– для забезпечення швидкого доступу на навчальні сайти, ресурси, довідники, словники;

– як власне навчальний засіб за умови розробки навчальних матеріалів, адаптованих для платформ мобільних засобів зв'язку (SMS-тести, навчальні посібники та інструкції на базі мобільних додатків);

– для навчальної комунікації (SMS-повідомлення, Twitter, вебінари, Skype та ін.).

Перспективним напрямком застосування мобільних технологій для підтримки традиційного навчального процесу є надання учням можливості доступу до навчальної інформації, що міститься в мережевих курсах, через освітній портал навчальної організації, адаптований для мобільних пристроїв. Зазначені ресурси можуть використовуватися студентами в режимі самостійної роботи для підготовки до участі в семінарах і диспутах, виконання групових проектів, завдань і курсових робіт, а також самотестування, проміжного і підсумкового тестування.

Крім того, студентам може надаватися можливість отримувати на свої мобільні пристрої повідомлення, нагадування, а також результати проходження контрольних заходів. В рамках підтримки очного навчального процесу мобільні пристрої можна використовувати також для проведення короткого тестування в кінці заняття, фіксування важливої інформації (використовуючи функцію аудіозаписи і фотокамери), оперативного доступу до довідкових та інформаційно-освітніх матеріалів під час занять. Мобільні пристрої незамінні для проведення практичних занять поза комп'ютерного класу.

Таким чином, впровадження мобільних технологій в навчальний процес дозволить:

– забезпечити безперервність і цілісність освітнього процесу за рахунок можливості безперервного доступу до довідкових матеріалів і освітніх ресурсів;

– підвищити якість викладання за рахунок оперативної реалізації зворотного зв'язку (наприклад, шляхом проведення короткого тестування в кінці заняття, що дозволить викладачам отримувати оперативну інформацію про засвоєння матеріалу студентами (яка частина навчального матеріалу засвоєна учнями краще, яким розділам слід приділити більше уваги і наскільки підтримується контакт з аудиторією);

– на основі отриманої інформації викладач може коригувати виклад матеріалу;

– забезпечити інформаційну та методичну підтримку практичних занять;

– надати більш якісні послуги в частині організації навчання:

– віддалений доступ до плану навчання, розсилка різних новин і повідомлень.

Одним з гарних прикладів використання подібних мобільних додатків є сервіс Google ClassRoom. Клас – це безкоштовний сервіс для навчальних закладів і некомерційних організацій. Також він доступний всім, у кого є осо-

бистий аккаунт Google. Завдяки Класу спілкування учнів і викладачів виходить на новий рівень – тепер воно не обмежується межами аудиторії.

Google Classroom пов'язує Google Drive, Google Docs, Sheets and Slides та Gmail разом, щоб допомогти навчальним закладам перейти до безпаперової системи. Пізніше було інтегровано Google Calendar, для того щоб допомогти з призначенням термінів виконання завдань, екскурсій та класних бесід. Студенти можуть бути запрошені в клас через базу інституту, за допомогою приватного коду, який потім може бути доданий в користувацький інтерфейс студента, або автоматично імпортуватися з шкільного домену. Кожен клас, створений за допомогою Google Classroom, створює окрему папку на Google диску викладача, куди студент може подати роботу для оцінки [2].

Звичайно, такий підхід є актуальним для країн і шкіл, де сучасні технології дуже активно використовуються в навчальному процесі. У нас це організувати вийде не скоро.

Головні особливості Google ClassRoom:

– налаштування класу – для кожного класу створюється свій код, який учні можуть використовувати для приєднання до спільноти. Цей процес усуває необхідність створення попередніх реєстрів;

– інтеграція з Google Drive – коли вчитель використовує Google Classroom, папка «Клас» автоматично створюється на його диску Google з новими вкладеннями для кожного створюваного класу;

– організація – коли учні використовують Google Classroom, папка «Клас» створюється на сторінці їх Google-диска з вкладеними папками для кожного класу, до якого вони приєднуються;

– автоматизація – при створенні завдання у вигляді Google-документа, платформа буде створювати і поширювати індивідуальні копії документа для кожного учня в класі;

– строки – при створенні завдання вчитель вказує термін виконання роботи. Коли учень надає завдання до початку терміну, на його документі з'являється статус «Перегляд», що дозволяє вчителям робити сортування;

– зв'язок – завдяки поєднанню класних оголошень, створених учителем, і інтегрованою можливістю коментування завдань, у викладачів і студентів завжди є можливість підтримувати зв'язок і бути в курсі статусу кожного завдання;

– робота/виправлення – коли учні приступили до своєї роботи, вчитель може забезпечити зворотний зв'язок в той момент, коли учень знаходиться в статусі «Перегляд» («Viewing»). Коли робота повертається учневі,

школяр знову перемикається в статус «Редакція» («Edit») і продовжує роботу над документом;

– зручний огляд – і вчителі, і студенти можуть бачити всі завдання на головному екрані Google Classroom. Це дозволяє контролювати роботу відразу в декількох класах.

При наявності підключення до Інтернету Клас можна відкрити на комп'ютері в будь-якому браузері, наприклад Chrome, Firefox, Internet Explorer або Safari. Клас підтримує всі нові версії основних браузерів одразу після їх виходу.

Отримати доступ до Класу також можна з мобільних пристроїв на базі Android і iOS.

Сервіс постійно поліпшується і допрацьовується, в процесі тестування було зібрано чимало відгуків і пропозицій від викладачів. Наприклад, спочатку вчителі не могли спілкуватися з учнями, поки вони не здадуть завдання – після цього була реалізована можливість комунікації вже в процесі виконання роботи. Вчителі також просили створити особливий розділ для публікації додаткових матеріалів до курсів.

В даний час Google Клас доступний на 42 мовах (російська також підтримується). Інтерфейс оптимізований для коректної роботи на мобільних пристроях, включаючи популярні електронні книги. Безумовно за таким форматом навчання майбутнє.

Застосування мобільних технологій для підтримки освітнього процесу може підвищити якість навчання і привабливість наданих освітніх послуг для студентів вузу. Навчання стає своєчасним, достатнім і персоналізованим. У зв'язку з вищевикладеним можна зробити висновок, що на даний момент дослідження, пов'язані з методиками і методами використання мобільних технологій, є своєчасними, перспективними і актуальними.

1.2 Переваги і недоліки використання мобільних додатків в освіті

Абсолютно ясно, що концепція сьогодення і майбутнього розвитку обміну інформацією, освіті буде будується на базі великих даних, мобільних пристроях, хмарних сервісах і соціальних технологіях, що відповідає потребам учнів, ефективному засвоєнню дидактичного матеріалу і контролю за якістю освіти.

Незаперечним є той факт, що мобільні технології впроваджуються і використовуються суспільством повсюдно, починаючи від повсякденного жит-

тя і включаючи всі сфери життєдіяльності суспільства: в побуті, в економіці, в бізнесі, в управлінні, в освіті, в суспільному житті. Діапазон застосування можливостей мобільних технологій постійно розширюється, стає доступним і легко використовуваним.

Мобільні технології дозволяють крім надання зв'язку незалежно від місцезнаходження користувача, планувати свою діяльність (наприклад, замовляти авіаквитки, бронювати готельні номери, робити покупки в on-line магазинах тощо), одержувати і поширювати інформацію, бути в курсі новин, бізнес -проектів, управляти особистим бізнесом, навчати і вчитися, виконувати роботу дистанційно, зберігати великий обсяг інформації, який може бути доступним скрізь і завжди [3].

Зростання мобільних технологій вносить постійні зміни і в освіту. Освіта більше не обмежена місцем або інструментами, такими як парти, комп'ютери або дорогі підручники. Замість цього вся сила навчання зосередилася у кожного на кінчиках пальців з миттєвим доступом і необмеженими можливостями.

При впровадженні будь-якої інновації потрібно бути готовим і до негативної сторони, тому розглянемо як безсумнівні переваги, так і негативні аспекти мобільного навчання. Таким чином, безперечними перевагами використання мобільних пристроїв і технологій в освітньому процесі є:

– мобільні пристрої можуть бути використані в будь-який час і в будь-якому місці: у вузі, вдома, в поїзді, в маршрутках, в готелях та ін. навчання стає воістину гнучким;

– студенти можуть підтримувати зв'язок з викладачем, а також взаємодіяти один з одним;

– викладачі та студенти можуть обмінюватися інформацією за допомогою електронної пошти, соціальних мереж (ВКонтакті, Facebook та ін.);

– можлива спільна робота за допомогою бездротової мережі або Bluetooth;

– облік індивідуальних особливостей студента – діагностика проблем, індивідуальний темп навчання;

– підвищення мотивації учнів за рахунок використання знайомих технічних засобів і віртуального оточення;

– створення персоналізованого професійно орієнтованого навчального простору студента;

– розвиток навичок і здібностей до безперервного навчання протягом життя;

– організація автономного навчання.

До негативних аспектів мобільного навчання в першу чергу необхідно віднести складнощі не стільки технічного і фінансового, скільки адміністративно-організаційного та методичного характеру. По-перше, складно переконати як викладачів, так і адміністрацію навчальних закладів, що дана форма навчання сприяє оптимізації навчального процесу, так як виконання завдань проходить на мобільних пристроях, які зазвичай заборонені у вузах для використання в навчальній аудиторії, оскільки всі мобільні пристрої виконують роль електронної шпаргалки. По-друге, викладачі не володіють (на відміну від учнів) відповідним рівнем ІКТ компетенції, який дозволяв би їм впроваджувати в традиційну форму завдання на основі мобільних технологій, використовувати вже існуючі навчальні програми для мобільних пристроїв, забезпечувати інтерактивну підтримку навчального процесу, розвивати ІКТ компетенцію самих учнів в цій сфері. По-третє, недостатньо готових навчальних мобільних ресурсів і програм для студентів різних рівнів та спеціальностей, але в той же час викладачі англійської мови знаходяться в більш виграшному положенні: існує велике різноманіття різних додатків та ігор на англійській мові, на основі яких можна створювати граматичні тести, пошукові та ігрові завдання. Технічно-фінансові проблеми зводяться до високої вартості деяких мобільних пристроїв (вже є недорогі альтернативи iPhone), маленькому екрану і невеликому шрифту, який використовується знову-таки не на всіх мобільних пристроях. Сучасний викладач сьогодні в силах перетворити мобільні пристрої і технології з загрози для навчання в допомогу і підтримку.

1.3 Аналіз подібних та існуючих систем підтримки навчального процесу

На даний момент існує величезне різноманіття мобільних додатків як освітнього так і не освітнього типів. До освітніх мобільних додатків можна віднести різноманітні вузькоспеціалізовані предметні програми, призначення яких заздалегідь зрозуміло. Для виконання даної магістерської роботи необхідно розглянути ряд додатків, які будуть націлені саме на підготовку до контрольних заходів в університетах. Виявити їх переваги та недоліки.

Варто відзначити, що додатків для підготовки до контролю знань досить багато. Але додатків, які націлені на роботу в певному університеті – мало. Додатків з такими ж цілями, як у додатку, що розробляється, які стосу-

валися б одеського державного екологічного університету або хоча б будь-якого іншого технічного університету України, немає зовсім. Єдиним порятунком для студентів можуть бути додатки іноземних університетів. Але необхідно мати хороший рівень принаймні англійської мови. При цьому зарубіжна література не завжди може допомогти нашому студенту в здачі іспитів, адже в кожному посібнику інформація трактується різним способом і переклад на рідну мову може бути вкрай несподіваним.

Існує кілька торгових майданчиків, на яких можна знайти мобільні додатки. Все залежить від операційної системи мобільного пристрою. Є три способи отримати додаток:

– Google Play;
– App Store;
– інсталяційні файли .apk на просторах інтернету.

Так як додаток розробляється під операційну систему Android, Доцільно буде використовувати офіційний магазин, так як в ньому всі програми повністю розроблені під операційну систему Android і мають власну ліцензію.

Пошук додатків здійснювався в Google Play. Google Play (колишня назва – Android Market) – магазин додатків, ігор, книг, музики і фільмів компанії Google і інших компаній, що дозволяє власникам пристроїв з операційною системою Android встановлювати і купувати різні додатки.

У Google Play можна знайти багато корисних і некорисних додатків. У магазині присутні платні і безкоштовні програми. На вересень 2017 в Google Play міститься більше 2,9 мільйона додатків для Android. Користувачі з більш 145 країн можуть купувати додатки, які мають до 122 локалізованих версій. Розробники понад 150 країн можуть передавати програми на Google Play, але не кожна з них підтримує реєстрацію продавця. Кількість скачувань додатків досягло 82 мільярдів.

У Google Play є 34 категорії (включаючи Ігри).

Також було відзначено, що російськомовних та україномовних програм практично немає, як і додатків з підготовки до контролю знань на факультеті комп'ютерних наук.

ABFM Exam Preparation (рис. 1.1) – мобільний додаток Американської Ради Сімейної Медицини (ABFM) надає сімейним лікарям корисну інформацію, щоб допомогти у підготовці до іспиту з сертифікації ABFM. Цей модуль перетворює Android-пристрій у безцінний інструмент для того, щоб підтримувати зв'язок з тим, що потрібно знати, коли ви починаєте свій рік сертифікації або повторної сертифікації.

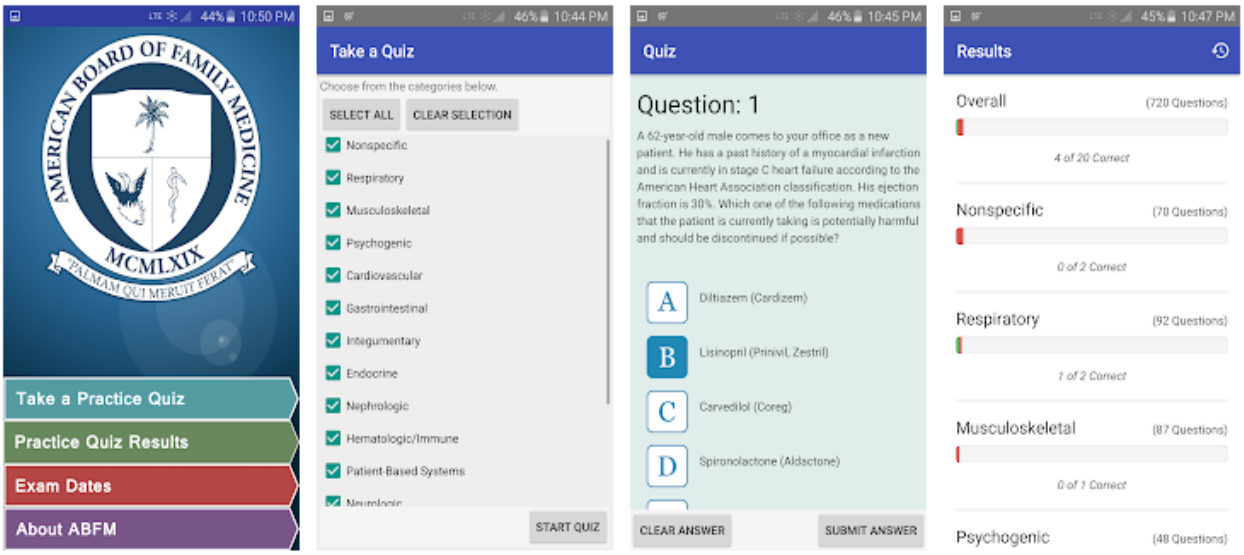

Рисунок 1.1 – Мобільний додаток – ABFM Exam Preparation

Американська Рада Сімейної Медицини (ABFM) є другою за величиною медичною спеціальністю у Сполучених Штатах. Заснована в 1969 році, вона є добровільною, неприбутковою, приватною організацією, метою якої є:

– покращення якості медичної допомоги, доступної для громадськості;

– встановлення та підтримка стандартів передового досвіду за фахом FamilyMedicine;

– підвищення стандартів медичної освіти для навчання в сімейній медицині;

– визначення за результатами оцінки відповідності фахівців сімейної медицини, які подають заяву та проводять сертифікати.

Особливості мобільного додатку:

– практика Вікторина – більше 400 питань, що вирішують проблеми, які часто зустрічаються сімейними лікарями. Кожна практична вікторина представляє 10 питань, за якими чітко пояснюється правильність відповіді, виключаючи відповідне посилання. Користувач може зосереджуватися на конкретній області для кожної вікторини та відстежувати загальні результати;

– майбутні дати іспитів – швидкий перегляд майбутніх іспитів ABFM;

– посилання на веб-сайт ABFM та веб-сайт журналу Американської Ради Сімейної Медицини.

Мобільний додаток добре продумано, має інтуїтивно зрозумілий, але не сучасний інтерфейс. Дозволяє вибирати теми, за якими необхідна підготовка. При неправильному виборі варіанту надає пояснення чому відповідь неправильна і яка була правильною.

При запуску вкладки з датами іспиту додаток повідомляє, що дана сторінка не доступна в Україні.

Для певних областей додатка (відео та терміни іспитів) потрібне підключення до Інтернету.

Незважаючи на те, що він не замінить добре розроблений та виконаний навчальний план, програма ABFM є додатковим інструментом для підготовки до іспиту з сертифікації ABFM. Цей модуль призначений для сімейних лікарів, але питання можуть бути корисними для медичних студентів та інших постачальників медичних послуг [4].

College Entrance Examination – найкраща онлайн-серія Prep Tests, заснована на Youth4Work.com, провідною онлайн-платформі, яка використовує передову аналітику даних, щоб допомогти студентам підготуватися до вступних іспитів до коледжу, поліпшити їх підготовку і прогрес на майбутнє (рис. 1.2).

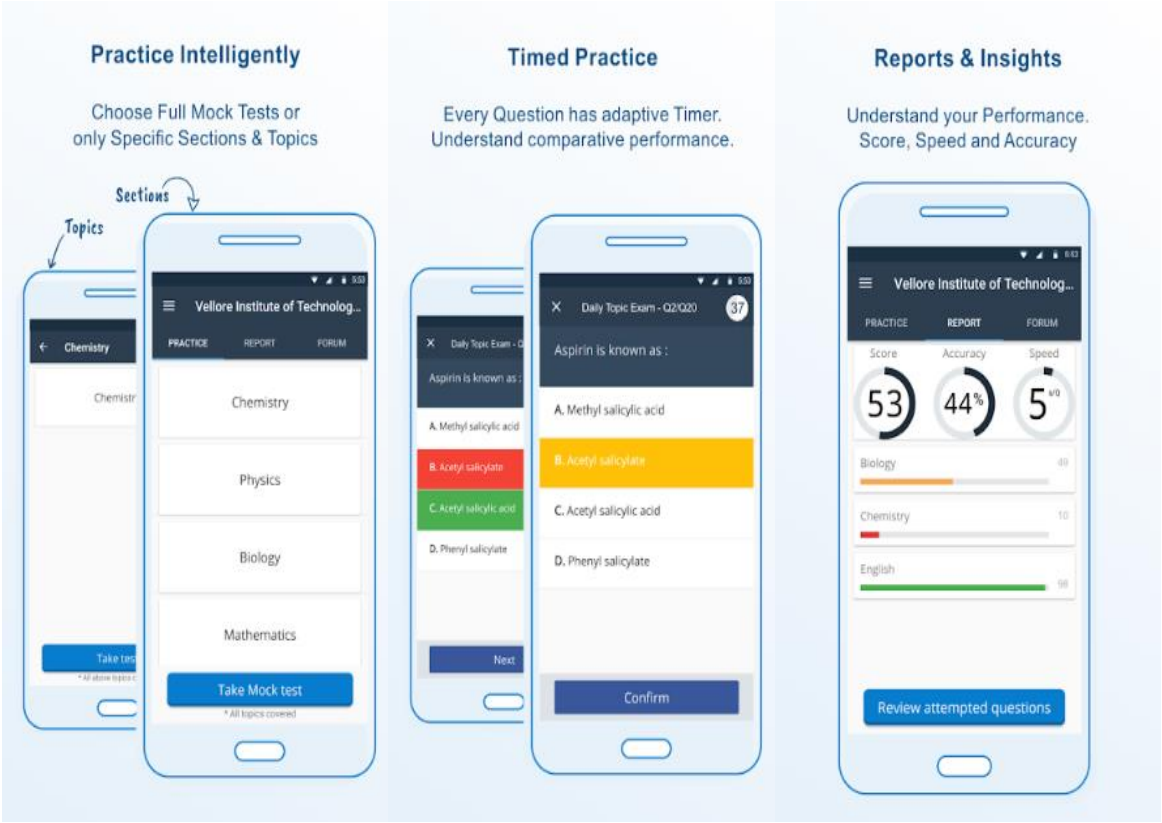

Рисунок 1.2 – Мобільний додаток – College Entrance Examination

Унікальні функції програми полягають в тому, що вона надає макетні тести, що охоплюють питання з усіх тем для вступних випробувань в коледжі, звіти, що відображають оцінку, точність і швидкість, що допомагає вам дізнатися, на якому рівні ваші знання, і відслідковувати свої успіхи, також має форуми для обговорення , щоб підтримувати зв'язок іншим студентами та бути добре інформованими про екзамени.

Цей мобільний додаток охоплює вступні іспити до 18 коледжів.

У список увійшли преміальні коледжі та університети, що пропонують відмінну освітню та навчальну середу для свого учня, і тисячі студентів прагнуть приєднатися до них щороку. Студенти також проводять заняття з коучингу і додаткові курси для цих вступних випробувань, за допомогою цього додатка можливість отримати всю необхідну інформацію та зання будуть мати ті студенти, які не можуть дозволити собі дороге навчання і навчання.

Прийом до коледжу є важливою віхою в житті студента, з цим додатком студент отримає допомогу в підготовці до того, щоб приручити вступні випробування коледжу з упевненістю і бути готовими до будь-яких питань.

Технічні характеристики:

– простий і інтуїтивно зрозумілий інтерфейс для відмінного навчання;
– адаптований для смартфонів і планшетів;
– займає менше місця на телефоні;
– добре оптимізовано, щоб споживати менше енергії телефону;
– чудовий час відгуку і без затримок в роботі мобільного додатка.

Творці впевнені, що додаток для вступних іспитів до коледжу допоможе студенту досягти успіху на будь-якому іспиті, і постійно вдосконалюють та випускають оновлення, щоб зробити цей додаток більш корисним в підготовці. Можна також використовувати форум для обговорення питань, пов'язаних з іспитом, і допомагати один одному досягати великих успіхів на іспитах [5].

Computer Science Solved MCQs – являє собою найсучасніші запитання на множинний вибір (MCQs) для студентів комп'ютерних наук. Ці MCQs включають базові матеріали для просування методологій в області комп'ютерних наук. Діапазони MCQs варіюються від Computer Fundamental до систем баз даних, від комп'ютерного обладнання до операційних систем, від мережі до штучного інтелекту і багатьох інших. Додаток найкраще підходить для студентів і учнів, які цікавляться інформатикою. Програма має MCQs в простому форматі розкритої карти, щоб кожен міг легко вибирати необхідну те-

му. Додаток включає в себе: найкращі MCQs для комп'ютерної техніки, MCQs для NTS, MCQs збірники NTS, MCQs для курсу обчислювальної техніки, останні MCQs для комп'ютерних наук, MCQs для науки, MCQs для комп'ютерного програмування (рис. 1.3)

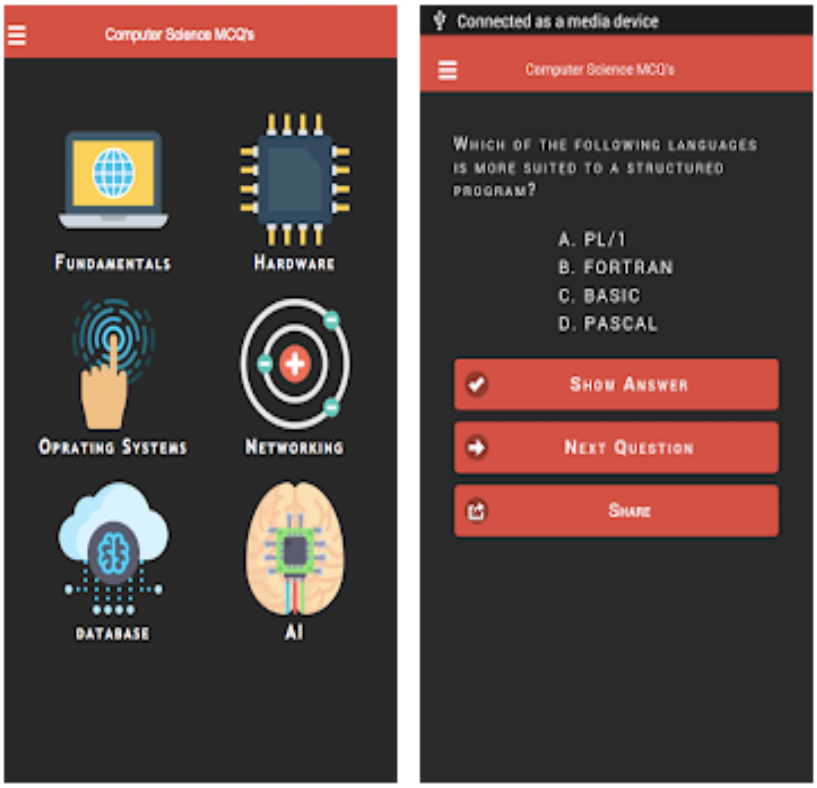

Рисунок 1.3 – Мобільний додаток – Computer Science Solved MCQs

Комп'ютерні науки MCQs є дуже важливим компонентом вступних іспитів для інженерних коледжів, тому в цьому додатку також розглянуті матеріали минулих видань ETEA для підготовки студентів до розробки для вступних іспитів в коледжі і університети державного сектора.

Базовий модуль додатку включає в себе загальні питання комп'ютерних наук, а також деякі загальні поради та рекомендації MS Office, які в основному охоплюють основи MS Word, основи MS Excel, основи MS PowerPoint, основи MS Access.

Модуль Advance включає питання, які тягнуть за собою основні концепції комп'ютерних мов, з яких ви можете вивчити основи HTML, CSS, програмування на PHP, Java, програмування ASP.NET, програмування на Python, C#, на C++, вивчати основи програмування ruby, на android і IOS. Також студент може вивчити основи мови програмування Swift, Javascript, Elixir, Scala і основи програмування мови R.

Унікальний алгоритм додатку налаштовує рівень складності питань з множинним вибором в залежності від прогресу, здатності і продуктивності користувачів. Тому, якщо студент добре розбираєтеся в основах, то є можливість вибору складності MCQs, які будуть перевіряти здібності на самих популярних платформах програмування, наприклад AWS, React, PostgreSQL, DevOps, Machine Learning, Node.js, Rails, Data Science, Linux, AngularJS, Django, Redux, AI, доповнена реальність, еластичний пошук, технології Blockchain тощо [6].

При вивченні роботи додатків були зроблені висновки, що кожен з додатків зручний у використанні, має велику кількість тем і питань та націлений на свою аудиторію. Як для студента українського університету, можна виділити основними недоліками те, що немає підтримки української мови. Немає тестів на теми певного університету, в якому навчається студент. У наш час необхідно впроваджувати сучасні технології в освіту, з метою підтримки інтересу до навчання в студентах поточного покоління.

2 ПРОГРАМНІ ЗАСОБИ ДЛЯ РОЗРОБКИ ANDROID-ДОДАТКУ

2.1 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. 2.1);
– доступ до розробки будь-якому користувачеві;
– абсолютно безкоштовна для розробки.

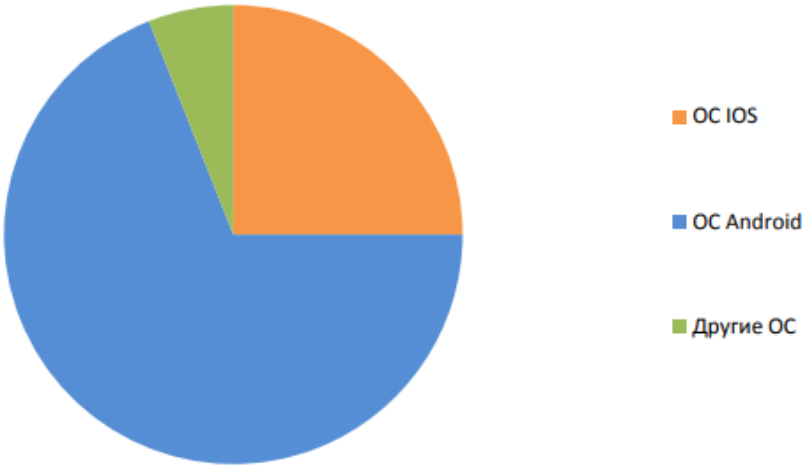

Рисунок 2.1 – Частка пристроїв на різних ОС

Android – операційна система для смартфонів, планшетних комп'ютерів, електронних книг, цифрових програвачів, "розумних" наручних годинників, ігрових приставок, нетбуків, смартбуків, окулярів Google, телевізорів, систем автоматичного керування автомобілем та інших пристроїв. ОС заснована на ядрі Linux і власної реалізації віртуальної машини Java від Google. Спочатку розроблялася компанією AndroidInc., Яку в 2005 році купила Google. Згодом Google ініціювала створення альянсу OpenHandsetAlliance (OHA), який зараз займається підтримкою і подальшим розвитком платформи. Android дозволяє створювати Java-додатка, що керують пристроєм через розроблені Google бібліотеки. AndroidNativeDevelopmentKit дозволяє перенести (але не налагоджувати) бібліотеки і компоненти додатків, написані на Сі та інших мовах. ОС Android встановлена на 86% смартфонів (2014 року) [7].

Android є найпоширенішою операційною системою (ОС) для мобільних пристроїв – телефонів і планшетів. Дана система має безліч характерних рис, які роблять її популярною і привабливою для великої кількості користувачів по всьому світу.

Операційна система Android є невимогливою і здатна працювати на різних конфігураціях. Саме тому більшість світових виробників оснащують свої пристрої даною ОС, оскільки інші програмні продукти призначені для окремих апаратів, що відповідають певній специфікації. Така гнучкість Android пов'язана з тим, що система побудована на ядрі Linux, що має відкритий програмний код, що дає необмежені можливості розробникам. Android може бути запущений на пристроях, що мають об'єм оперативної пам'яті менше 256 Мб. Найбільш нові версії системи вимагають 512 Мб оперативної пам'яті, що також є невеликим значенням для сучасних апаратів.

Система не вимагає наявності високопродуктивного процесора і може працювати на пристроях, оснащених ядром з частотою 600 МГц. Операційна система дає можливість установки додатків з офіційного репозиторію Google, який надає найбільшу в світі базу програм. Це пов'язано з тим, що кожен розробник може самостійно написати будь-яку програму для апарату і розмістити її в магазині. Можливість також реалізована завдяки відкритості операційної системи. Варто відзначити, що додатки на пристрої під управлінням Android можуть бути встановлені як безпосередньо з телефону або планшета, так і через комп'ютер шляхом завантаження файлу .apk і його подальшої установки на апараті [8-9].

Відмінною особливістю Android є його інтегрованість з сервісами Google – Gmail, Hangouts, Voice Search тошо.

На Android офіційно реалізована підтримка Chrome, що дозволяє синхронізувати вкладки, що відкриваються в браузері на смартфоні з комп'ютерним браузером.

Наприклад, ви можете почати перегляд сторінок з вашого телефону і при бажанні продовжити вивчати інформацію, відкривши цю ж вкладку на комп'ютері, не вдаючись до допомоги повторного пошуку.

«Андроїд» має досить простий і інтуїтивно зрозумілий інтерфейс. Всі потрібні програми розміщуються одночасно на головному екрані і в меню апарату, яке викликається натисканням на центральну сенсорну клавішу або відповідну кнопку на екрані. Всі настройки розташовуються в секції «Налаштування», а кожна дія користувача пояснюється коментарями і підказками при першому запуску апарату.

Операційна система швидко реагує на натискання користувача і виробляє установку і скачування потрібних програм і файлів зі швидкістю, яка не програє іншим сучасним мобільним ОС.

2.2 Загальна схема роботи програми Android

Додатки для Android в своїй роботі використовує вікна (аналогічно Windows), проте в даній системі вищевказані вікна носять іншу назву – Activity. Як і в Windows, кожне вікно має свій життєвий цикл і свої особливості (рис. 2.2).

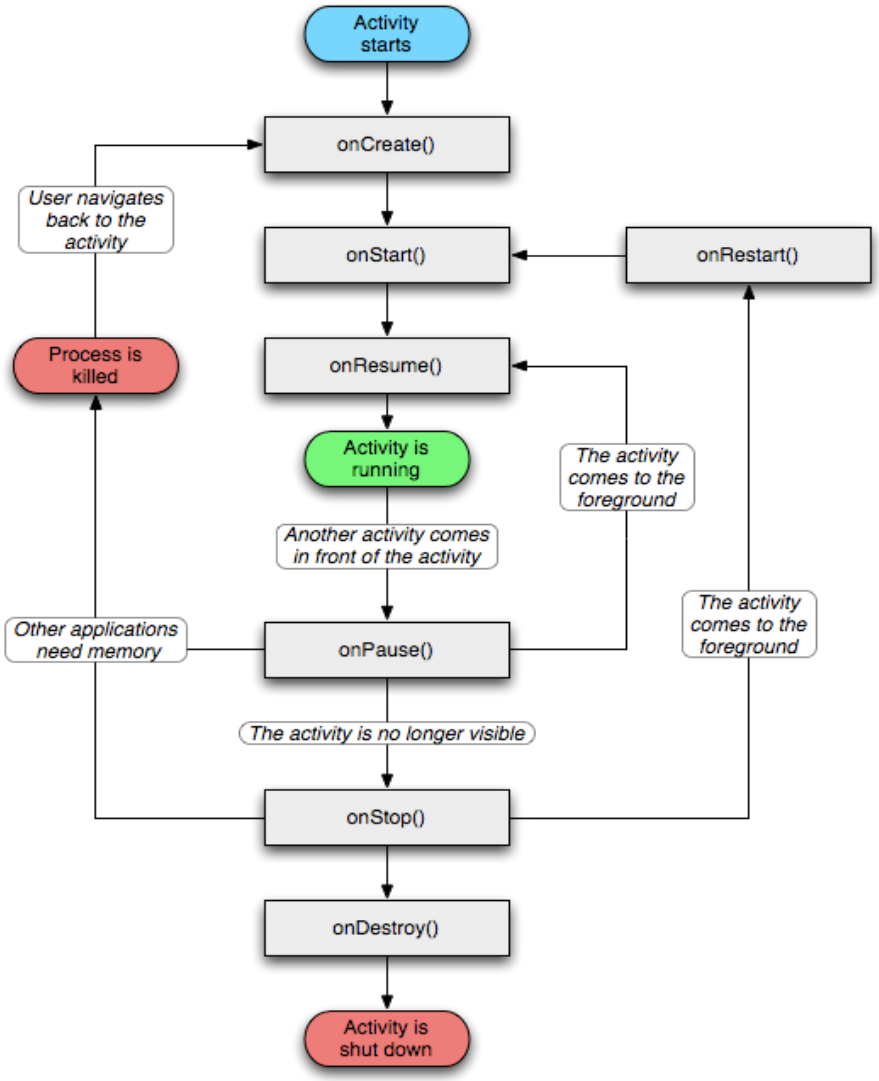

Рисунок 2.2 – Життєвий цикл додатки для системи під управлінням Android

При створенні нового вікна викликається метод onCreate(), при розробці даний метод перевизначається і в ньому відбувається ініціалізація програ-

ми та його компонентів. Далі викликаються методи onStart() і onResume(). Обидва методи викликаються перед відображенням вікна при його створенні, або відновленні (при перемиканні з іншої програми, при розгортанні згорнутого додатки і тп). При згортанні викликаються методи onPause() і onStop(). При закритті програми і вікна викликається onDestory(), в даному методі можна зберегти призначені для користувача дані і параметри. Повний опис та послідовність виклику методів можна знайти на офіційному сайті [10].

2.3 Аналіз сучасних засобів розробки мобільних додатків

Інструменти, за допомогою яких створюються додатки на Андроїд, можна умовно поділити на дві великі групи:

– програми, які встановлюються на ПК;
– «хмарні» сервіси з закритим доступом.

Основна відмінність між двома групами полягає в тому, що для роботи в програмах на ПК важливо розуміти середовище розробки утиліт, знати, як підключати бібліотеки, розбиратися в структурі фреймворка і так далі. «Хмарні сервіси», в свою чергу, надають не тільки каркас додатка, а й набір шаблонів, які можна з'єднувати подібно конструктору, а також, якщо є така, можливість, додавати в код утиліти власні команди.

При створенні мобільного додатка важливо не обмежуватися рамками середовища, тому далі будуть розглядатися програмне забезпечення, що встановлюється на ПК, як більш широке за можливостями.

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) [11-13].

Вище 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. 2.1).

Таблиця 2.1 – Вибір засобу розробки мобільного додатку

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
Супровід і розширення функціональності (рівнем нижче ніж у розробників системи)	Низький поріг входження	Низький поріг входження	Високий поріг входження	Середній поріг входження
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. 2.2).

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 2.2 – Таблиця балів за критеріями

Критерии	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
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.

2.4 Вибір засобів реалізації мобільної системи підтримки навчального процесу

Для створення програми на ОС Android, можуть бути обрані різні середовища розробки, такі як, Eclipse, Embarcadero JBuilder, JDeveloper тощо, але для роботи обрана AndroidStudio від компанії Google. Причиною вибору став простий спосіб підключення, а так само зручність інтерфейсу і швидкодії програми.

Android Studio – це інтегроване середовище розробки (IDE) для роботи з платформою Android, анонсована 16 травня 2013 року на конференції Google I/O. 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.

AndroidStudio, заснована на програмному забезпеченні IntelliJ IDEA від компанії JetBrains, офіційний засіб розробки Android додатків. Дане середовище розробки доступне для Windows, OS X і Linux.

Технологія Java включає в себе мову програмування Java, засоби трансляції вихідного тексту програми – вихідного коду – в спеціальну форму, придатну для виконання комп'ютером, і засоби виконання Java-програм на різних платформах, тобто в різних операційних системах і на різному апаратному забезпеченні. Основна особливість JAVA-технології в тому, що перетворена на етапі трансляції в спеціальний код Java-програма повністю незалежна від машини [14].

Якщо виконуваний код, отриманий з програм на інших розповсюджених мовах, зазвичай не придатний для виконання комп'ютером "іншої платформи", то до виконуваного коду Java таке обмеження не відноситься. Але необхідно, щоб для "цільової платформи" була реалізація так званої Java-машини – середовища виконання JAVA-програм.

JAVA-технології, що активно просувають компанія SUN, набули широкого поширення (далеко не тільки в web-рішеннях). А незалежність від платформи Java, що дозволяє інтегрувати засоби виконання Java-програм в браузери, що працюють в самих різних операційних системах, визначила поширення Java в якості елемента Web-технологій.

JAVA використовується для створення складних інтерактивних елементів, пов'язаних з web-сайтом. Наприклад, на Java реалізуються складні інструменти для роботи з базами даних, розміщеними в Web. Або графічні інтерфейси, що вимагають виведення складних інтерактивних елементів. І,

звичайно, багато іншого, від мережевих шахових програм до засобів редагування звукових файлів.

Для зберігання даних використовують різні СКБД, такі як, SQLite, MySQL, Oracle, але для роботи обрана NoSQL.

Кожна система підтримує різні моделі і структури баз даних. Ця модель і визначає, як створювана СКБД буде оперувати даними. Існує досить небагато моделей БД, які надають способи чіткого структурування даних, найпопулярніша з таких моделей – реляційна модель.

Реляційна модель і реляційні БД можуть бути дуже потужним інструментом, але тільки якщо програміст знає як з ними поводитися. Нещодавно, стали набирати популярність NoSQL системи з обіцянкою позбутися старих проблем БД і додати новий функціонал. Виключаючи жорстку структуру даних, при цьому зберігши реляційний стиль, ці СКБД пропонують більш вільний спосіб роботи з ними і набагато більші можливості для їх налаштування. Хоча не обходиться і без виникнення нових проблем.

Бази даних NoSQL спеціально створені для певних моделей даних і мають гнучкі схеми, що дозволяє розробляти сучасні програми. Бази даних NoSQL набули широкого поширення в зв'язку з простотою розробки, функціональністю і продуктивністю при будь-яких масштабах. У них застосовуються різні моделі даних, в тому числі документні, графові, пошукові, з використанням пар «ключ-значення» і зберіганням даних в пам'яті.

Протягом десятиліть центральне місце в розробці додатків займала реляційна модель даних, яка використовувалася в реляційних базах даних, таких як Oracle, DB2, SQL Server, MySQL і PostgreSQL. Але в середині – наприкінці 2000-х років помітне поширення стали отримувати і інші моделі даних. Для позначення класів БД і моделей даних був введений термін «NoSQL». Часто «NoSQL» використовується як синонім до терміну «нереляційний».

NoSQL прибирає всі обмеження реляційної моделі (недостатня продуктивність, трудомістке горизонтальне масштабування, недостатня продуктивність в кластері) і полегшує кошти зберігання і доступу до даних. Такі БД використовують неструктурований підхід (створення структури на льоту), тим самим знімаючи обмеження жорстких зв'язків і пропонуючи різні типи доступу до специфічних даними.

Для подання загальної картини необхідно порівняти звичну SQL та NoSQL:

– структури даних і їх типи – реляційні БД використовують строгі схеми даних, NoSQL БД допускають будь-який тип даних;

– запити – незалежно від типу ліцензії, реляційні бази даних в тій чи іншій мірі відповідають стандартам SQL, тому дані з них можна отримувати за допомогою мови SQL. NoSQL БД використовують специфічні способи запитів до даних;

– масштабованість – обидва ці типу СКБД досить легко піддаються вертикальному масштабуванню (тобто збільшення системних ресурсів). Проте, так як NoSQL це більш сучасний продукт, саме такі СКБД пропонують більш прості способи горизонтального масштабування (тобто створення кластера з декількох машин);

– надійність – коли справа доходить до збереження даних і гарантії виконання транзакцій SQL БД як і раніше займають лідируючі позиції;

– підтримка – реляційні СКБД мають не малу історію за плечима. Вони дуже популярні і пропонують як платні, так і безкоштовні рішення. При виникненні проблем, все ж набагато простіше знайти відповідь, якщо справа стосується реляційних систем, ніж NoSQL, особливо якщо рішення досить складне за своєю природою (наприклад MongoDB);

– зберігання та доступ до складних структур даних – спочатку реляційні системи припускали роботу зі складними структурами, саме тому вони перевершують інші рішення по продуктивності [15].

3 ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Середа 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 – 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 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 форму-

ється 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.

3.2 Порівняння реляційних та нереляційних БД

Спосіб використання веб-додатків з даними істотно змінився за останнє десятиліття. З'явилася необхідність роботи з великими обсягами даних, і більшій кількості користувачів потрібен доступ до цих даних одночасно. Це означає, що масштабованість і продуктивність стали більш складними для реляційних баз даних, заснованих на схемі. Ця проблема привела до появи ряду систем управління базами даних NoSQL («Not only SQL», що підкреслює, що використання SQL-стилю запитів не є вирішальним для цілей таких сховищ даних), які націлені на користь збільшення продуктивності і підтримки горизонтального масштабування.

Однією з важливих відмінностей між базами даних NoSQL і реляційними базами даних є те, що NoSQL – це форма неструктурованого зберігання. Це означає, що бази даних NoSQL не мають фіксованої структури таблиць, такий як ті, які знаходяться в реляційних базах даних (рис. 3.1).

Структури даних, що використовуються базами даних NoSQL (наприклад, ключ-значення, широкий стовпець, граф або документ), відрізняються від тих, які використовуються за замовчуванням в реляційних базах даних, що робить деякі операції швидше в NoSQL. Деякі моделі баз даних NoSQL

також дозволяють розробникам зберігати в базі даних серіалізовані об'єкти, а не тільки прості строкові значення.

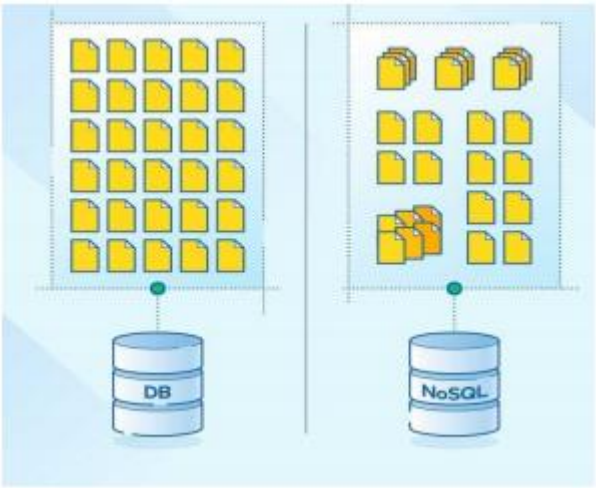

Рисунок 3.1 – Зберігання даних в реляційній і NoSQL базах даних

Серіалізация представляє процес запису стану об'єкта в потік, відповідно процес вилучення або відновлення стану об'єкта з потоку називається десеріалізацію. Серіалізация дуже зручна, коли йде робота зі складними об'єктами. Серіалізация використовується для передачі об'єктів по мережі і для збереження їх у файли. Наприклад, потрібно створити розподілене додаток, різні частини якого повинні обмінюватися даними зі складною структурою. У такому випадку для типів даних, які передбачається передавати, пишеться код, який здійснює сериализацию і десеріалізацію. Об'єкт заповнюється потрібними даними, потім викликається код сериализации, в результаті виходить, наприклад, XML-документ. Результат сериализации передається приймаючій стороні за, скажімо, електронній пошті або HTTP. Додаток-одержувач створює об'єкт того ж типу і викликає код десеріалізациі, в результаті отримуючи об'єкт з тими ж даними, що були в об'єкті додатки-відправника [16].

При роботі з базами даних NoSQL масштабування простіше і дешевше, ніж при роботі з реляційними базами даних. Це відбувається через горизонтального масштабування і розподілу навантаження на всіх вузлах, а не звичайною вертикалі, яка виконується з реляційними системами баз даних.

Звичайно, бази даних NoSQL не ідеальні і не завжди можуть бути правильним вибором. По-перше, більшість баз даних NoSQL не підтримують функції надійності, які підтримуються реляційними базами даних (вимоги ACID). Щоб підтримувати надійність і узгодженість, розробники повинні ре-

алізовувати свій власний код, що робить систему більш складною. Це може обмежити кількість додатків, які можуть покладатися на бази даних NoSQL, таких як банківські системи або управління персональними даними. Інша проблема, виявлена в більшості баз даних NoSQL, – це несумісність з SQL-запитами. Це означає, що необхідний ручний або пропрієтарний мову запитів, що додає більше часу і складності.

Необхідно побудувати таблицю порівняння баз даних NoSQL і реляційних баз даних (табл. 3.1).

Таблиця 3.1 – Порівняння реляційних і NoSQL баз даних

Характеристика	NoSQL	Реляційна БД
Ефективність	Висока	Низька
Надійність	Погана	Хороша
Доступність	Хороша	Хороша
Узгодженість	Погана	Хороша
Сховище даних	Оптимізовано для великого об’єму даних	Від середніх до великих розмірів
Масштабованість	Горизонтальна	Вертикальна
Типи	Безліч різних типів, включаючи сховища «ключ-значення», бази даних документів, сховища сімейств колонок і графові бази даних	Один тип (база даних SQL) з незначними варіаціями
Приклади БД	MongoDB, Cassandra, HBase, Neo4j	MySQL, Postgres, Microsoft SQL Server, Oracle Database

Існують різні моделі, що визначають логічну структуру бази даних. Кожна модель даних має свою область застосування, але найбільш популярною в даний час є реляційна модель даних, що забезпечує цілісність даних, проте має проблеми з масштабуванням і гнучкістю БД [17].

З появою NoSQL виникли інші способи представлення даних. Існують наступні види СКБД NoSQL:

– сховище виду «ключ – значення»;

– документоорієнтовані;
– графові;
– сховище сімейств колонок.

У сховищах виду «ключ – значення» використовується асоціативний масив (також відомий як карта або словник) в якості основної моделі даних. У цій моделі дані представляються у вигляді набору пар ключ-значення, так що кожен можливий ключ з'являється не більше ніж один раз в колекції. Такі сховища спроектовані з прицілом на масштабованість, немає схем побудови баз даних, немає зв'язків між значеннями. Однак єдині ефективні запити – це прості, важко використовувати такі операції SQL як JOIN і GROUP BY. Також потрібно вдосконалити код, щоб зібрати колекції пар ключ-значення в об'єкти.

Приклади таких сховищ – Berkeley DB, MemcacheDB, Redis, Riak.

Документоорієнтовані БД являють собою систему зберігання ієрархічних структур даних (документів). Сховища документів схожі на сховища ключових значень, оскільки вони не мають схеми і засновані на моделі з ключем. Однією з інших визначальних характеристик документоорієнтованих бази даних є те, що база даних також пропонує мову запитів, який витягує документи на основі їх вмісту (наприклад, вибірку за запитом без повного завантаження окремих документів в оперативну пам'ять; пошук частини документів; організація окремих колекцій документів одного типу).

Документоорієнтовані бази даних є вдосконаленою версією сховищ «ключ-значення», тільки вони все ще не дуже гарні для систем з множинними зв'язками між елементами, але знайдуть застосування, але знайдуть застосування, де важко впорядкувати дані за таблицями і сутностей, і потрібні швидкі транзакції.

Приклади СКБД даного типу – CouchDB, Couchbase, MarkLogic, MongoDB, eXist, Berkeley DB XML.

Графові бази даних є узагальненням мережевий моделі даних. Вони зазвичай використовуються в додатках для соціальних мереж і дозволяють розробникам більше зосередитися на відносинах між об'єктами, а не на самих об'єктах. У цьому контексті вони дійсно дозволяють використовувати масштабовану і зручну у використанні середовище.

Приклади: Neo4j, OrientDB, AllegroGraph, Blazegraph, InfiniteGraph.

У сховищі сімейств колонок дані зберігаються в стовпцях, на відміну від систем управління реляційні базами даних, де дані зберігаються в рядках. Дані впорядковані у вигляді розрідженій матриці, рядки і стовпці якої вико-

ристовуються в якості ключів. Будь-яка кількість стовпців (і, отже, багато різних типів даних) може бути згруповано або агрегований в міру необхідності для запитів або представлень даних.

Прикладами СКБД даного типу є: Apache HBase, Apache Accumulo, Hypertable, SimpleDB.

На відміну від реляційної моделі, яка зберігає логічну бізнес-сутність додатки в різні фізичні таблиці з метою нормалізації, NoSQL сховища оперують з цими сутностями як з цілісними об'єктами (рис. 3.2).

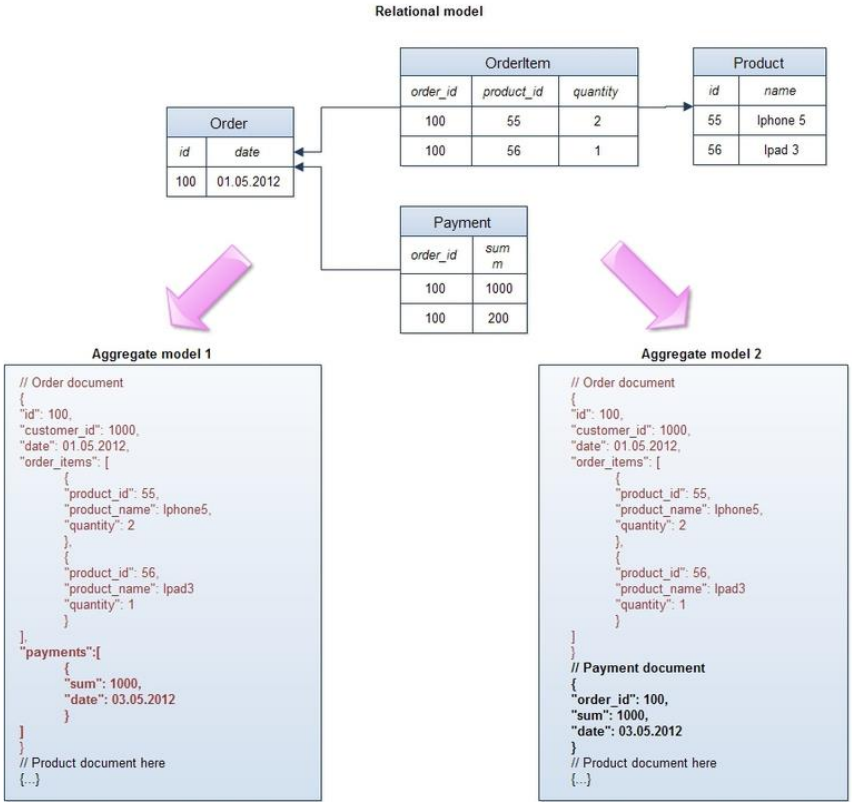

Рисунок 3.2 – Вигляд реляційної та нереляційної моделі

У цьому прикладі продемонстровано агрегати для стандартної концептуальної реляційної моделі e-commerce "замовлення – позиції замовлення – платежі – продукт". В обох випадках замовлення об'єднується з позиціями в один логічний об'єкт, при цьому кожна позиція зберігає в собі посилання на продукт і деякі його атрибути, наприклад, назва (така денормализация необхідна, щоб не запитувати об'єкт продукту при добуванні замовлення – головне правило розподілених систем – мінімум "Джоін" між об'єктами). В одному агрегаті платежі об'єднані із замовленням і є складовою частиною об'єкта, в іншому – винесені в окремий об'єкт. Цим демонструється головне правило проектування структури даних в NoSQL базах – вона повинна підкорятися

вимогам програми та бути максимально оптимізована під найбільш часті запити. Якщо платежі регулярно витягуються разом із замовленням – має сенс їх включати в загальний об'єкт, якщо ж багато запити працюють тільки з платежами – значить, краще їх винести в окрему сутність.

3.3 Система керування базами даних Firebase

Додаток працює з базою даних Firebase, яка знаходиться на сервері. Основний сервіс – хмарна СКБД класу NoSQL, що дозволяє розробникам додатків зберігати і синхронізувати дані між декількома клієнтами. Firebase служить базою даних, яка змінюється в реальному часі і зберігає дані в JSON. Будь-які зміни в базі даних відразу синхронізуються між усіма клієнтами, або девайсами, які використовують одну і ту ж базу даних. Іншими словами, оновлення в Firebase відбуваються миттєво.

Разом зі сховищем, Firebase також надає призначену для користувача аутентифікацію, і тому всі дані передаються через захищене з'єднання SSL. Ми можемо вибрати будь-яку комбінацію email і пароля для аутентифікації, будь то Facebook, Twitter, GitHub, Google, або щось інше.

Головні переваги Firebase:

– швидкість роботи – пакеті розробника Firebase зібрані інтуїтивно зрозумілі API, які спрощують і прискорюють розробку якісних додатків. Також у розпорядженні розробника всі необхідні інструменти для розширення бази користувача і підвищення прибутку – залишається тільки вибрати відповідні для цілей;

– готова інфраструктура – розробнику не доведеться створювати складну інфраструктуру або працювати з декількома панелями управління. Замість цього він зможете зосередитися на потребах користувачів;

– статистика – в основі Firebase лежить безкоштовний аналітичний інструмент, розроблений спеціально для мобільних пристроїв. Google Analytics для Firebase дозволяє отримувати дані про дії ваших користувачів і відразу ж вживати заходів за допомогою додаткових функцій;

– кросплатформеність – Firebase працює на будь-яких платформах завдяки пакетам розробника для Android, iOS, JavaScript і C++. Розробник також можете звертатися до Firebase, використовуючи серверні бібліотеки або REST API;

– масштабованість – якщо додаток стане популярним і навантаження на нього зросте, розробнику не доведеться змінювати код сервера або залуча-

ти додаткові ресурси – Firebase зробить це за нього. Крім того, більшість функцій Firebase безкоштовні і залишаться такими незалежно від масштабу проектів. Платних функцій чотири. У них передбачено безкоштовний пробний період і два тарифних плани;

– безкоштовна підтримка по електронній пошті. Крім того, команда Firebase і фахівці з розробки Google дадуть відповіді на ваші питання на ресурсах Stack Overflow і GitHub.

В розпорядженні розробника знаходиться масштабний функціонал (рис. 3.3)

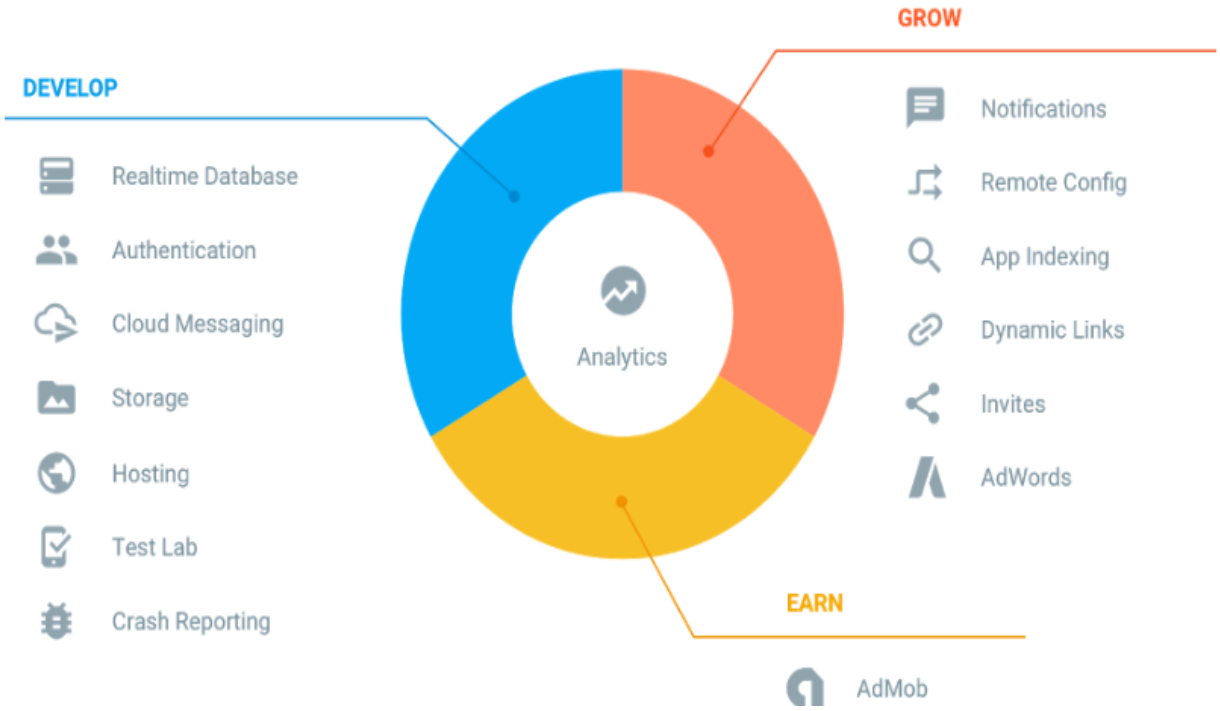

Рисунок 3.3 – Основний функціонал та можливості Firebase

– Analytics – аналітика за додатком: розмір аудиторії, інформація про користувачів, події в додатку та інше;

– Authentication – користувачі можуть прив'язати свої облікові записи до додатка, а до них ми можемо прив'язати будь-які дані. З коробки підтримуються наступні провайдери авторизації: Google, Facebook, Twitter, GitHub, анонімний вхід і імейл-пароль для своєї реєстрації. Бракує тільки VK-авторизації;

– Realtime Database – база даних, працює з живими змінами в реальному часі;

– Storage – сховище для файлів користувачів, можна легко зробити персональне сховище, а можна і ділитися файлами;

– Hosting – моментальне розгортання веб-додатків і мобільних додатків за допомогою безпечної глобальної мережі доставки контенту;

– Test Lab for Android – можливість тестувати додатки Android на самих різних пристроях;

– App Indexing – можливість зв’язати інформацію з веб-сайту з внутрішніми сторінками додатка, також є можливість індексувати дані додатки і відображати їх в результатах пошуку на пристрої;

– Crash Reporting – збір інформації про збої в додатку (на ранніх версіях і сам був джерелом краш, але начебто полагодили);

– Notifications – повідомлення, заміна старих Google Cloud Messaging;

– Remote Config – спосіб змінювати поведінку програми прямо зі свого сервера, змінюючи потрібні параметри;

– Dynamic Links – корисний спосіб прокинути контекст в додаток (наприклад, користувач читав про аспірин на сайті, перейшов в маркет, встановив додаток, і йому відкрилася сторінка з аспірином);

– AdMob – рекламний сервіс з безліччю форматів, по праву займає лідируючі позиції в мобільній рекламі. У цій мережі реклами завжди багато, і вона модерується.

3.4 Проектування БД

NoSQL – спосіб структуризації даних полягає в позбавленні від обмежень при зберіганні і використанні інформації. Бази даних NoSQL, використовуючи неструктурізірованний підхід, пропонують багато ефективних способів обробки даних в окремих випадках (наприклад, при роботі зі сховищем текстових документів).

NoSQL – СКБД не використовують реляційну модель структуризації даних. Існує багато реалізацій, вирішуючих це питання по-своєму, часто дуже специфічне. Ці рішення допускає необмежене формування записів і зберігання даних у вигляді ключ-значення.

На відміну від традиційних РСКБД, деякі бази даних NoSQL, наприклад, MongoDB, дозволяють групувати колекції даних з іншими базами даних. Такі СУБД зберігають дані як одне ціле. Ці дані можуть являти собою одиночний об'єкт на зразок JSON і разом з тим коректно відповідати на запити до полів. NoSQL бази даних не використовують загальний формат запиту (як SQL в реляційних базах даних). Кожне рішення використовує власну систему запитів.

Нижче представлені суті користувач і месенджер бази даних мобільного додатка (рис. 3.4, 3.5). З першого погляду вони можуть здатися незрозумілими. Нереляційних база даних являє собою набір, що складається з безлічі полів виду ключ-значення. Ключі генеруються автоматично за допомогою різних методів шифрування на стороні Firebase. Метод шифрування вибирається розробником на етапі формування бази даних. Кожному користувачеві відповідає його унікальний ключ, до якого наводиться в зіставлення значення досягнень користувача, розбитих по кожній темі. Так і сутності кожної кімнати месенджера є свої ключі і значення. Ключі відповідають за ідентифікацію кожного нового повідомлення, значення відображають час і автора повідомлення і безпосередньо саме повідомлення.

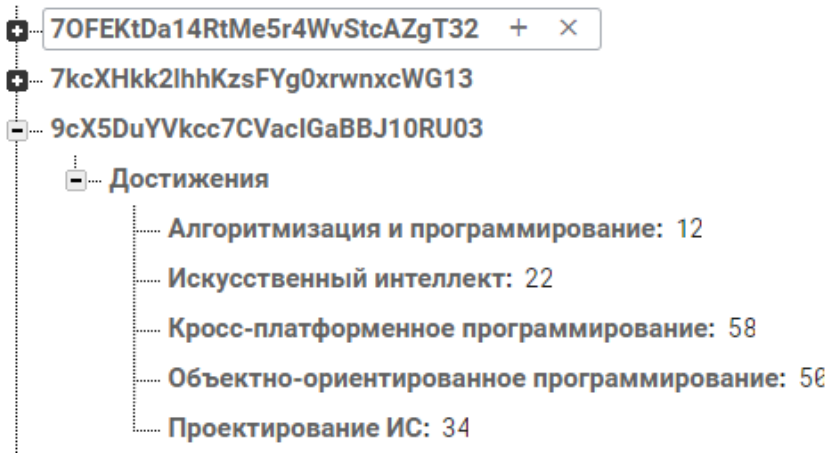

Рисунок 3.4 – Представлення сутності Користувач

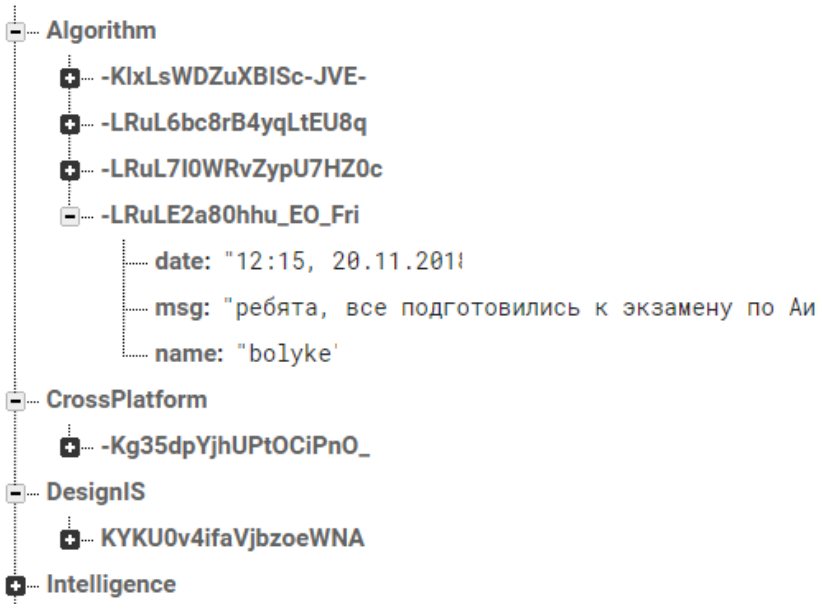

Рисунок 3.5 – Представлення сутності Месенджер

Використання NoSQL в даному випадку абсолютно виправдано, так як база даних буде постійно зростати, будуть додаватися нові користувачі і канали спілкування, повідомлення в чатах можуть зростати до великих обсягів і використовувати базу даних з вертикально масштабованістю не буде добрим вибором. Ефективність нереляційних бази даних дозволить здійснювати швидке отримання будь-якої необхідної інформації.

3.5 Архітектура системи підтримки навчального процесу

Концепція розробляється програмного продукту являє собою додаток для ОС Android, що надає такі можливості:

– доступ до особистого кабінету;
– налаштування профілю;
– дисципліни та їх теми для підготовки до екзамену, заліку або модулю;
– вибору режиму та умов для проходження тестування;
– створення власного словника з термінами та визначеннями;
– месенджер для спілкування зі студентами та викладачами.

Додаток для ОС Android складається з набору активностей, кожній з яких відповідає екран керування. Кожна активність представлена в проекті класом, реалізованим на мові Java, що зберігається в однойменному файлі з розширенням .java. Кожній активності відповідає xml файл-опис.

В xml-файлі описано у вигляді xml-коду розташування візуалізованих об'єктів. При запуску активності система Android автоматично розпізнає розмір екрану мобільного пристрою і призводить виведений контент у відповідність з розміткою, описаної в xml файлі. Таким чином, одна і та ж активність буде виглядати однаково незалежно від діагоналі використовуваного пристрою. Також, для кожної програми Android повинен існувати xml файл, в якому у вигляді xml коду будуть прописані мінімальні вимоги до системи, а також активність, яка викликається при запуску програми [18].

Розроблюваний мобільний додаток побудований на архітектурі клієнт-серверного додатка. Клієнт-серверні додатки є найпоширенішими і в той же час найскладнішими в розробці. Проблеми виникають на будь-якому етапі, від вибору засобів для виконання запитів до методів кешування результату. Архітектура "клієнт-сервер» визначає загальні принципи організації взаємодії в мережі, де є сервери, вузли-постачальники деяких специфічних функцій (сервісів) і клієнти, споживачі цих функцій.

Як правило комп'ютери і програми, що входять до складу інформаційної системи, не є рівноправними. Деякі з них володіють ресурсами (файлова система, процесор, принтер, база даних та ін.), інші мають можливість звертатися до цих ресурсів. Комп'ютер (або програму), керуючий ресурсом, називають сервером цього ресурсу (файл-сервер, сервер бази даних, обчислювальний сервер). Клієнт і сервер будь-якого ресурсу можуть знаходиться як в рамках однієї обчислювальної системи, так і на різних комп'ютерах, пов'язаних мережею.

У будь-якій мережі (навіть тимчасовій), побудованої на сучасних мережевих технологіях, присутні елементи клієнт-серверної взаємодії, найчастіше на основі дворівневої архітектурі. Дворівневою (two-tier, 2-tier) вона називається через необхідність розподілу трьох базових компонентів між двома вузлами (клієнтом і сервером).

Дворівнева архітектура використовується в клієнт-серверних системах, де сервер відповідає на клієнтські запити безпосередньо і в повному обсязі, при цьому використовуючи тільки власні ресурси. Тобто сервер не викликає сторонні мережеві програми, але не звертається до сторонніх ресурсів для виконання будь-якої частини запиту (рис. 3.6).

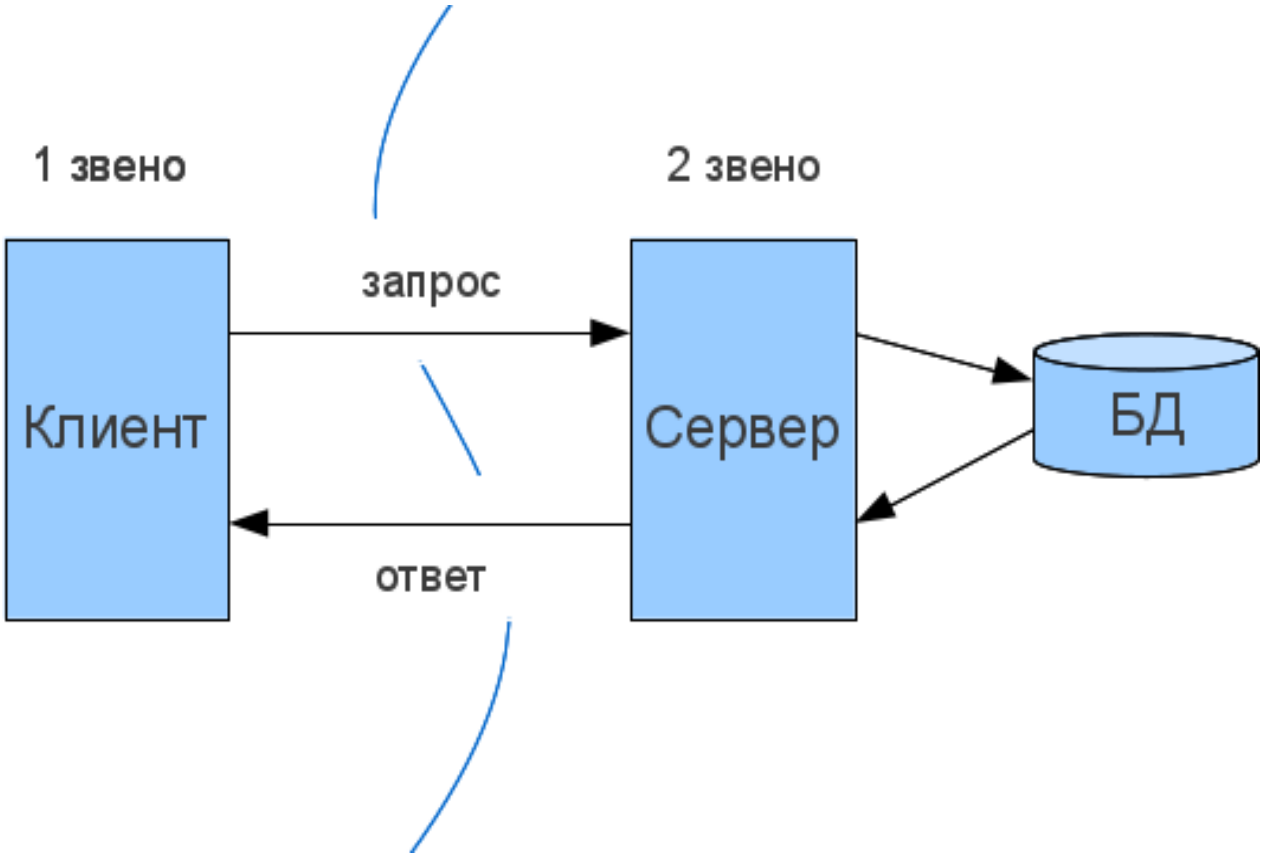

Рисунок 3.6 – Дворівнева клієнт-серверна архітектура

Розташування компонентів на стороні клієнта або сервера визначає наступні основні моделі їх взаємодії в рамках дворівневої архітектури:

– сервер терміналів – розподілене представлення даних;
– файл-сервер – доступ до віддаленої базі даних і файлових ресурсів;
– сервер БД – віддалене уявлення даних;
– сервер додатків – віддалений додаток.

Не дивлячись на те, що дворівнева архітектура доволі зручна, вона не відповідає вимогам мобільного додатку. Саме тому було обрано трирівневу архітектуру.

Трирівнева архітектура – архітектурна модель програмного комплексу, що передбачає наявність у ньому трьох компонентів: клієнта, сервера додатків (до якого підключено клієнтську програму) і сервера баз даних (з яким працює сервер додатків).

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відділяється окремий програмний рівень, на якому зосереджується прикладна логіка програми. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів додатків, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних (рис. 3.7).

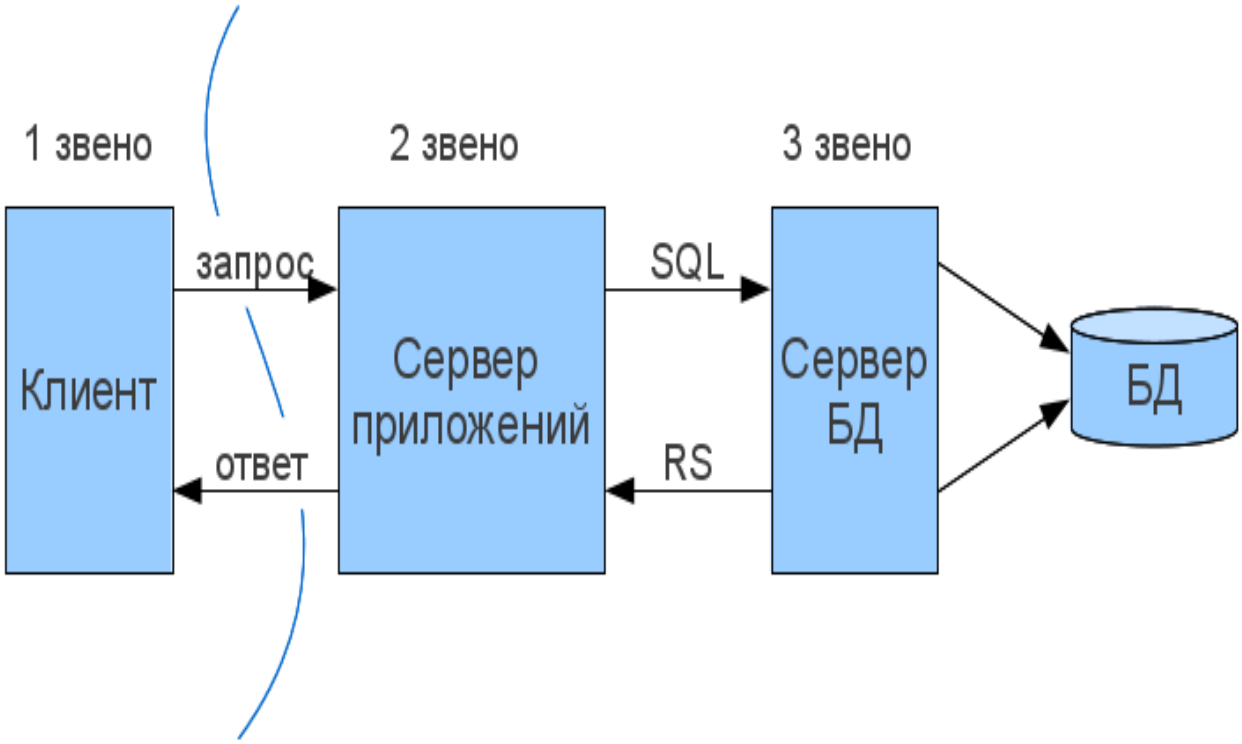

Рисунок 3.7 – Трирівнева клієнт-серверна архітектура

Клієнт – це інтерфейсний (зазвичай графічний) компонент комплексу, що надається кінцевому користувачу. Цей рівень не повинен мати прямих зв'язків з базою даних (за вимогами безпеки і масштабованості), бути навантаженим основний бізнес-логікою (за вимогами масштабованості) і зберігати стан додатки (за вимогами надійності). На цей рівень зазвичай виноситься тільки найпростіша бізнес-логіка: інтерфейс авторизації, алгоритми шифрування, перевірка значень, що вводяться, на допустимість і відповідність формату, нескладні операції з даними (сортування, угруповання, підрахунок значень), вже завантаженими на термінал.

Сервер додатків розташовується на другому рівні, на ньому зосереджена велика частина бізнес-логіки. Поза його залишаються тільки фрагменти, що експортуються на клієнта, а також елементи логіки, занурені в базу даних (збережені процедури і тригери). Реалізація даного компонента забезпечується сполучною програмним забезпеченням. Сервери додатків проектуються таким чином, щоб додавання до них додаткових примірників забезпечувало горизонтальне масштабування продуктивності програмного комплексу і не вимагало внесення змін до програмний код програми.

Сервер баз даних забезпечує зберігання даних і виноситься на окремий рівень, реалізується, як правило, засобами систем управління базами даних, підключення до цього компоненту забезпечується тільки з рівня сервера додатків.

У найпростіших конфігураціях все компоненти або частина з них можуть бути суміщені на одному обчислювальному вузлі. У продуктивних конфігураціях як правило використовується виділений обчислювальний вузол для сервера баз даних або кластер серверів баз даних, для серверів додатків – виділена група обчислювальних вузлів, до яких безпосередньо підключаються клієнти (термінали).

У порівнянні з клієнт-серверної або файл-серверної архітектурою трирівнева архітектура забезпечує, як правило, більшу масштабованість (за рахунок горизонтальної масштабованості сервера додатків і мультиплексування з'єднань, велику конфігурованість (за рахунок ізольованості рівнів один від одного), більш широкі можливості по забезпеченню безпеки і відмовостійкості. Крім того, в порівнянні з клієнт-серверними додатками, що використовують прямі підключення до серверів баз даних, знижуються вимоги до швидкості і з табільності каналів зв'язку між клієнтом і серверної частиною. Реалізація програм, доступних з веб-браузера або з тонкого клієнта, як правило, має на увазі розгортання програмного комплексу в трирівневої архітектурі.

При цьому зазвичай розробка додатків для трирівневої програмних комплексів складніше, ніж для клієнт-серверних додатків , також наявність додаткового сполучного програмного забезпечення може накладати додаткові витрати в адмініструванні таких комплексів.

3.6 UML-діаграма класів

Для розробки моделей використовуються засоби моделювання, що представляють собою комбінацію уніфікованої мови моделювання UML (Unified Modeling Language) і мови об'єктних обмежень OCL (Object Constraint Language). Зазначені мови є, сьогодні, фактичним стандартом розробки комерційних програмних проектів. Уніфікована мова моделювання UML є діаграматиною і моделі, що представляються в цій мові, це – діаграми, побудовані з використанням графічних символів і правил, прийнятих в UML.

Діаграматине уявлення моделі наочно, легко і швидко сприймається людиною, і в концентрованому вигляді містить велику кількість інформації щодо модельованої системи. Однак, як правило, моделі, представлені тільки UML-діаграмами, містять значну кількість невизначеності і їх недостатньо для їх відображення в завершений програмний код. Мова об'єктних обмежень OCL є символьним мовою, який розроблений спеціально як доповнення UML, що дозволяє уточнювати діаграматину модель. Уточнення моделі здійснюється за допомогою пропозицій, що формулюють обмеження, в рамках яких, повинні функціонувати ті чи інші модельні елементи.

Комбінуючи UML-діаграми, специфікуючу структуру, і OCL-пропозиції, специфікуючі обмеження, можна отримати модель, що містить мінімум невизначеності і легко відображається в програмний код. Середовище моделювання, що представляє собою комбінацію UML і OCL, базується на ідеях об'єктно-орієнтованої парадигми програмування, і тому цінність моделей, синтезованих в цьому середовищі, полягає в тому, що вони є специфікаціями, достатніми для «ручної» або «автоматичної» генерації програмного коду в рамках цієї парадигми.

Переважна більшість сучасних CASE-систем (Computer Aided Software Engineering) в змозі генерувати програмний код, використовуючи UML-OCL моделі в якості вихідних специфікацій.

UML – є графічною мовою для візуалізації, опису параметрів, конструювання та документування різних систем (програм зокрема). Діаграми

створюються за допомогою спеціальних CASE засобів, наприклад Rational Rose і Enterprise Architect. На основі технології UML будується єдина інформаційна модель. Наведені вище CASE засоби здатні генерувати код на різних об'єктно-орієнтованих мовах, а так само мають дуже корисною функцією реверсивного інжинірингу. (Реверсивний інжиніринг дозволяє створити графічну модель з наявного програмного коду і коментарів до нього.)

Діаграма класів – статична діаграма. Являє собою статичний вид програми. Діаграма класів використовується не тільки для візуалізації, опису і документування різних аспектів системи, але і для створення виконуваного коду програмного додатка.

Діаграма класів описує атрибути і операції класу, а також обмеження, що накладаються на систему. Діаграма класів широко використовуються при моделюванні об'єктно-орієнтованих систем, оскільки вони є єдиними діаграмами UML, які можуть відображатися безпосередньо з об'єктно-орієнтованими мовами.

В ході роботи була побудована діаграма класів, що показує набір класів, інтерфейсів, асоціацій, спільної роботи і обмежень мобільної інформаційної системи по підготовці студентів до контролю знань (рис. 3.8).

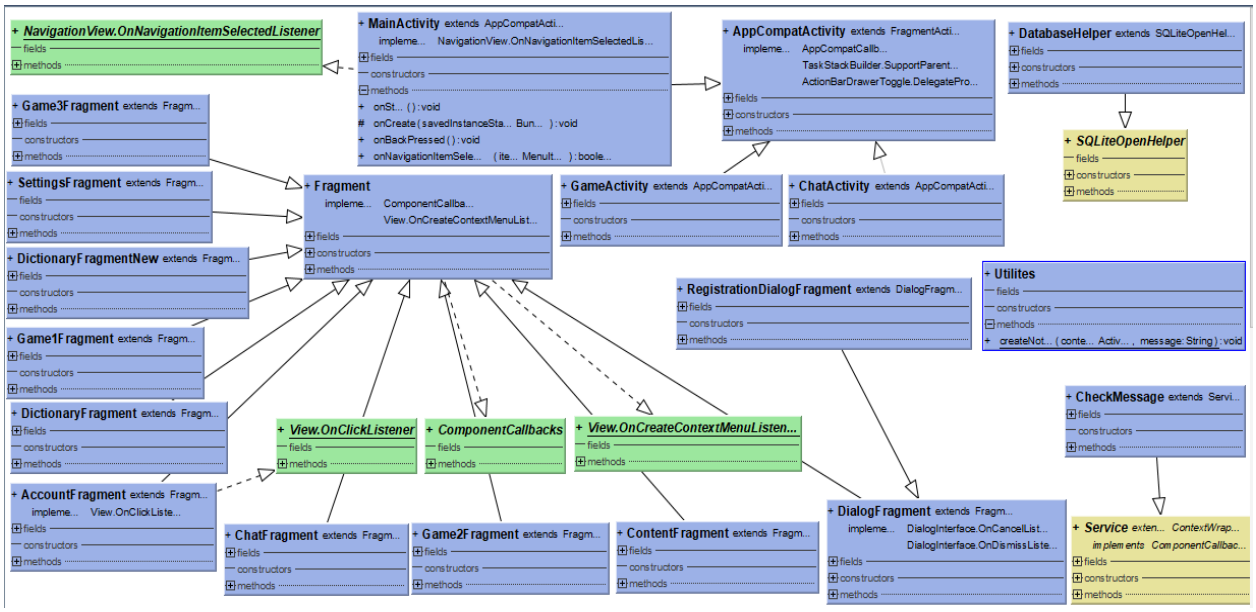

Рисунок 3.8 – UML діаграма класів

3.7 Діаграма варіантів використання

Діаграма використання UML – це основна форма вимог до системи/програмного забезпечення для нової програми, що розробляється. У ви-

падках використання вказується очікувана поведінка (що), а не точний спосіб його вчинення (як). Варіанти використання можна зобразити як текстовим, так і візуальним уявленням (наприклад, UML). Ключовою концепцією моделювання варіантів використання є те, що вона допомагає розробляти систему з точки зору кінцевого користувача. Це ефективний метод передачі поведінки системи в призначені для користувача терміни шляхом вказівки всього зовні видимого поведінки системи.

Діаграма використання – графічне зображення взаємодій між елементами системи.

Варіант використання – це методологія, яка використовується в системному аналізі для виявлення, уточнення і організації системних вимог. У цьому контексті термін «система» відноситься до того, що розробляється або експлуатується, наприклад, до веб-сайту з продажу та обслуговування товарів поштою. Діаграми використання використовуються в UML (Unified Modeling Language), стандартної нотації для моделювання об'єктів і систем реального світу.

Мета варіанту використання полягає в тому, щоб визначити закінчений аспект або фрагмент поведінки деякої сутності без розкриття внутрішньої структури цієї сутності. В якості такої сутності може виступати вихідна система або будь-який інший елемент моделі, який володіє власною поведінкою, подібно підсистемі або класу в моделі системи.

Цілі системи можуть включати в себе планування загальних вимог, перевірку апаратного проектування, тестування і налагодження розроблюваного програмного продукту, створення довідкової довідки по оперативної допомоги або виконання завдання, орієнтованої на споживача. Наприклад, приклади використання в середовищі продажів продукту включають в себе упорядкування товарів, оновлення каталогу, обробку платежів і відносини з клієнтами. Діаграма використання містить чотири компоненти:

– межа, яка визначає систему інтересів щодо навколишнього світу;

– актори, як правило, особи, пов'язані з системою, визначені відповідно до їх ролей;

– приклади використання – це конкретні ролі, які відіграють актори всередині та навколо системи;

– взаємовідносини між акторами, а також випадки їх використання.

В ході роботи була побудована діаграма варіантів використання, що показує можливі дії користувача та його взаємодію з сервером мобільної інформаційної системи по підготовці студентів до контролю знань (рис. 3.9).

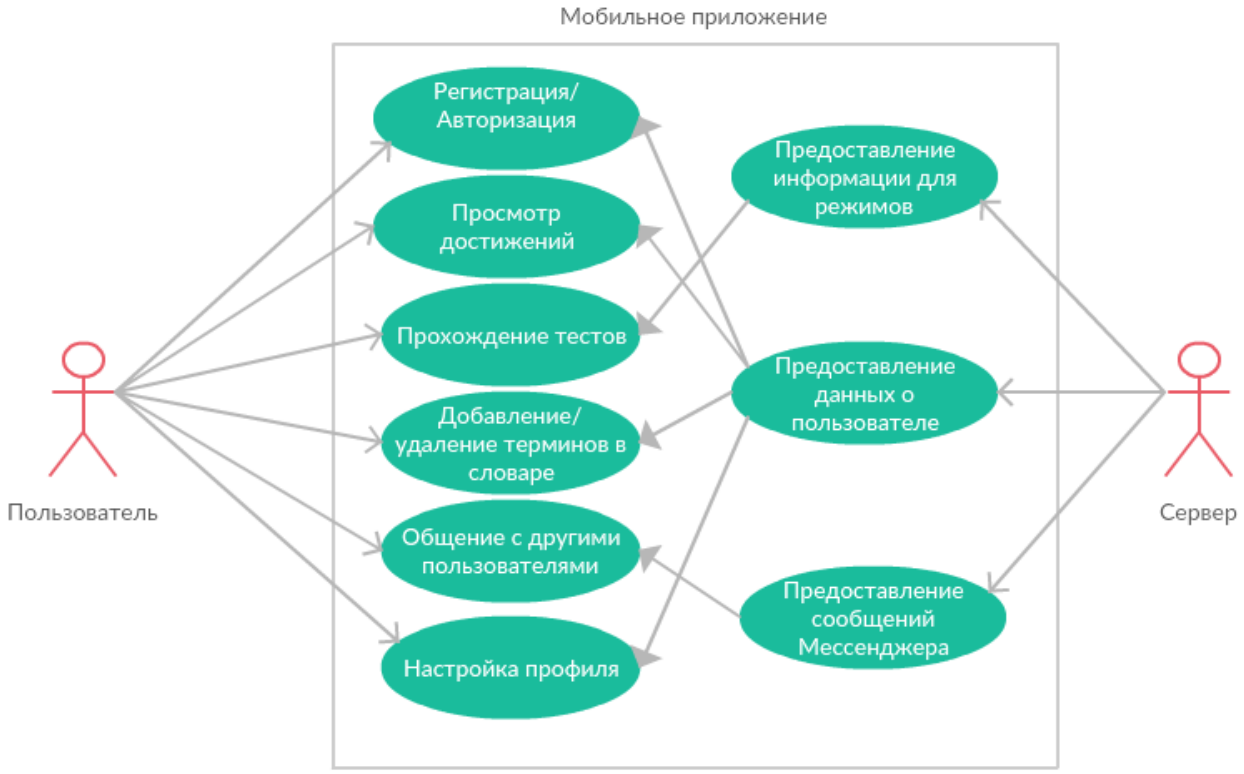

Рисунок 3.9 – UML діаграма варіантів використання

4 РОБОТА ДОДАТКУ

4.1 Опис функціональності

При запуску програми запускається головна активність, для користувача надається можливість авторизувати або зареєструватися (рис. 4.1).

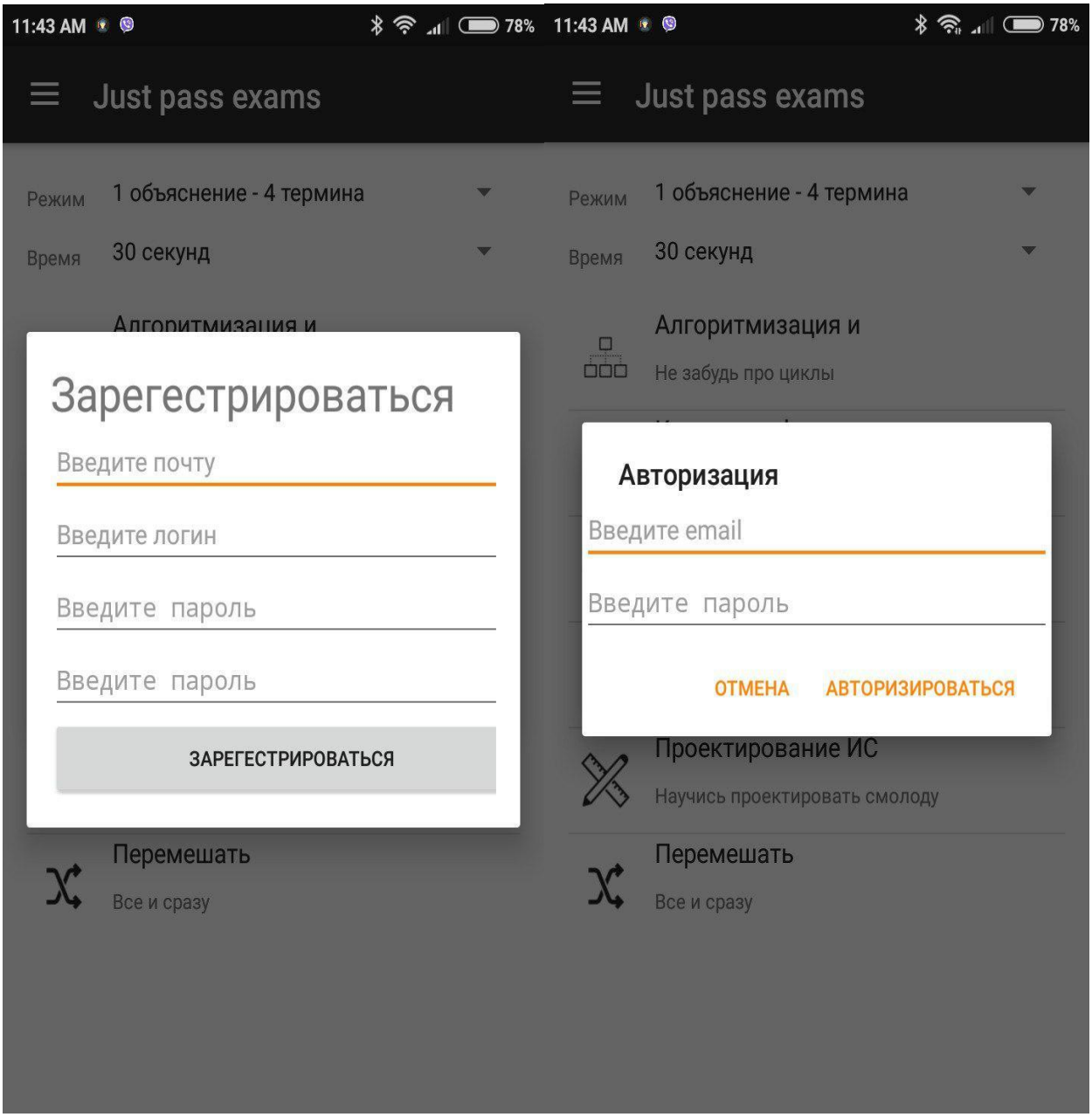

Рисунок 4.1 – Активності авторизації та реєстрації

Також при реєстрації і авторизації передбачене повідомлення користувача про помилки в разі введення невірної інформації у відповідні полях до тих пір, поки дані не пройдуть валідацію (рис. 4.2).

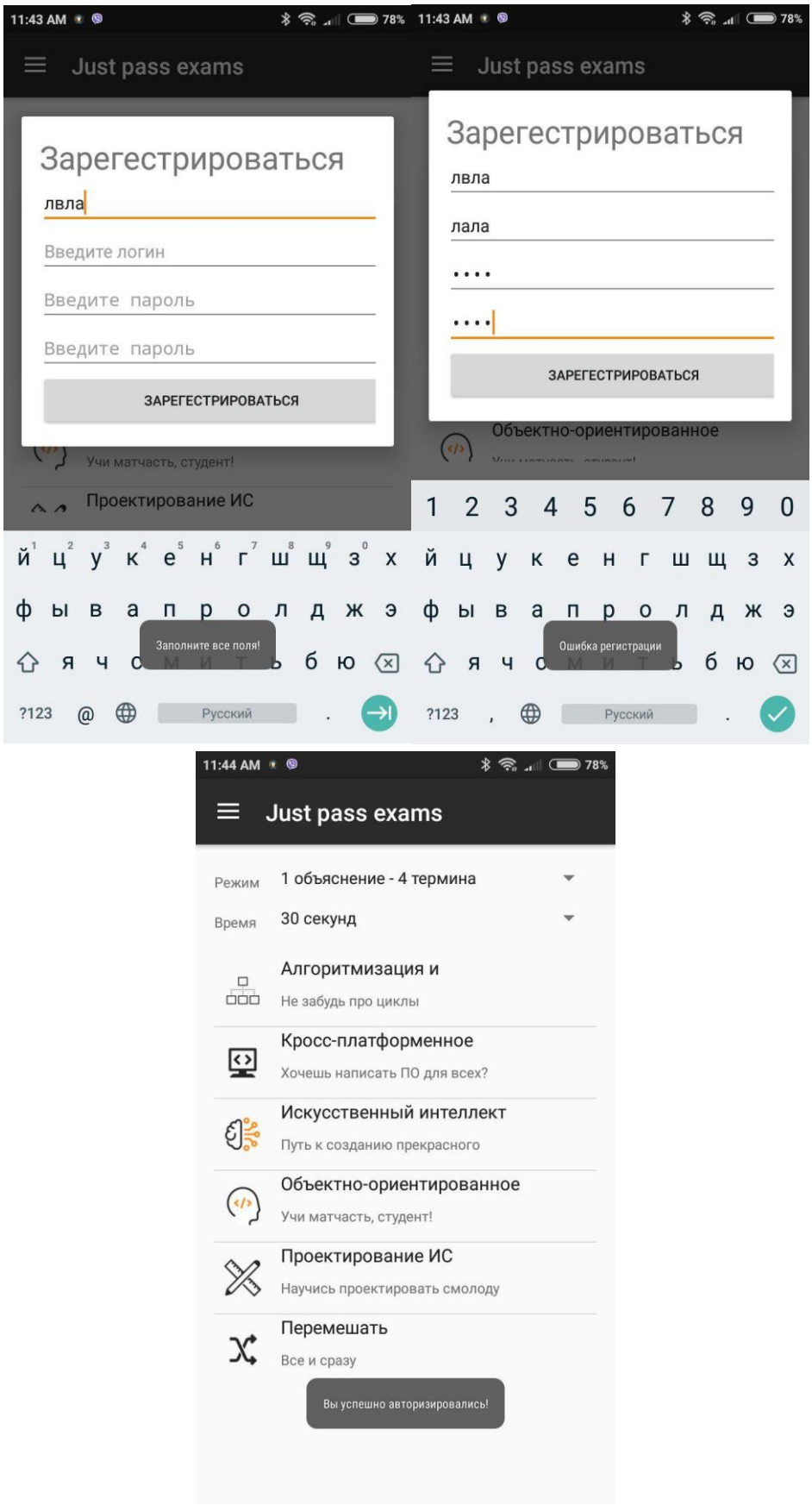

Рисунок 4.2 – Повідомлення про помилки при реєстрації та успішну авторизацію

При успішній авторизації користувач бачить перед собою активність з доступними темами і перемикачі режиму та часу проходження тесту. А також кнопку відкриття активності навігації (рис. 4.3).

В активності навігації користувач може вибрати будь-який з пунктів, які виконують свою функціональність і змінити налаштування свого профілю.

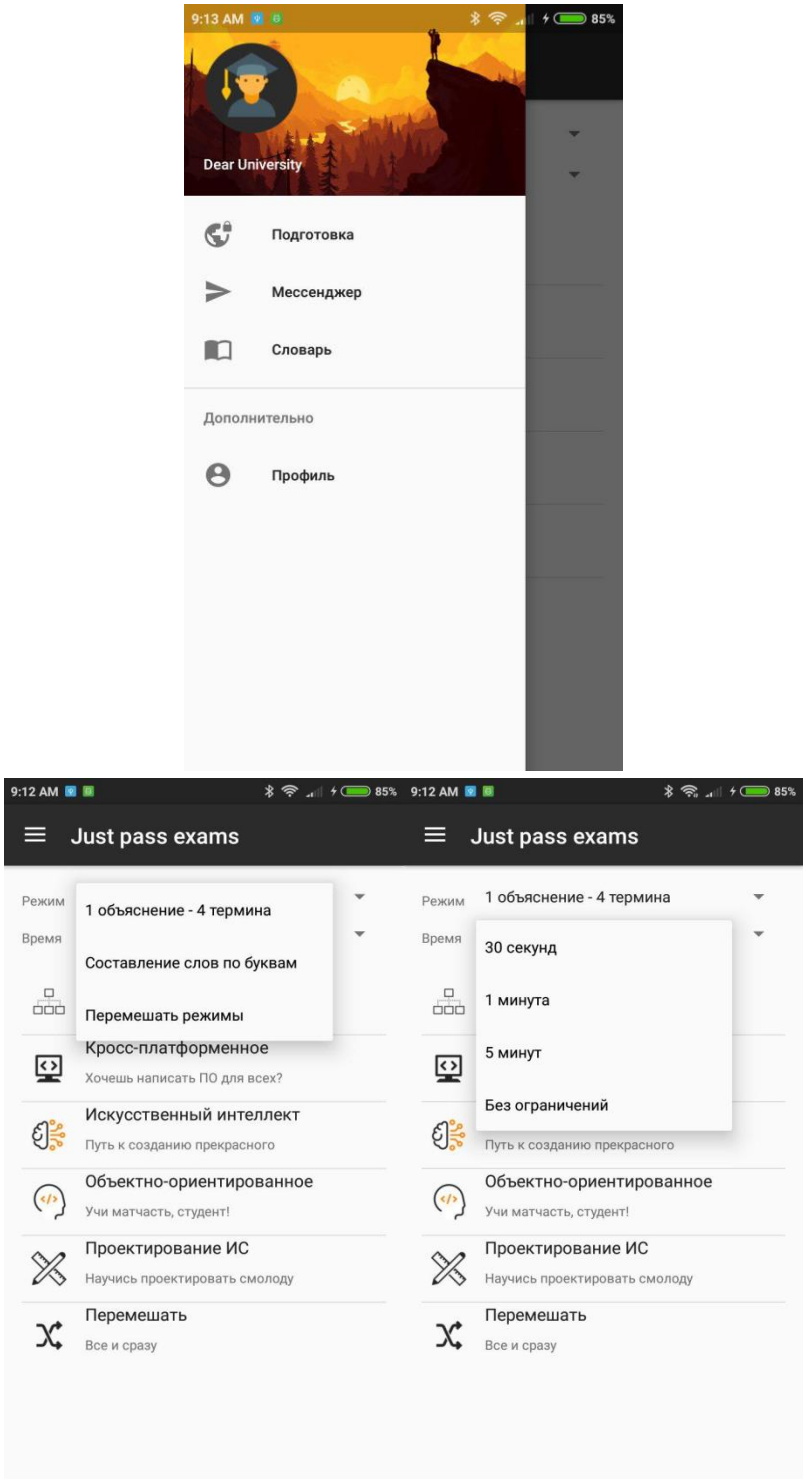

Рисунок 4.3 – Головна активність та активність навігації

Користувач може обрати одну з тематик тестування і підготуватися до майбутнього іспиту. У цьому йому допоможе як і передбачені теми і тести, так і можливість додати в словник свої терміни і визначення.

Кожен режим може бути обмежений певним часом, також може бути і без обмежень. Вгорі відображається лічильники часу і кількості правильних відповідей – студент завжди може бачити свій результат.

У додатку існує три режими: 1 пояснення – 4 терміни, складання слів по буквах і перемішаний режим.

Перший режим являє собою графічне представлення звичайного тестування, звичного для студента (рис. 4.4). Його метою є підготовка студента до вибору з декількох варіантів, де правильним буде лише один. У разі неправильної відповіді студент закінчує свою спробу і може повторити знову.

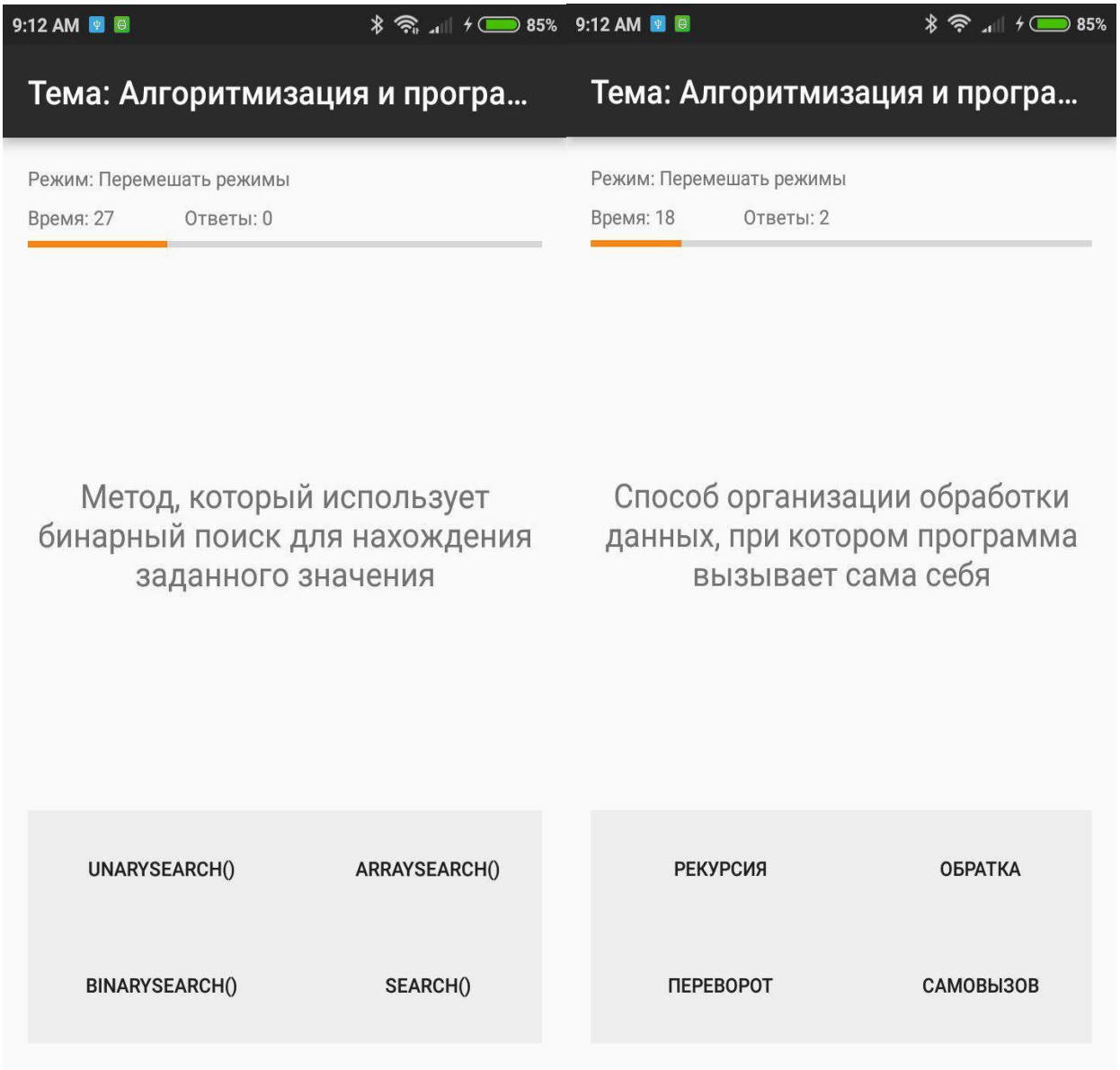

Рисунок 4.4 – Режим “1 пояснення – 4 терміни”

Другий режим "Складання слів по буквах" являє собою кросворд, в якому перемішані літери і студенту необхідно встановити правильний порядок букв для того, щоб дати правильну відповідь (рис. 4.5). Даний режим складніше першого і дозволяє студенту краще запам'ятовувати правильну відповідь, не відволікаючи його на роздумування і хаотичний вибір серед варіантів. Також в режимі передбачено очищення введення в разі, якщо студент невірно почав вибирати літери.

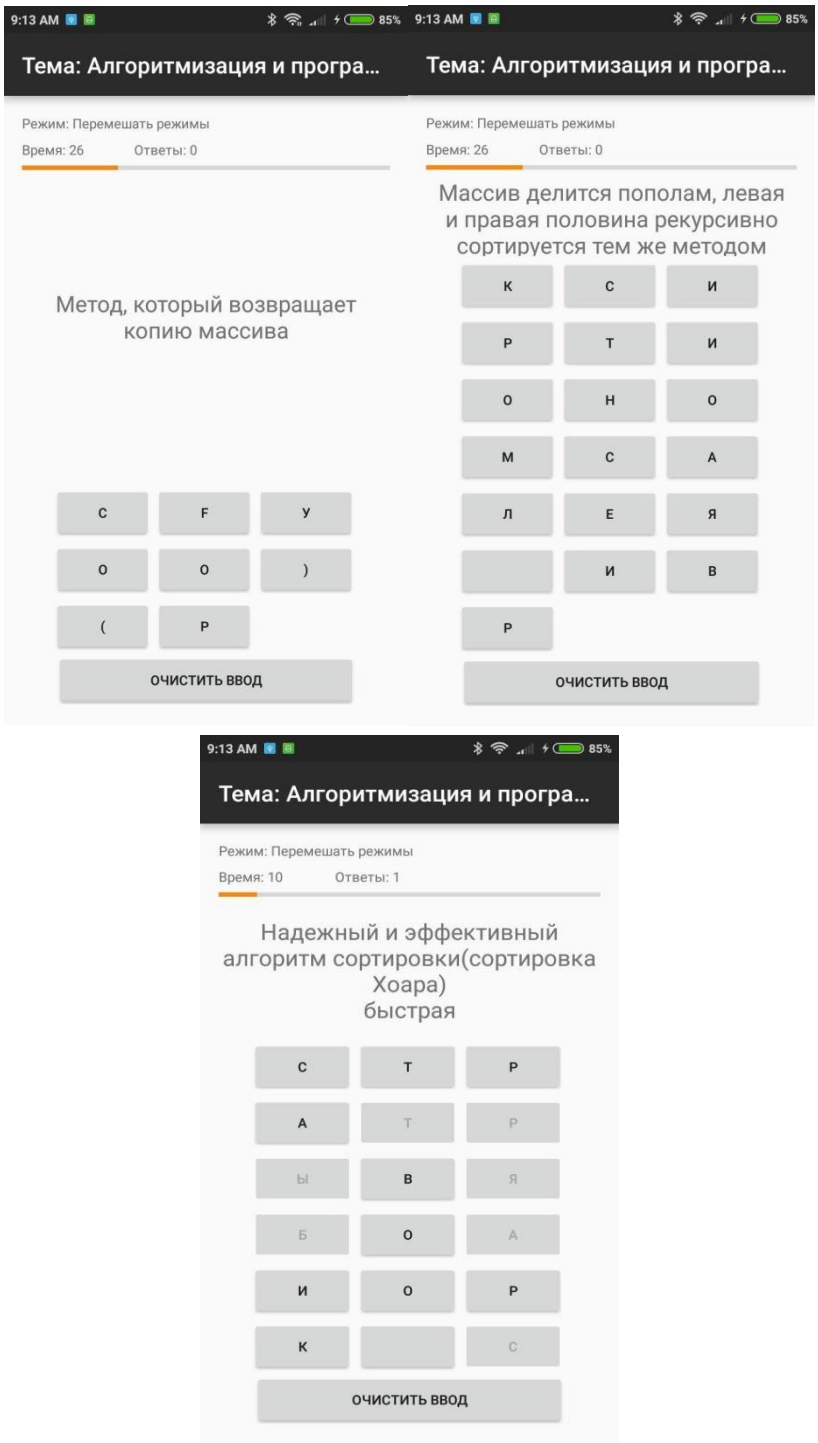

Рисунок 4.5 – Режим “Складання слів по буквах”

Третій режим "Перемішаний режим" являє собою комбінацію всіх доступних режимів та у випадковій послідовності видає студенту різні питання (рис. 4.6). Має всі ті ж функції і обмеження, як інші режими. Тренує в студента адаптацію до певних умов і раптової зміни оточення.

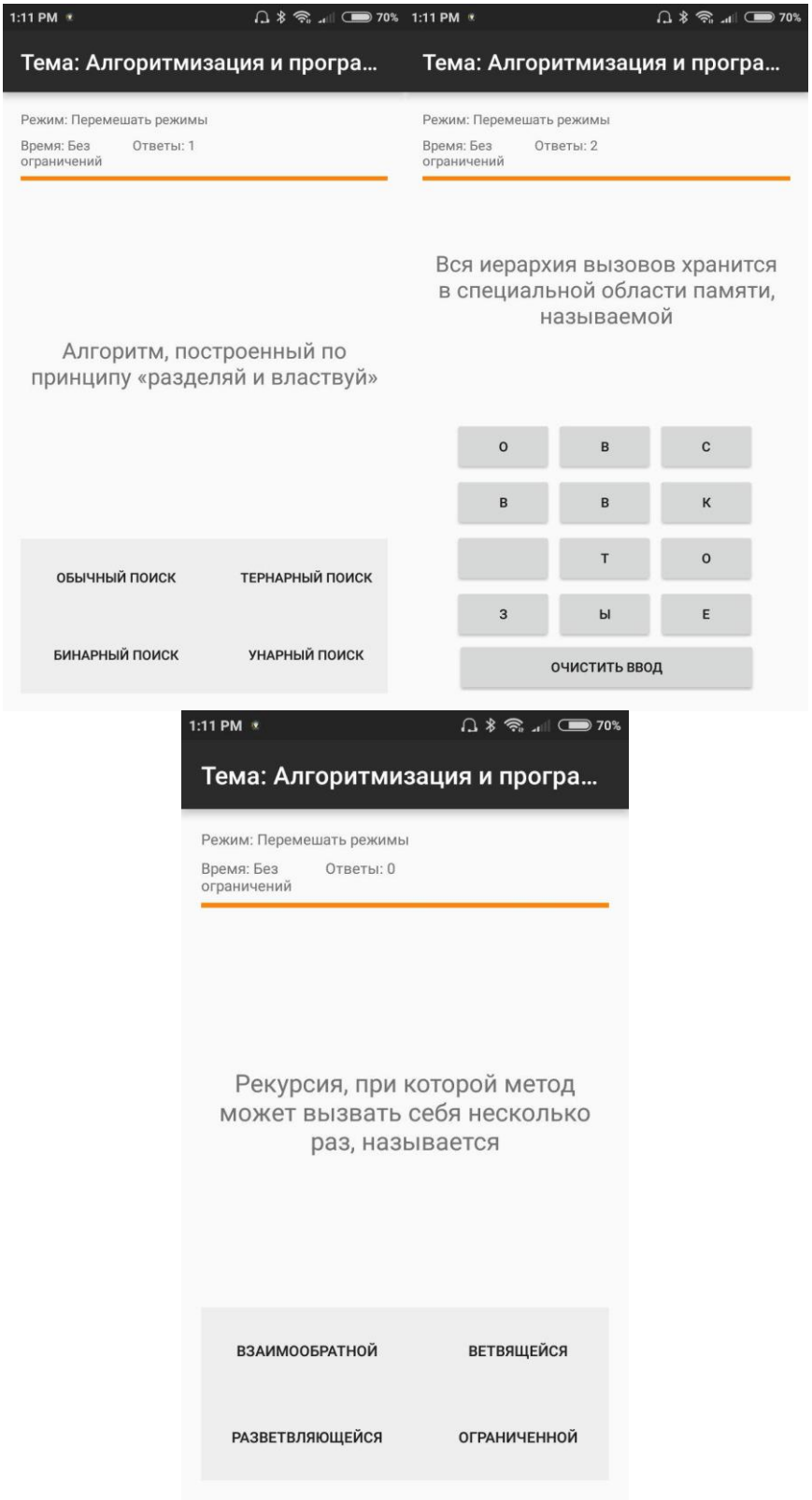

Рисунок 4.6 – Режим “Перемішаний режим”

Також можна перемішати все теми між собою і вивчати відразу всю інформацію. В навігації крім підготовки є Мессенджер. Функції месенджера недоступні до тих пір, поки студент не набере хоча б 50 балів, тобто не дасть 50 правильних відповідей при підготовці. Мессенджер розділений на кімнати, кожна кімната відповідає тематиці, в якій студент набрав необхідну кількість балів. Завдяки своєчасним повідомленнями, користувач ніколи не упустить нові повідомлення, які з'явилися в месенджері (рис. 4.7).

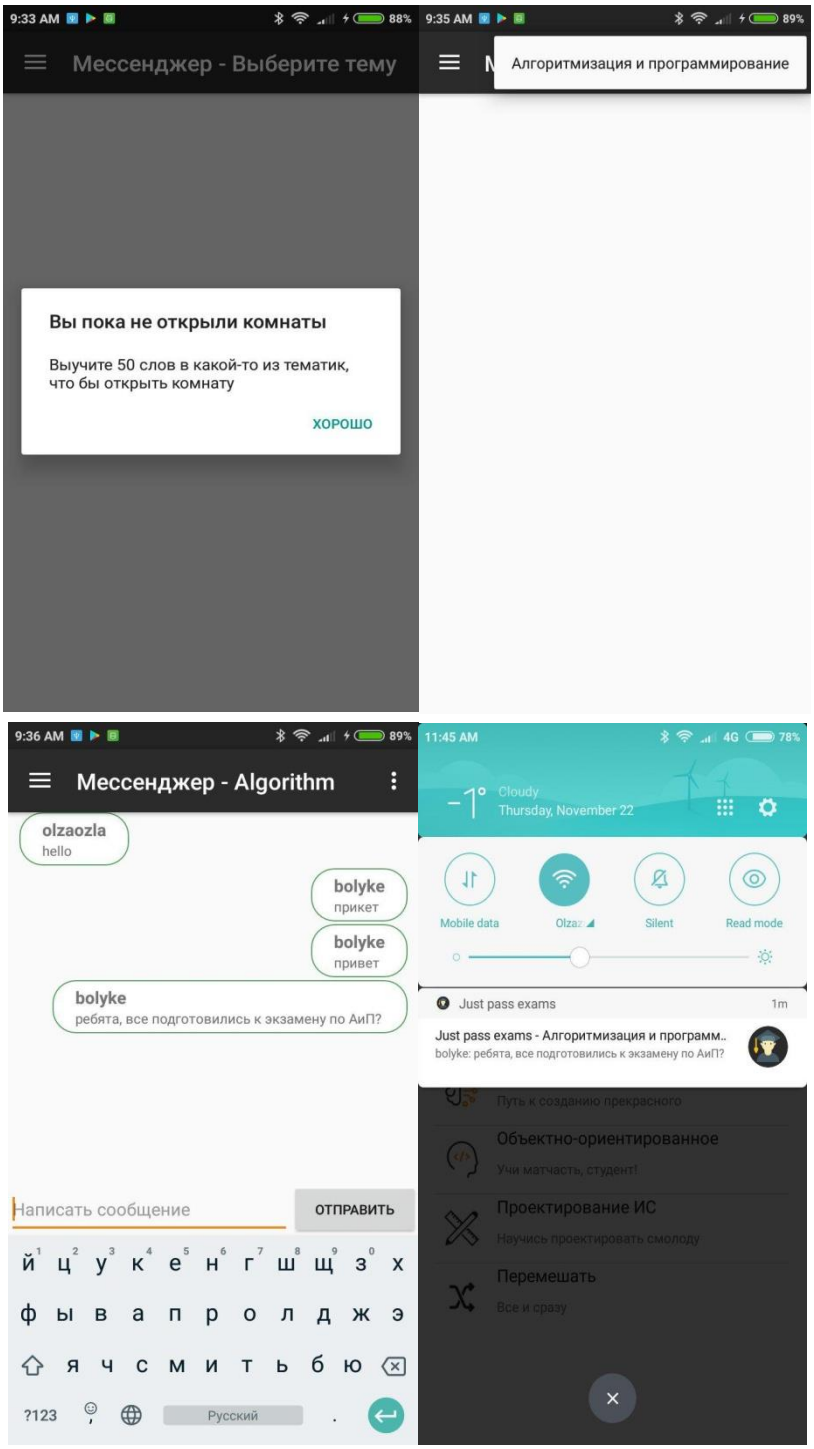

Рисунок 4.7 – Функціональність Мессенджер

У мобільному додатку передбачено словник студента, в який він може додати безліч нових визначень і повторювати їх у зручний для нього час. Кількість визначень не обмежена ні кількістю, ні тематиками (рис. 4.8). Користувач має можливість в будь-який зручний йому час підготуватися до необхідного контрольного тесту. Додаток працює без доступу в інтернет, тому у студента буде доступ до питань навіть в зоні, де немає покриття провайдера.

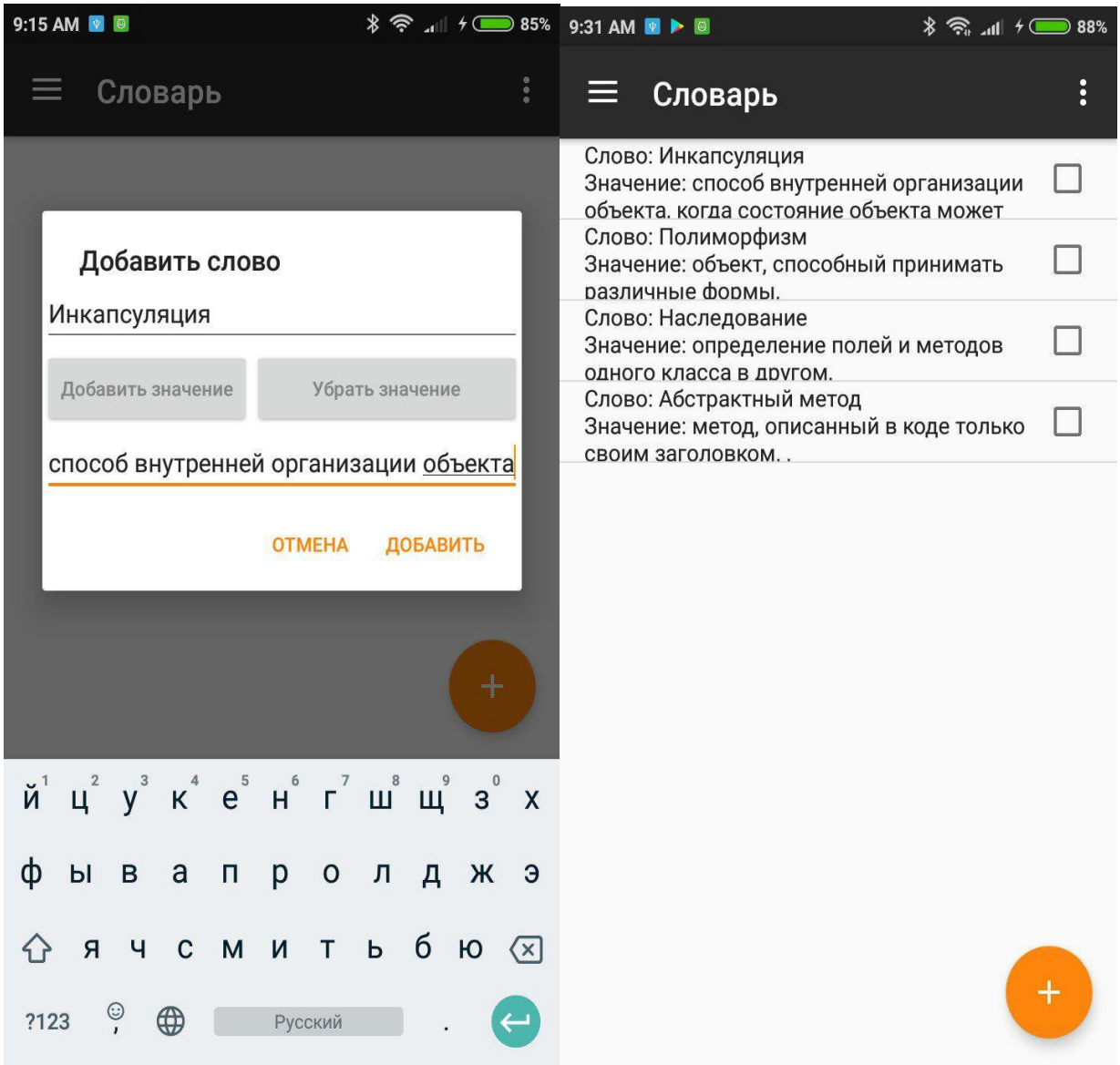

Рисунок 4.8 – Функціональність Словник

Крім додавання визначень в словник також є можливість видалення вже не потрібних визначень (рис. 4.9). Дана можливість буде дуже корисна тим студентам, які навчаються не перший рік, і необхідно видалити інформацію за минулі семестри, або тим, хто зробив помилку в написанні, та необхідно замінити позначення на вірне.

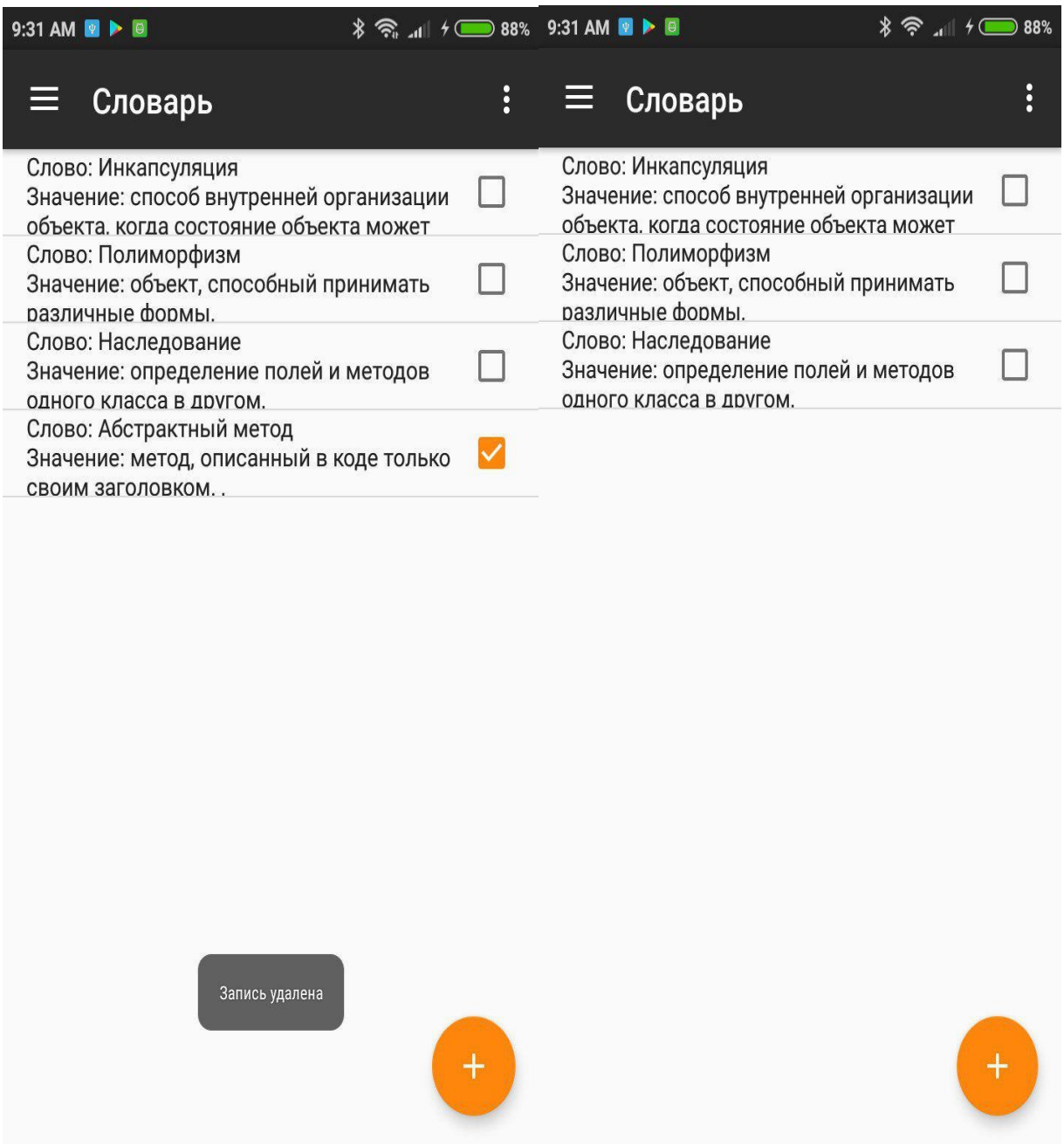

Рисунок 4.9 – Функціональність видалення термінів

Нарешті, в додатку є можливість переглянути свій профіль і в разі потреби поміняти інформації або вийти з програми (рис. 4.10). У профілі надана інформація про рівень користувача, вибрана мова та досягнення.

У досягнення користувача входить перелік доступних тематик і набрані бали для кожної.

Додаток є досить зручним у використанні і необхідним в житті звичайного студента. Воно дозволяє підготуватися до будь-яких складнощів у студентському житті і бути завжди в курсі нових подій. Дозволяє спілкуватися з однокурсниками і обговорювати майбутні іспити, заліки або модулі.

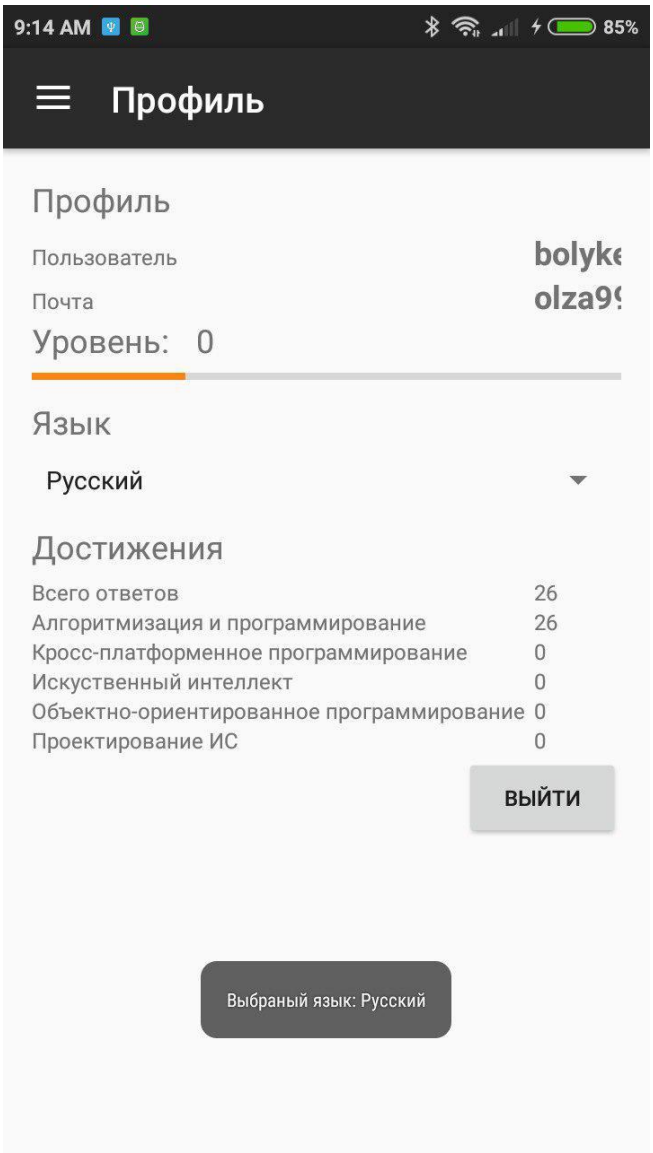

Рисунок 4.10 – Активність Профіль

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.

В ході роботи було виконано проектування і розробку мобільного додатку для підготовки студентів до іспитів, заліковим, контрольним і модульним робіт.

У додатку зареєстрований користувач має можливість перегляду своєї статистики тестів, додавання термінів в особистий словник, спілкування з одногрупниками і викладачами, проходження тестів в різних режимах і на різні теми.

Не завжди є можливість читати методичний посібник або книгу. На сьогоднішній день люди багато часу проводять в громадському транспорті. У тому числі і студенти: по дорозі до навчального закладу і додому. Це той золотий час, коли можна поглинати інформацію в усіх її обсягах і не відволікатися на побутові проблеми. І це сама найпростіша причина, чому додаток з підготовки до контролю знань може стати в нагоді студенту. Адже воно є мобільним довідник, в якому є вся необхідна інформація для підготовки, а навіть якщо її немає – студент сам може внести її і покращувати свої знання в будь-який час і в будь-якому місці.

ПЕРЕЛІК ПОСИЛАНЬ

1. Голицина М. Н., Половникова Н. Л. Мобильное обучение как новая технология в образовании // Образовательные технологии и общество. – СПб.: Питер, 2011. – 252 с.
2. Хортон У., Хортон К. Электронное обучение: инструменты технологии / Пер. с англ. М.: КУДИЦ-ОБРАЗ, 2009. – 640 с.
3. Баженов Р. И., Корнилков А. П., Лопатин Д. К. Проектирование web-ориентированной информационной системы университета на основе клиент-серверных технологий // Актуальные проблемы гуманитарных и естественных наук. – СПб.: Питер, 2014. – 244 с.
4. ABFM Exam Preparation. [Електронний ресурс] – Режим доступу: https://play.google.com/store/apps/details?id=org.abfm.examprep
5. College Entrance Exams Prep. [Електронний ресурс] – Режим доступу: play.google.com/store/apps/details?id=com.youth4work.College_Entrance_ExaminationComputer
6. Science Solved MCQs. [Електронний ресурс] – Режим доступу: play.google.com/store/apps/details?id=com.threemajors.computermcqs
7. Коматинени С. Android 4 для профессионалов. Создание приложений для планшетних компьютеров и смартфонов: Научно-популярное издание. – Санкт-Петербург: ТОВ «И.Д. Вильямс », 2012. – 880 с.
8. Дон Гриффитс, Девид Гриффитс Программирование для Android. – СПб.: Питер, 2018. – 912 с.
9. Билл Филлипс, К. Стюарт, Кристин Марсикано Android. Программирование для профессмоналов. – СПб.: Питер, 2017. – 688 с.
10. Ян Клифтон Проектирование пользовательского интерфейса на Android. – Санкт-Петербург: ДМК Прес, 2017. – 452 с.
11. Тереза Нейл Мобильная разработка. Галерея шаблонов. – СПб.: Питер, 2013. – 216 с.
12. Юлия Сирых Срвременный веб-дизайн. Настольний и мобильний. – М.: Вильямс, 2018. – 384 с.
13. Леонтьев Б.В. Web-Дизайн: Тонкости, хитрости и секреты / Леонтьев Б.В. – М.: Майор, 2001. – 170 с.
14. Найбільший в Європі ресурс для IT-фахівців. [Електронний ресурс] – Режим доступу: https://habrahabr.ru
15. Прамодкумар Дж. Садаладж, Мартин Фаулер NoSQL. Новая методология разработки нереляционних баз даных. – М.: Вильямс, 2017. – 192 с.

16. Дейт К.Дж. Введение в системы баз данных: Пер. с англ. – 6-е изд. – К.: Диалектика, 1998. – 784 с.

17. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.

18. Алексей Голощапов Google Android: програмирование для мобильные устройства. – СПб.: БХВ-Петербург, БХВ-Петербург, 2011. – 448 с.