

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Он-лайн словник для ОС Android

Освітньо-кваліфікаційний рівень - бакалавр
Напрямок – 6.050102 “Комп’ютерна інженерія”

Керівник проекту:

_____ (підпис)

Щербаков Є.В.

_____ (ініціали, прізвище)

Консультант з охорони праці:

_____ (підпис)

Критська Я.О.

_____ (ініціали, прізвище)

Студент:

_____ (підпис)

Розкін Д.Г.

_____ (ініціали, прізвище)

Група:

_____ КІ-14з

Севєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки 6.050102 "Комп'ютерна інженерія"
(шифр і назва)
Спеціальність _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
_____ І.С. Скарга-Бандурова
« _____ » _____ 20__ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Розкіну Дмитру Григоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Онлайн словник для ОС Android

керівник проекту (роботи) доц. Щербаков Є.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " " 20__р. № _____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз та постановка задачі; особливості розроблення додатків для ОС, Android, існуючих засобів розробки, проектування додатку.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Комп'ютерна презентація

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Ст. викладач Критська Я. О.		

7. Дата видачі завдання _____

Керівник доц. Щербаков Є.В. _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Постановка завдання	14.05.18-19.05.18	
2	Аналіз предметної області	20.05.18-25.05.18	
3	Вибір засобів для розробки системи	26.05.18-02.06.18	
4	Розробка структури системи	03.06.18-06.06.18	
5	Розробка складових частин системи	07.06.18-09.06.18	
6	Оформлення пояснювальної записки	14.06.18-15.06.18	

Студент _____
(підпис)

Розкін А.І.
(прізвище та ініціали)

Керівник _____
(підпис)

доц. Щербаков Є.В.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра: 86 с., 28 рис., 3 табл., 21 бібліографічне джерело, 2 додатки.

Об'єкт розробки: розділи Android-додатку.

Мета роботи: розробка словника для ОС Android .

В проекті виконано:

1. У розділі “Дослідження предметної області та постановка задачі” було виконано зрівняння аналогічних систем, дослідження галузі словникових додатків, були поставлені задачі щодо розробки системи.

2. У розділі “Обґрунтування інструментів для розробки онлайн-словника для ОС Android” були розглянуті інструменти, за допомогою яких буде розроблятися система.

3. У розділі “Проектування онлайн-словника для ОС Android” описано проектування додатку.

4. У розділі “Охорона праці” був проведений аналіз шкідливих виробничих факторів. На основі цього аналізу запропоновані заходи усунення цих факторів.

Отримано наступні результати: Розроблений онлайн-словник для ОС Android може використовуватися для реалізації доступності в сфері електронних словників та має функції завдяки яким має конкурентоспроможність.

Практичне значення, галузь застосування роботи: освітня галузь.

Ключові слова: словник, додаток, Android, Інтернет, Java-програмування, комп'ютерні компоненти.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А,. м. Сєвєродонецьк, 93400.

ЗМІСТ

ВСТУП.....	6
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	7
1.1 Мережа Інтернет.....	7
1.2 Мобільний додаток.....	7
1.3 Актуальність використання онлайн-словника.....	11
1.4 Порівняння з аналогічними системами.....	12
1.5 Технічне завдання.....	13
2 ОБГРУНТУВАННЯ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ СИСТЕМИ УПРАВЛІННЯ ПРОДАЖАМИ.....	14
2.1 Android-програмування.....	14
2.2 Емуляція та Android Studio.....	16
2.3 Мова Java.....	17
2.4 Реляційна система управління базами даних SQLite.....	21
3 ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-СЛОВНИК ДЛЯ ОС ANDROID.....	24
3.1 Проектування структури системи.....	24
3.2 Розробка дизайну системи.....	25
3.3 Структура бази даних.....	27
3.4 Головна сторінка.....	28
3.5 Пошук.....	31
3.6 Розділи словника.....	32
3.7 Навігаційне меню.....	34
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	37
4.1 Загальні питання з охорони праці.....	37
4.1.1 Правові та організаційні основи охорони праці.....	38
4.1.2 Організаційно-технічні заходи з безпеки праці.....	39
4.2 Аналіз стану умов праці.....	39
4.2.1 Вимоги до приміщень.....	40
4.2.2 Вимоги до організації місця праці.....	40

4.2.3 Навантаження та напруженість процесу праці.....	41
4.3 Виробнича санітарія.....	42
4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу	42
4.3.2 Пожежна безпека	45
4.3.3 Електробезпека	45
4.4 Гігієнічні вимоги до параметрів виробничого середовища	46
4.4.1 Мікроклімат	46
4.4.2 Освітлення.....	47
4.4.3 Шум та вібрація, електромагнітне випромінювання	49
4.4.4 Вентилювання.....	49
4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	50
ВИСНОВОК.....	56
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	57
ДОДАТОК А.....	59
ДОДАТОК Б	81

ВСТУП

Мобільні телефони давно перестали бути чимось незвичайним і чудово справляються зі своєю функцією - є засобом комунікації між людьми. При цьому, недавно з'явилися, але вже міцно ввійшли в наше життя смартфони настільки функціональні, що важко сказати, чого вони не вміють: це і плеєр, і фотоапарат, і можливість використання Інтернет-ресурсів, та інше. По суті, всі смартфони стали невеликою копією комп'ютера, який постійно можна мати при собі. У наш час все більше і більше смартфонів, комунікаторів, планшетних ПК і інших видів пристроїв, зручних для використання як в повсякденному житті, так і в закордонних поїздках зокрема, випускаються на базі ОС Android. Які ж причини поширення даної операційної системи?

Головна мета цього дипломного проекту є проектування додатку на базі ОС Android, який представляє собою словник перекладу з англійської мови на українську. За допомогою цього додатку користувач, який має базові навички роботи з ОС Android, зможе виконувати пошук потрібних слів, прослуховувати їх, додавати або видаляти їх із обраного

В якості основного інструменту, який буде використовуватися при розробці, була вибрана мова Java, а також операційна система Android і деякі технології, які описані в цій пояснювальній записці.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Мережа Інтернет

Інтернет - всесвітня система об'єднаних комп'ютерних мереж для зберігання і передачі інформації. Часто згадується як Всесвітня мережа і Глобальна мережа, а також просто Мережа. Побудована на базі стека протоколів TCP / IP. На основі Інтернету працює Всесвітня павутина і безліч інших систем передачі даних. До середини 2015 року кількість користувачів досягла 3,3 млрд осіб. Багато в чому це було обумовлено значним поширенням стільникових мереж з доступом в Інтернет стандартів 3G і 4G, розвитком соціальних мереж і здешевленням вартості інтернет-трафіку

Сьогодні, коли світ так з'єднаний інтернетом, важко повірити в те, що всього півстоліття тому цієї мережі не існувало. У реальності, соціальних медіа, ваших улюблених сайтів новин, навіть слова "онлайн" не існувало б, якби не відбулася передача двох букв з комп'ютера в Лос-Анджелесі на інший комп'ютер в Менло-Парку (штат Каліфорнія) в 1969 році.

Інтернет почав своє існування завдяки людині на ім'я Боб Тейлор, який працював в Управлінні перспективних досліджень (англ. скор. ARPA) Міністерства оборони США у Вашингтоні. У його офісі було три комп'ютерних терміналу, кожен з яких був підключений до одного з проектів ARPA в штаті Массачусетс, в Лос-Анджелесі і Менло-Парку. У Тейлора викликало досаду то, що він не міг спілкуватися з усіма трьома проектами з одного терміналу, і що три проекти не мали можливості спілкуватися групою для обміну інформацією.

1.2 Мобільний додаток

Мобільний додаток - програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Багато мобільних додатків встановлено на самому пристрої або можуть бути завантажені на нього з онлайн-магазинів додатків, таких як App Store, BlackBerry App World, Google Play, Imobile market, Windows Phone Store і інших, безкоштовно або за плату.

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

Мобільні додатки розробляються на базі операційних систем.

Мобільна операційна система - операційна система для смартфонів, планшетів, ПК або інших мобільних пристроїв. Хоча ноутбуки і можна віднести до мобільних пристроїв, однак операційні системи, як правило, використовуються на них, мобільними не рахуються, так як спочатку розроблялися для великих стаціонарних настільних комп'ютерів, які традиційно не мали спеціальних «мобільних» функцій, та й не потребували них. Ця різниця розмита в деяких нових операційних системах, що представляють гібрид того і іншого.

Мобільні операційні системи поєднують в собі функціональність ОС для ПК з функціями для мобільних і кишенькових пристроїв: сенсорний екран, стільниковий зв'язок, Bluetooth, Wi-Fi, GPS-навігація, камера, відеокамера, розпізнавання мови, диктофон, музичний плеєр, NFC і інфрачервоне дистанційне управління.

Портативні пристрої мобільного зв'язку (наприклад, смартфони) містять дві операційні системи. Основну програмну платформу взаємодії з користувачем доповнює друга, низькорівнева пропрієтарна операційна система реального часу, яка обслуговує радіоустаткування. Дослідження показали, що

такі низькорівневі операційні системи уразливі перед шкідливими базовими станціями, здатними отримати контроль над мобільним пристроєм.

Мобільні додатки розроблюються на базі таких операційних систем: Android, Lineage OS, Fire OS, Flyme OS, iOS, Windows 10 Mobile, Sailfish OS, Tizen, Remix OS.

Android – це операційна система, яка була створена спеціально для портативних електронних пристроїв, нетбуків, смартфонів і планшетів, на базі Linux. Її основні переваги – відкритість, простота і мінімум використовуваного обсягу пам'яті.

Android, яка отримує підтримку від найбільшої в світі пошукової системи, швидко стала найпопулярнішою мобільної ОС у світі. Її успіху багато в чому сприяла відкритість вихідного коду і політика Google, яка дозволяє виробникам безкоштовно встановлювати Android на свою продукцію. "Робота" воліють такі великі компанії, як HTC, Samsung, Sony і Motorola.

Android, встановлена навіть на просту "звонілку", перетворює її в повноцінний "розумний" телефон. З його допомогою можна виходити в Інтернет, дивитися викачані з Мережі фільми або відеоролики на YouTube, читати книги, спілкуватися з друзями в соціальних мережах, слухати музику, знімати на вбудовану камеру фотографії та відразу публікувати їх в Facebook, словом, робити майже все те ж саме, що і на комп'ютері.

Сьогодні в світі близько 70% всіх процесорних пристроїв використовують операційну систему Android, і це служить найбільш вагомим доказом її переваг.

Серед безперечних переваг Android користувачі найчастіше вказують:

- а) високу швидкодію системи;
- б) відмінна якість веб-серфінгу, швидкий доступ в інтернет;
- в) чудову інтеграцію з численними сервісами Google;
- г) великий магазин додатків на всі смаки і потреби;
- д) відсутність заборон на використання додатків від сторонніх розробників;

- е) відсутність ліцензій на ОС, завдяки чому користуватися нею можуть всі бажаючі;
- ж) відсутність плати за користування ОС і багатьма додатками.

Чим відрізняються між собою версії Android? Починаючи з 2007 року, коли була створена ОС Android, і до наших днів, вона постійно вдосконалюється. Практично щороку з'являються нові версії системи, які мають все більш широкими можливостями і надають більший комфорт користувачам. Більш старі версії призначені переважно для мобільних телефонів, але вже починаючи з 3-й версії Андроїд адаптується для планшетів.

Сучасні виробники мобільних і планшетних пристроїв відразу встановлюють на свої продукти сучасні версії ОС. Однак при покупці техніки попередніх років випуску необхідно звертати увагу на встановлену там версію Андроїда і, що не менш важливо, на можливість її поновлення в разі потреби.

Може здатися дивним, але ні диктофона, ні примітивних ігор, ні навіть простого калькулятора ви в смартфоні від Google не знайдете, але, на щастя, про них не забувають в своїх апаратах Samsung, Sony, HTC та інші виробники. А взагалі понад 700 тисяч інших програм від сторонніх розробників можна вибрати самостійно через інтернет-магазин Google Play. У ньому є як платні, так і безкоштовні програми.

Що краще Android або Windows? З моменту появи мобільних телефонів, що поєднують в собі функції власне телефону і портативного комп'ютера, не вщухають суперечки про те, яка ОС краще – Android або мобільна версія Windows. Спочатку, безумовно, пальму першості утримував Android, який з самого початку був прекрасно адаптований для роботи з портативними пристроями. Однак з появою Windows 8 і її мобільної версії ситуація змінилася, і тепер однозначно вказати лідера досить складно, але Android характеризується високою доступністю засобів розробки. Засоби розробки для платформи Android безкоштовні, в той час як розробка, наприклад, під iPhone (від компанії Apple) вимагає чималих початкових фінансових вкладень. Крім усього перерахованого вище, перевагою ОС Android є наявність безкоштовних

бібліотек для роботи зі сторонніми ресурсами (Google Map API, ін.), в той час як для Windows Phone Mobile такі бібліотеки не поширені. Обидві операційні системи чудово працюють, забезпечують високу швидкість і якісний доступ в інтернет, для них існує величезна кількість додатків, як ігрових, так і використовуваних для роботи, розваг і в інших цілях.

1.3 Актуальність використання онлайн-словника

У вік комп'ютерних технологій неможливо собі уявити вивчення іноземної мови без онлайн словника або перекладача. Паперові словники поступово відходять на другий план, тому що все, у кого є комп'ютер, планшет і телефон користуються додатками і програмами для перекладу слів, які мають ряд незаперечних переваг у порівнянні з традиційним паперовим словником. По-перше, онлайн словники дають всю необхідну інформацію за секунди, досить ввести потрібне слово і вираз в поле пошуку. По-друге, онлайн програми і словники мають у багато разів більше можливостей і функцій, ніж їх паперові аналоги. Електронні словники наводять максимальну кількість значень слова, пояснень, прикладів, на які в паперовому словнику просто немає місця. По-третє, онлайн словники інтерактивні. Наприклад, вони дозволяють не тільки побачити слово, а й прослухати, як слово вимовляється. У деяких словниках є можливість прослухати навіть приклади або почитати тлумачення пояснення користувачів. Крім того, онлайн словники не займають місця і їх можна завжди мати при собі. І останнє, що хочеться підкреслити - ви можете безкоштовно користуватися онлайн словниками. Переважна більшість онлайн словників англійської містять повну та вичерпну інформацію у вільному доступі, за яку не потрібно платити. Якщо ж вам потрібен спеціалізований словник, або просунуті функції, за це може вимагатися певна плата.

1.4 Порівняння з аналогічними системами

Мобільний словник - зручний додаток: Android-смартфон завжди при собі, а перекладати іноземні слова з української на англійську (або навпаки) доводиться досить часто. Якщо ви подорожуєте або пишете текст англійською (або іншою) мовою, словник для Android незамінний.

Всі сучасні онлайн-словники надають приблизно однаковий функціонал, відрізняються вони все тільки дизайном і кількістю наданих функцій, які є у додатку. Виходячи з вище перерахованого, всі інтернет-словники переслідують одні цілі для підвищення конкурентоспроможності, а саме підвищення якості реалізації самої системи і максимальну зручність користування сервісом, щоб охопити якомога більшу кількість користувачів.

Вимоги, які більшість користувачів очікує отримати при відкритті додатку є досить простий інтерфейс, який не буде складний для розуміння як ним користуватися, зручне переміщення по фрагментам словника, досить швидко сприйнята інформація про використання різноманітних функцій додатку: двосторонній переклад, підтримка рукописного введення тексту, вибіркова завантаження мовних розширень на Android в міру потреби, додавання слів в обране.

Більшість онлайн словників мають надмірну кількість функцій, в яких користувачеві нелегко розібратися. Багато вивчаючих іноземну мову, а надто початківців, якраз таки хочуть простоти та доступності, в основному такими є люди, які користуються онлайн-словником перший раз. Користувач такого додатку, швидше за все, вибере словник де буде відповідний їм потребам функціонал, адже вони не захочуть розбиратися в функціях для досвідчених користувачів.

Для того щоб максимально спростувати інтерфейс онлайн-додатку і вийти на конкурентоспроможний шлях з аналогічними інтернет-системами словників,

а також виходячи з бажань клієнтів, основною метою в створенні цього проекту робиться розробка зрозумілого і зручного в користуванні інтерфейсу, як можна більше інформації про слово і який простий в сприйнятті.

1.5 Технічне завдання

В контексті даної дипломної роботи повинно бути виконано:

- а) дослідження онлайн-додатків та процесу їх роботи;
- б) порівняння з аналогами;
- в) ознайомлення з існуючим інструментарієм в розробці Android-додатків;
- г) розробка дизайну системи;
- д) розробка системи адміністрування;
- е) розробка фрагментів(сторінок).

В результаті виконання даного дипломного проекту буде створений онлайн-словник для ОС Andoid. Додаток повинен бути зручний для користувача, простий в управлінні, інформативний. Повинен асоціюватися з турботою про того, хто вивчає іноземну мову. Додаток повинен функціонувати як в онлайн-режимі, так і оффлайн-режимі, а також надавати можливість користувачам мережі переглядати інформацію про вибране слово. Додаток повинен дозволяти додавати слово в меню обраного.

2 ОБГРУНТУВАННЯ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ СИСТЕМИ УПРАВЛІННЯ ПРОДАЖАМИ

2.1 Android-програмування

Android-програмування - це один з розділів програмування, який орієнтується на створення додатків, які забезпечують функціональність для різноманітних програм на операційній системі Android.

Існує десять мов програмування для Android які дозволяють створити мобільний додаток будь-якого типу і складності:

- а) Java. Не буде великим перебільшенням назвати Java офіційною мовою Android. У всякому разі, майже вся освітня документація, всі інтернет-курси засновані на цьому. А ще це найпопулярніший мову за оцінкою ТЮВЕ, другий за кількістю початкових кодів на GitHub, та й взагалі велика красива мова. Саме тому вивчення Java має бути першочерговим завданням для будь-якого Android-розробника;
- б) C#. При всьому нескінченному скепсисі, спрямованому в бік продуктів Microsoft, варто визнати, що C # цього не заслуговує. Це чудова мова, що увібрала в себе все краще від Java, при цьому врахувавши і виправивши багато недоліків. Що стосується розробки додатків під Android, то тут до послуг одні з найбільш функціональних середовищ Visual і Xamarin Studio;
- в) Python. Той факт, що Android не підтримує використання Python для створення нативних додатків, ще не означає, що це неможливо. Любителі цієї “змійної” мови розробили безліч інструментів, що дозволяють скомпілювати код на Python в потрібний стан. Найпопулярнішим фреймворком є Kivy, який без зусиль допоможе вам створити додаток для Play Market на чистому Python;
- г) HTML, CSS і JavaScript.. Стандартний мовної набір працівника інтернету: HTML, CSS і JavaScript. Не знаючи ці три мови, можливо розробляти

додатки досить вузької спрямованості. Працювати з HTML, CSS і JavaScript можна використовуючи середовища PhoneGap Build або, в більш спеціалізованому випадку, Adobe Cordova. Великих знань вони від вас не вимагатимуть, а результат забезпечать. Або ось з останнього, React Native від Facebook - це вже наступний рівень зручності взаємодії, але досвіду і документації накопичилося мало;

- д) Lua. Lua - мова, яка старше Java, куди менш популярна, але все одно затребувана. У неї є ряд переваг, на кшталт динамічної типізації, щодо простого синтаксису, але до наших днів вона дожила завдяки використанню в іграх. Саме зручність створення програмного проширення між движком і оболонкою відкрило перед Lua двері в світ кишенькових гаджетів. Corona SDK - середовище для розробки мобільних кроссплатформених додатків, переважно ігор, де головним інструментом є Lua. З 2015 року воно поширюється безкоштовно, розрахована на початківців розробників;
- е) C / C ++. Насправді Google надає розробникам два середовища розробки: SDK, призначеної для роботи з Java, і NDK, де нативними мовами є C / C ++. Так, звичайно неможливо написати цілий додаток з використанням лише цих мов, але з їх допомогою створюється бібліотека, яку згодом за допомогою Java можна підключити до основного тіла програми;
- ж) Kotlin. Сама по собі відмінна мова, а в зв'язці з Java вона ще краще. Дійсно, офіційно випущений лише два роки тому, Kotlin дуже швидко завойовує серця розробників по всьому світу практично повною відсутністю недоліків. За його допомогою (точніше за допомогою рідного середовища IntelliJ IDEA) ви не будете відчувати жодних проблем в розробці нативних додатків для Android.

Незважаючи на те, що переважна більшість розробників немає ніякого діла до NDK, проте задіявши цей інструмент можна отримати кращі результати по продуктивності і використанню внутрішніх ресурсів. А це саме те, що на Android відрізняє хорошу ідею додатку від оптимальної реалізації.

Засоби для розробки програмного забезпечення для Android (software development kit, або SDK) включають в себе засоби для розробки та відлагодження програм, бібліотеки, емулятор. Для розробки необхідно завантажити Android SDK для x86 платформ з Windows XP або Vista, Mac OS X 10.4.8 чи старшою, або Ubuntu Linux (Dapper Drake або старшу версію). Під іншими дистрибутивами Linux'a також повинно працювати, але на пряму, така можливість, не підтримується.

Хоча Android базується на ядрі Linux, він стоїть дещо осторонь Linux-спільноти та Linux-інфраструктури. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java.

2.2 Емуляція та Android Studio

Емуляція (англ. Emulation) в обчислювальній техніці - комплекс програмних, апаратних засобів або їх поєднання, призначене для копіювання (або емуляції) функцій однієї обчислювальної системи (гостя) на іншій, відмінній від першої, обчислювальної системі (хост) таким чином, щоб емулювати поведінку як можна ближче відповідала поведінці оригінальної системи (гостя). Метою є максимально точне відтворення поведінки на відміну від різних форм комп'ютерного моделювання, в яких імітується поведінка деякої абстрактної моделі. Наприклад, моделювання урагану або хімічної реакції не є емуляцією.

Як правило емулятор складається з декількох модулів, кожен з яких відповідає окремій емуліруемой підсистемі оригінального пристрою. У найбільш загальному випадку емулятор складається з наступних блоків:

- а) модуль емуляції або симуляції CPU (в даному випадку терміни св ааааапрактично рівнозначні);
- б) модуль емуляції підсистеми пам'яті;
- в) модулі емуляції різних пристроїв введення-виведення.

На рис. 2.1 представлено середовище розробки стандартний домашній екран Android Studio 3.0 з емулятором.

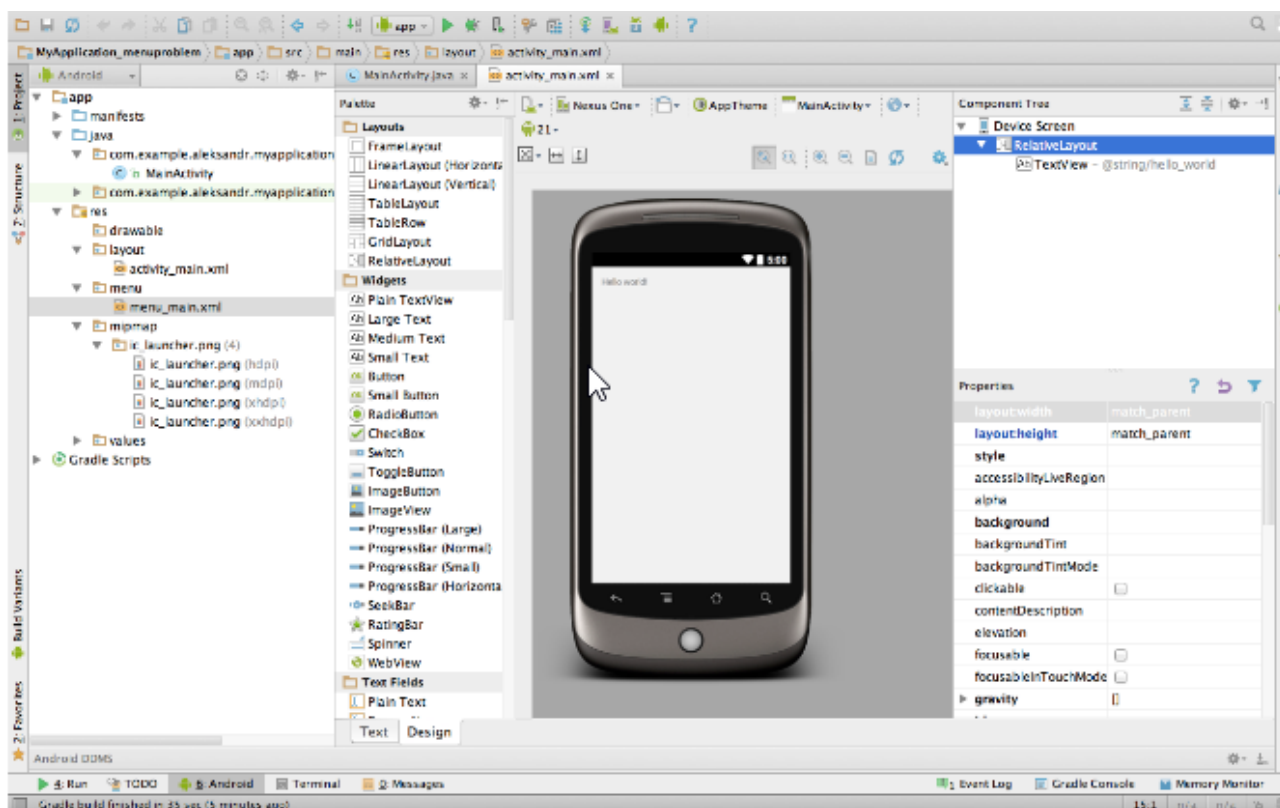


Рисунок 2.1 - Стандартний домашній екран Android Studio 3.0

Системні шини з метою спрощення емуляції і збільшення продуктивності, як правило, не емулюються. Замість цього віртуальна периферія звертається безпосередньо до процесору або підсистемі пам'яті.

Для того, щоб тестувати програми, знадобиться Android Virtual Device (AVD). Це емулятор Android-смартфона, на який можна встановлювати створені нами додатки, і запускати їх там.

2.3 Мова Java

Як мова розробки в дипломній роботі використовується об'єктно-орієнтована мова програмування *Java*. Це мова програмування, розроблена компанією Sun Microsystems. Програми Java зазвичай компілюються в спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній Java-машині (JVM) незалежно від комп'ютерної архітектури. Дата офіційного випуску - 23 травня 1995 року. Гідність подібного способу виконання програм - в повній незалежності байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання. Часто до недоліків концепції віртуальної машини відносять те, що виконання байт-коду віртуальною машиною може знижувати продуктивність програм і алгоритмів, реалізованих на мові Java. Останнім часом було внесено ряд удосконалень, які дещо збільшили швидкість виконання програм на Java: застосування технології трансляції байт-коду в машинний код безпосередньо під час роботи програми (JIT-технологія) з можливістю збереження версій класу в машинному коді, широке використання переносних орієнтованого коду (native-код) в стандартних бібліотеках, апаратні засоби, що забезпечують прискорену обробку байт-коду (наприклад, технологія Jazelle, підтримувана деякими процесорами фірми ARM). За даними сайту shootout.alioth.debian.org, для семи різних завдань час виконання на Java становить в середньому в півтора-два рази більше, ніж для C / C ++, в деяких випадках Java швидше, а в окремих випадках в 7 разів повільніше. З іншого боку, для більшості з них споживання пам'яті Java-машиною було в 10-30 разів більше, ніж програмою на C / C ++. Також примітно дослідження, проведене компанією Google, згідно з яким відзначається істотно нижча продуктивність і більше споживання пам'яті в

тестових прикладах на Java в порівнянні з аналогічними програмами на C. Ідеї, закладені в концепцію і різні реалізації середовища віртуальної машини Java, надихнули безліч ентузіастів на розширення переліку мов, які могли б бути використані для створення програм, що виконуються на віртуальній машині. Ці ідеї знайшли також вираз в специфікації загальної інфраструктури CLI, закладеної в основу платформи .NET компанією Microsoft. Останнім релізом є версія 1.6, в якій було вироблено поліпшення системи безпеки, поліпшення підтримки XML, а також додана підтримка скриптової мови JavaScript на основі механізму Mozilla Rhino, поліпшена інтеграція з робочим столом, додані деякі нові можливості в створенні графічних інтерфейсів.

Java-програма може бути визначена як сукупність об'єктів, які взаємодіють за допомогою виклику методів один одного:

- а) об'єкт. Об'єкти мають стан і поведінку. Наприклад: собака може мати стан колір, ім'я, а також і поведінка - кивати, бігти, лаїть, їсти. Об'єкт є екземпляром класу;
- б) клас. Може бути визначений як шаблон, який описує поведінку об'єкта;
- в) метод. Метод є в основному поведінкою. Клас може містити кілька методів. Саме в методах логічно записані дані маніпулюють і виконують всі дії;
- г) змінні примірника. Кожен об'єкт має свій унікальний набір змінних екземпляра. Стан об'єкта створюється значеннями, присвоєними цим змінним примірника.

Особливості синтаксису Java:

- а) чутливість до регістру. Java чутливий до регістру, тобто ідентифікатор "Hello" та "hello" мають різний зміст;
- б) назва класів - для всіх перша буква повинна бути в верхньому регістрі;
- в) якщо кілька слів використовуються, щоб сформувати назву класу, перша буква кожного внутрішнього слова повинна бути в верхньому регістрі, наприклад, «MyJavaClass»;

- г) назва методів - в синтаксисі Java все імена методів повинні починатися з літери нижнього регістру;
- д) якщо кілька слів використовуються, щоб сформувавши ім'я методу, то перша буква кожного внутрішнього слова повинна бути в верхньому регістрі, наприклад, «public void myMethodName ()»;
- е) назва файлу програми - найменування файлу програми має точно збігатися з ім'ям класу;
- ж) при збереженні файлу, Ви повинні зберегти його, використовуючи ім'я класу (пам'ятаєте про чутливість до регістру) і додати «.java» в кінці імені (якщо імена не співпадають, Ваша програма не компілюватися), наприклад, «MyJavaProgram» - це назва класу, тоді файл повинен бути збережений як «MyJavaProgram.java»;
- з) `public static void main (String args [])` - обробка програми починається з методу `main ()`, який є обов'язковою частиною кожної програми (рис. 2.2).

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

Рисунок 2.2 - Проста програма на мові Java.

2.4 Реляційна система управління базами даних SQLite

База даних - це структурована сукупність даних. Ці дані можуть бути будь-якого типу - це може бути як простий список майбутніх покупок, так і перелік експонатів музею, або список великої кількості інформації якоїсь корпорації. SQLite - компактна вбудована база даних. Вихідний код бібліотеки переданий в суспільне надбання. У 2005 році проект отримав нагороду Google-O'Reilly Open Source Awards.

Слово «вбудована» означає, що SQLite не використовує парадигму клієнт-сервер, тобто движок SQLite не є окремо працюючим процесом, з яким взаємодіє програма, а являє собою бібліотеку, з якої програма компонується, і движок стає складовою частиною програми. Таким чином, в якості протоколу обміну використовуються виклики функцій бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується програма. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції записи весь файл, який зберігає базу даних, блокується. ACID-функції досягаються в тому числі за рахунок створення файлу журналу.

Кілька процесів або потоків можуть одночасно без будь-яких проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, якщо ніяких інших запитів в даний момент не обслуговується; в іншому випадку спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

У комплекті поставки йде також функціональна клієнтська частина у вигляді виконуваного файлу `sqlite3`, за допомогою якого демонструється реалізація функцій основної бібліотеки. Клієнтська частина є кроссплатформенної утилітою командного рядка.

Завдяки архітектурі движка можливо використовувати SQLite як на вбудовуваних системах, так і на виділених машинах з гігабайтними масивами даних.

Для управління базами даних SQLite існує програмне рішення під назвою SQLiteStudio. Це менеджер SQLite баз даних, який дозволяє переглядати і редагувати SQLite 3, SQLite 2 та SQLCipher бази даних вбудованих системах, так і на виділених машинах з гігабайтними масивами даних (рис. 2.3).

Особливості:

- а) портативний - немає необхідності установки або видалення. Досить
- б) завантажити, розпакувати і запустити;
- в) інтуїтивно зрозумілий інтерфейс;
- г) потужний, але легкий і швидкий інструмент;
- д) всі функції SQLite3 і SQLite2 помістилися в простому графічному інтерфейсі;
- е) крос-платформний - працює на Windows 9x / 2k / XP / 2003 / Vista / 7, Linux, MacOS X і повинен працювати на інших Unix системах;
експорт в різні формати (SQL-оператори, CSV, HTML, XML, PDF, JSON);
імпорт даних з різних форматів;
- ж) численні невеликі доповнення, такі як форматування коду, історія запитів, які виконуються у вікнах редактора, перевірка синтаксису на льоту, і багато іншого;
- з) підтримка Unicode;
- и) настроюються кольору, шрифти і значки;
- і) відкритий вихідний код, який опублікований під ліцензією GPLv3.

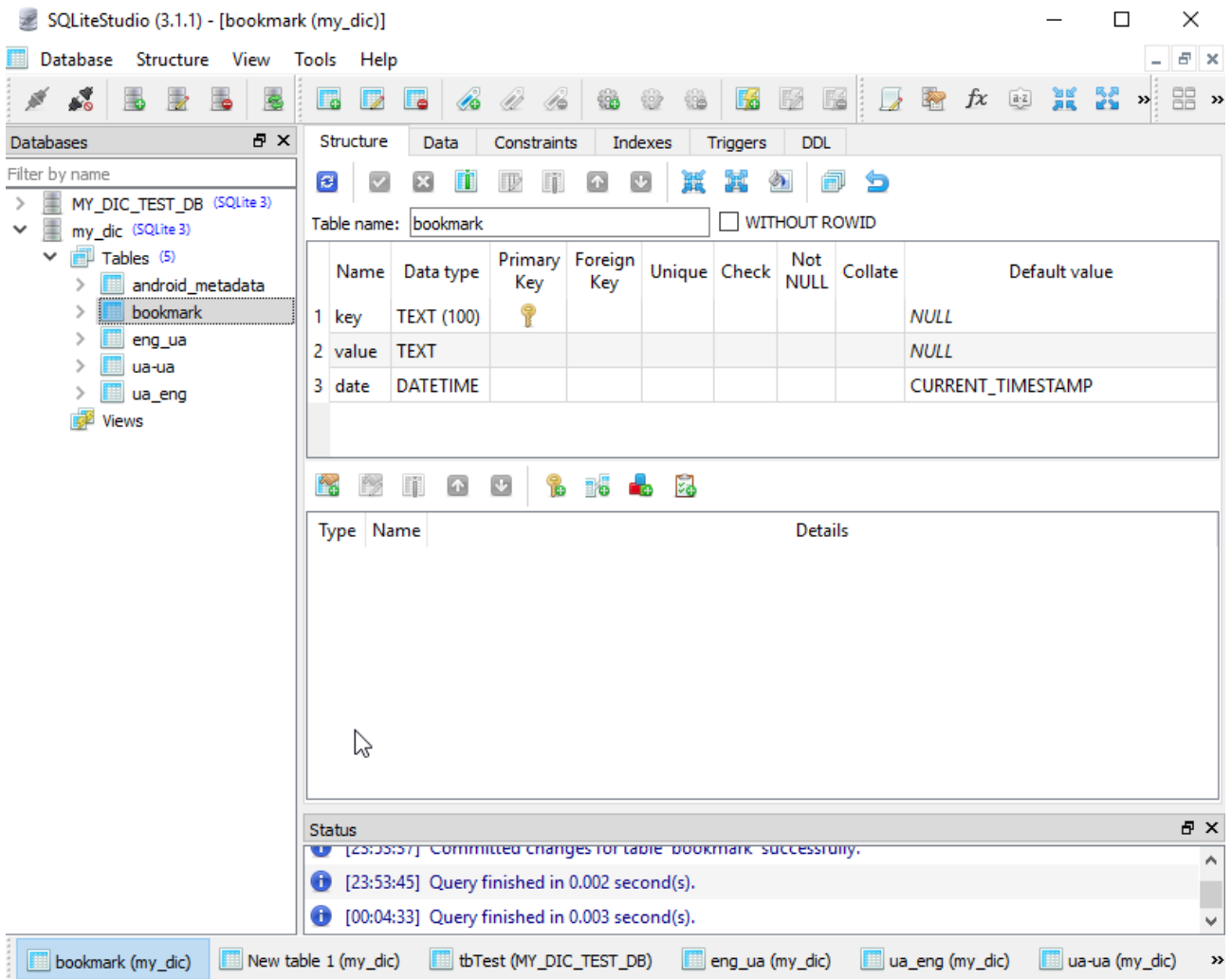


Рисунок 2.3 - Середовище SQLiteStudio.

3 ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-СЛОВНИК ДЛЯ ОС ANDROID

3.1 Проектування структури системи

Розробка структури системи онлайн-словник для ОС Android проводилася з обліком наступних вимог:

- а) спроектувати головну сторінку, яка буде проста у використанні і відвідавши яку, користувач відразу зрозумів що це онлайн-словник;
- б) проектувати зрозуміле і зручне меню навігації;
- в) розробити систему відображення контенту;
- г) спроектувати внутрішню сторінку обраного користувачем слова;
- д) розробити сторінку на якій будуть слова з обраного.

Проаналізувавши і узагальнивши ці дані, була спроектована структура для додатку Android (рис. 3.1).



Р

Рисунок 3.1 - Загальна схема онлайн-словника.

3.2 Розробка дизайну системи

Не секрет, що число пристроїв на Android велике, вони розрізняються по залізу, розмірами і якістю екрану, потужності процесора та інше. На відміну від iPhone-програмістів, які знають напевно на якому пристрої буде запущено їх додаток, Android-розробникам необхідно приділяти увагу сумісності додатків з різними пристроями.

Починаючи з версії Android 1.6, в маніфест був доданий тег `<support-screens>`, який використовується для визначення класу пристроїв, на яких може бути запущено додаток. Атрибути тега `smallScreens`, `normalScreens`, `largeScreens`, `xlargeScreens` відповідають певним вище значень екрану і можуть приймати значення `true` або `false`. Дефолтні значення атрибутів варіюються в залежності від використовуваної версії Android (більш детальну інформацію можна подивитися тут.). При визначенні значення атрибута як `true`, ОС Android отримує сигнал про те, що додаток сумісно з відповідним типом екрану і не застосовує додаткові кошти для сумісності (що відбувається при значенні `false`). Варто також відзначити, що ці кошти (функції) працюють тільки на сумісність з великими розмірами екранів (таким чином якщо значення `normalScreen` - `true`, інші - `false`, додаток буде також сумісно з екранами `large` і з `xlarge`, але не сумісно зі `small`) . Даний тег також використовується Android Market'ом для фільтрації додатків.

Для щільності також є атрибут - `anyDensity`, який також приймає значення `true` / `false`. Якщо значення атрибута - `true`, ОС Android не використовує функції для сумісності з різними густиною екрану. У цьому випадку програма має використовувати `dp` для промальовування UI елементів, або самостійно управляти обчисленням розмірів для різної щільності. Якщо значення - `false`, OS Android включає функції для масштабування елементів відповідно до щільністю екрану.

ОС Android також надає засоби для визначення ресурсів, які будуть використані для конкретних розмірів екранів і щільності. Ресурси розміщуються у відповідних папках:

Існують загальні рекомендації для створення сумісного додатка:

- використовувати значення `wrap_content`, `fill_parent`, `dp` в макетах;
- уникати використання `RelativeLayout`;
- використовувати методи класу `ViewConfiguration` для отримання стандартних значень розмірів, швидкості, часу;
- використовувати різні ресурси для різних значень щільності і розмірів екрану.

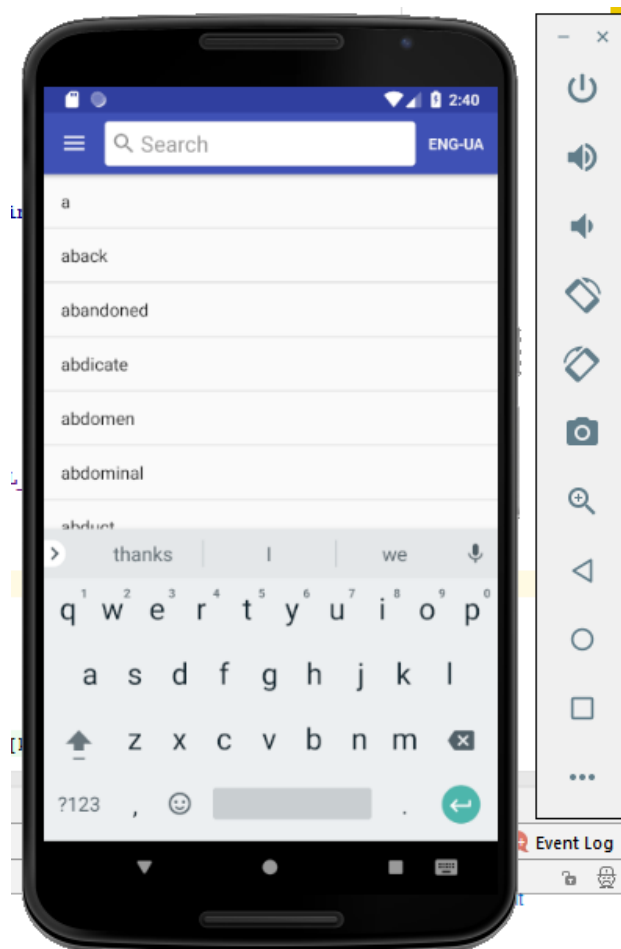


Рисунок 3.2 – Зображення додатку на екрані смартфона

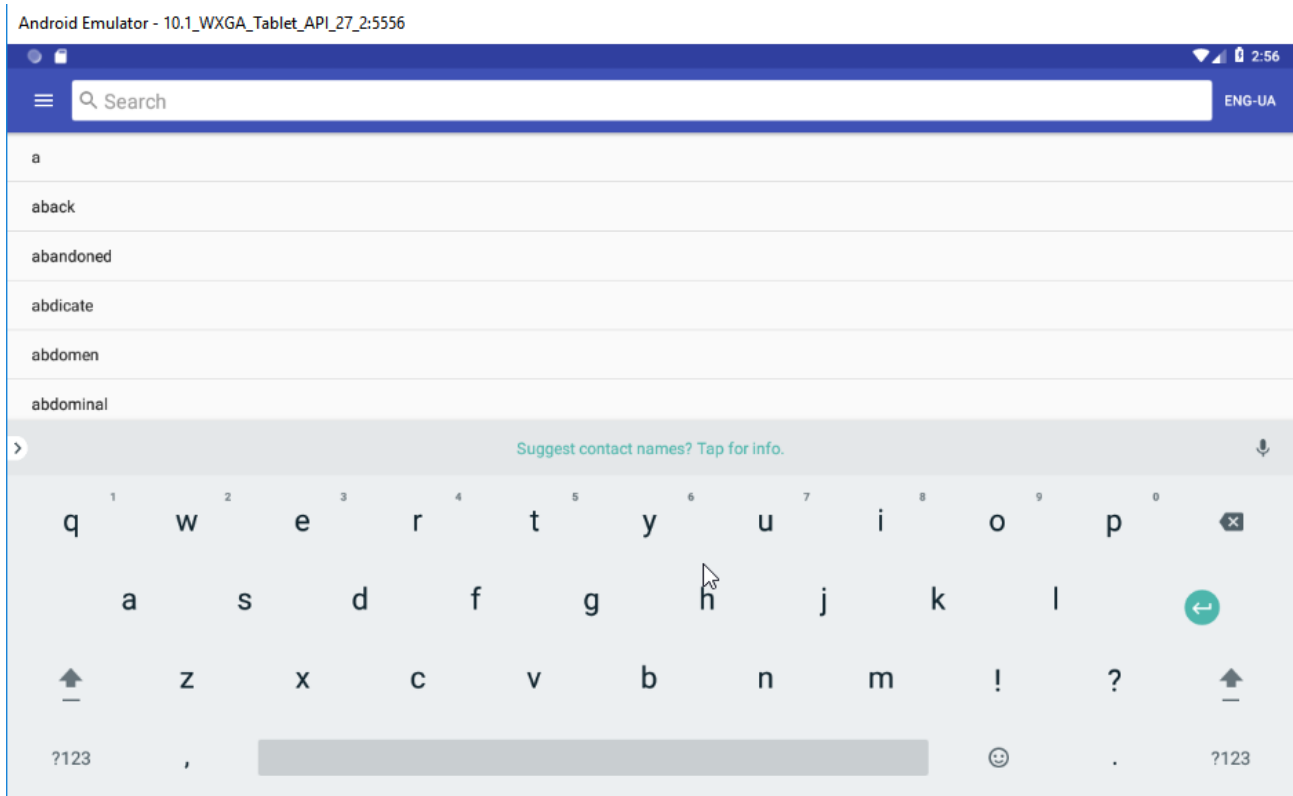


Рисунок 3.3 – Зображення додатку на екрані планшета

3.3 Структура бази даних

Основним джерелом інформації з якого система виводить інформацію є база даних. Оскільки функціональна частина системи була написана за допомогою мови програмування Java, то було прийнято рішення взяти на озброєння саме СУБД SQLite, це поєднання на даний момент є найпоширенішим і найбільш практичним.

Попутно з реалізацією функціоналу додатку проектувалася і база даних. Вона складається з декількох таблиць, які впорядковано зберігають інформацію. Для кожного розділу інформації зроблена своя таблиця. Таблиці мовних пар

зберігаються в різних таблицях, все це зроблено для того, щоб управляти великим обсягом даних. На рисунку 3.4 зображена загальна структура системи бази даних, також на цьому зображенні типи даних, які зберігаються в базі, їх обмеження.

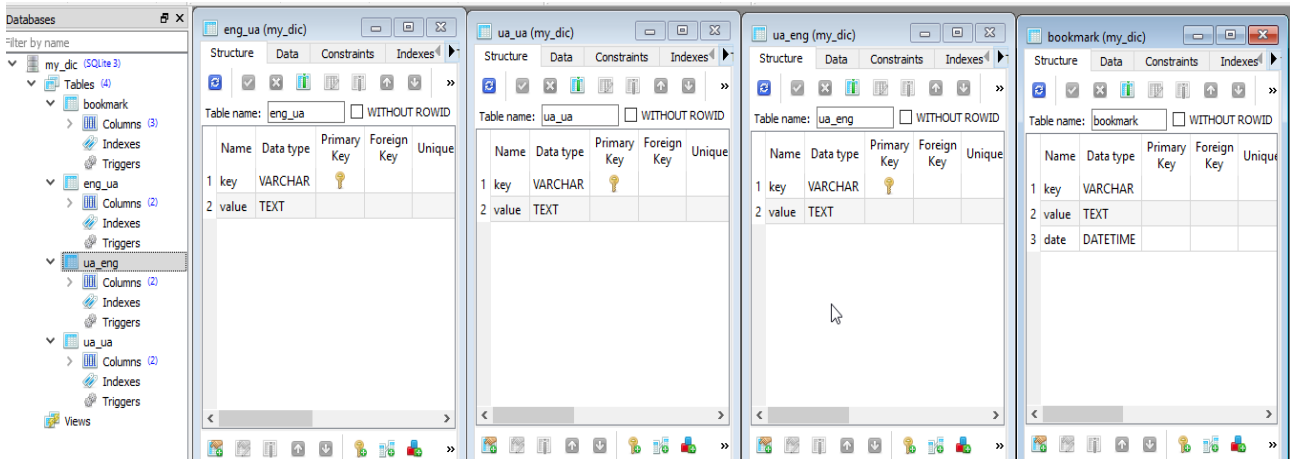


Рисунок 3.4 - Загальна структурна схема бази даних

3.4 Головна сторінка

Головна сторінка додатку містить в собі поле пошуку, віртуальну клавіатуру, основне навігаційне меню, а також основну область додатку для того, щоб відвідувач з першої сторінки міг зрозуміти, що це саме словник англо-український словник. При вході в додаток відвідувач опиняється на головній сторінці, яка виглядає як на рис. 3.5.

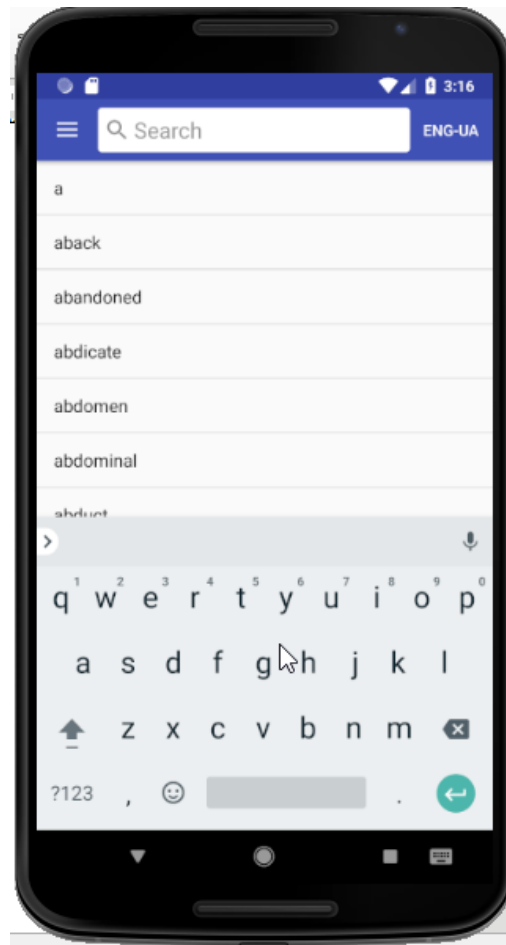


Рисунок 3.5 – Головна сторінка додатку

На даній сторінці розташований основний контент додатку. Тут відображені слова в алфавітному порядку починаючи з літери “а”. Також на головній сторінці знаходиться навігаційне меню, натиснувши на посилання з цього меню, відбувається перехід на інші сторінки сайту, в меню є такі посилання:

- а) «Bookmark» - це посилання створене для переходу до слів, які користувач додав в обране;
- б) «Rate» - натиснувши це посилання, користувач може віддати свій голос на користь додатку;

- в) «Share» - це посилання веде на сторінку на якій зображені ярлики відповідних ресурсів, на яких користувач може розмістити своє посилання;
- г) «Help» - сторінка з інструкцією користування та інформацією про зворотній зв'язок;
- д) «About» - це посилання перенаправляє користувача на сторінку в якій знаходиться опис додатку та його творців;

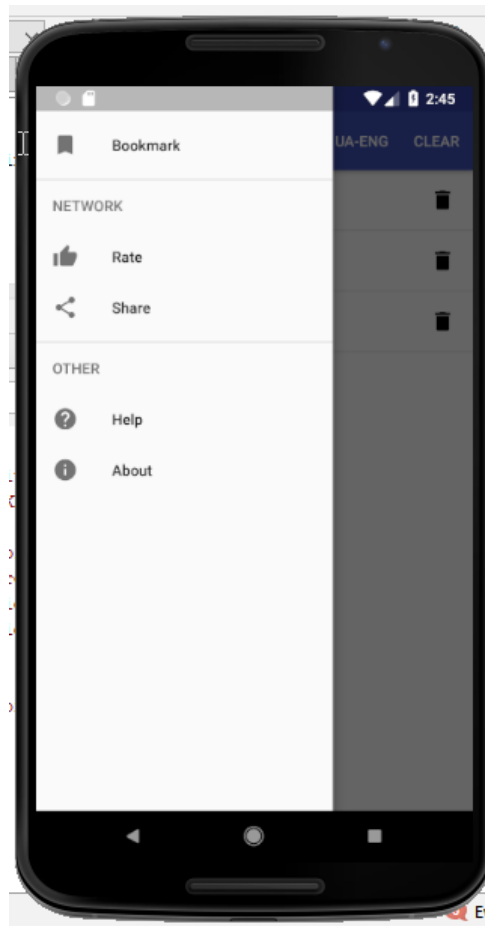


Рисунок 3.6 – Головна сторінка з навігаційним меню.

3.5 Пошук

Робота зі словником неможлива без функції пошуку потрібного слова. Ця опція представлена в онлайн-словнику для кожної мовної пари. Щоб нею скористатися, потрібно лише обрати відповідну мовну пару та почати вводити слово у поле пошуку (рис. 3.6).

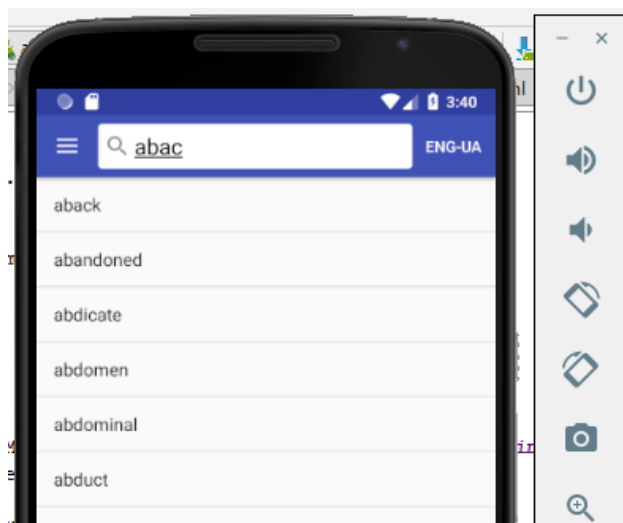
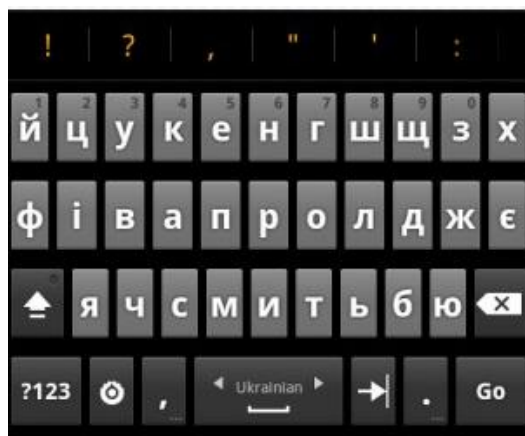


Рисунок 3.6 – Процес роботи пошуку.

Для того, щоб пошук працював належним чином для мовних пар “UA-ENG” та ”UA-UA”, потрібно завантажити на свій пристрій відповідну клавіатуру та вибрати українську мову. Українська мовна розкладка буде виглядати приблизно як на рис. 3.7.



Рисунку 3.7 — Українська клавіатура для пошуку.

3.6 Розділи словника

У додатку розроблена система вибору мовних пар. Користувач онлайн-словника має можливість обрати одну з трьох мовних пар: англо-українську ("ENG-UA"), українсько-англійську ("UA-ENG"), українсько-українську ("UA-UA").

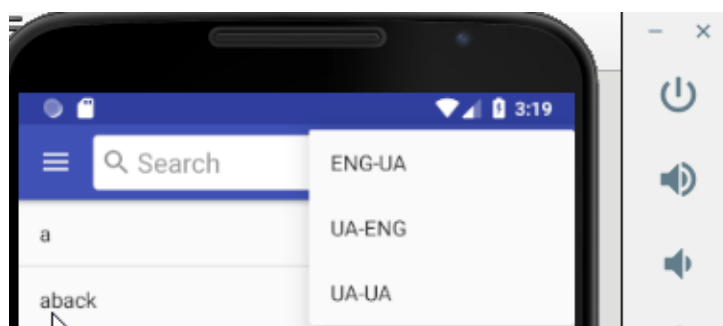


Рисунок 3.7 – Меню вибору мовної пари

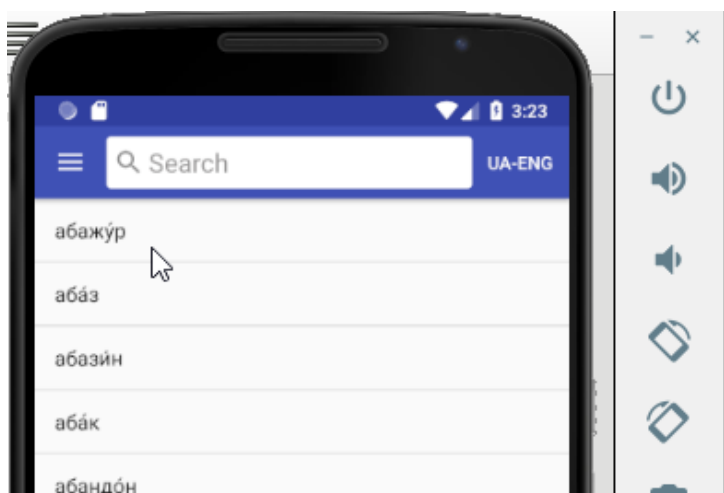


Рисунок 3.8 – Контент мовної пари UA - ENG

Після того, як користувач натискає на потрібне слово, він потрапляє саме на внутрішню сторінку з його перекладом, де він має можливість прочитати транскрипцію, ознайомитись з його значенням, прослухати слово або додати його в обране (рис. 3.9 та рис. 3.10).

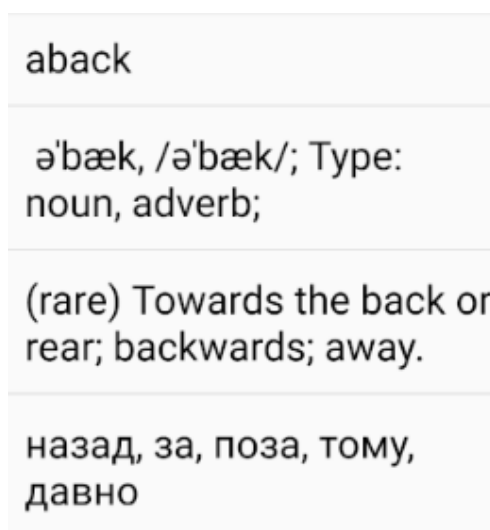


Рисунок 3.9 — Інформація про слово та його переклад



Рисунок 3.10 — Прослуховування слова та додавання його в обране

3.7 Навігаційне меню

Меню є стандартним компонентом інтерфейсу користувача в додатках багатьох типів. Для забезпечення звичної та однакової технології роботи з додатком слід представляти дії користувача і інші варіанти вибору в своїх операціях за допомогою API-інтерфейсів класу.

У меню додатку відображається вертикальний список пунктів, який прив'язаний до подання, який викликав меню. Він добре підходить для надання можливості додаткових варіантів дій, що відносяться до певного контенту або для видачі варіантів для другої частини команди. Деякі дії у меню не стосуються безпосередньо відповідного контенту.. Спливаюче меню призначене для розширених дій, що відносяться до областей контенту в операції (рис. 3.9).

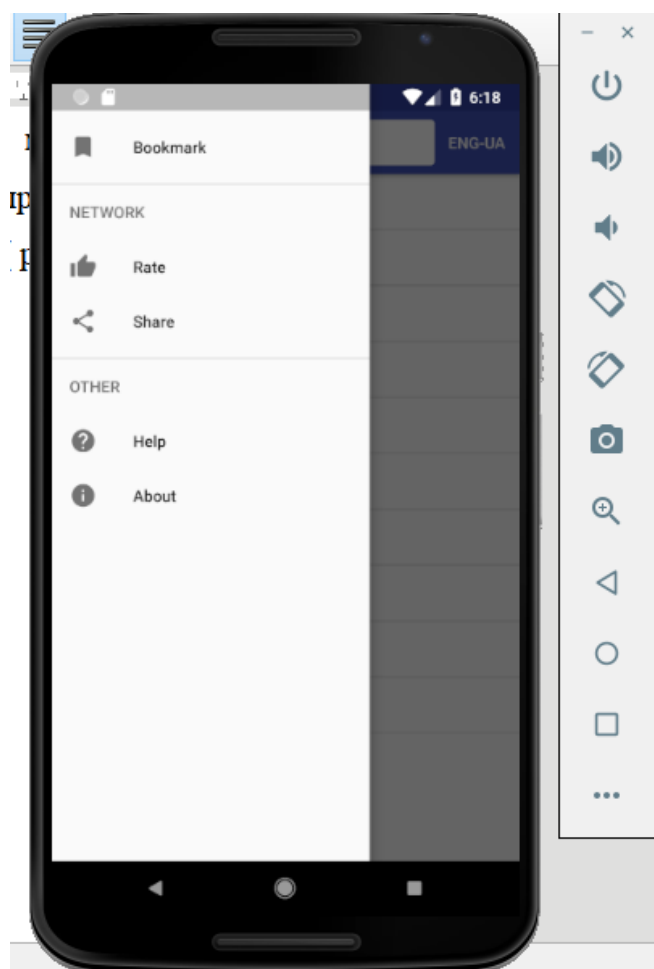


Рисунок 3.9 - Навігаційне меню додатку онлайн-словник.

Вивчаючи іноземну мову, користувачеві часто потрібно повертатися до слова, щоб запам'ятати його значення надовго. Саме для цих цілей і призначений розділ “Bookmark” Натиснувши на значок закладки у внутрішньому описі слова, користувач заносить його в розділ “Bookmark”, при цьому закладка змінює колір на червоний (рис. 3.10).



Рисунок 3.10 — Слово знаходиться в обраному.

Всі слова, занесені в “Bookmark” розташовуються не в алфавітному порядку, а додаються зверху у список по мірі надходження (рис 3.11).

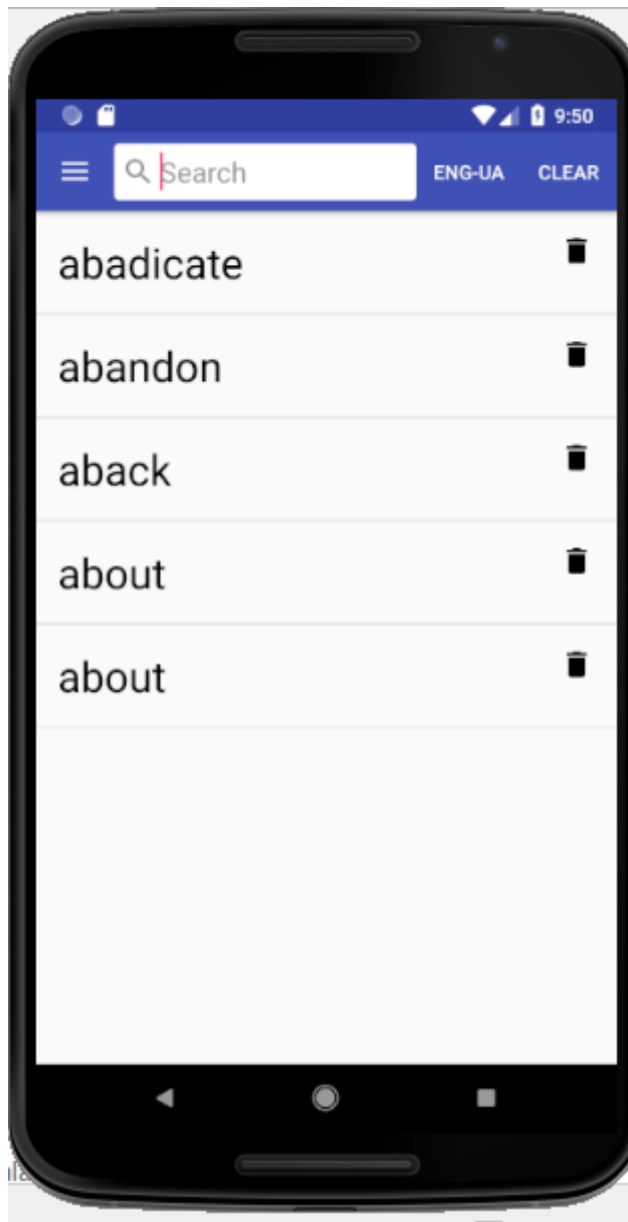


Рисунок 3.11 — Екран меню “Bookmark”

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи бакалавра було спроектувати систему управління продажами комп'ютерних компонент в мережі Інтернет, і як результат було створено повноцінний проект, котрий можна використовувати у розробці такої системи. За цим проектом в подальшому розроблятиметься реальна система, яка значно полегшить процес продажу комп'ютерних компонент у мережі Інтернет. Так як в процесі проектування використовувалося спеціалізоване приміщення, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде розроблятися розроблена система.

4.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика,

електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів шиї, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці. Забезпечення цих умов покладається на власника або уповноважений ним орган (далі роботодавець). Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Роботодавець повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизмові, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників. Він не має права вимагати від працівника виконання роботи, поєднаної з явною небезпекою для життя, а також в умовах, що не відповідають законодавству про охорону праці. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація, небезпечна для його життя чи здоров'я або людей, які його оточують, і навколишнього середовища.

4.1.1 Правові та організаційні основи охорони праці

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

Державна політика в галузі охорони праці визначається відповідно до

Конституції України Верховною Радою України і спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Відповідно до статті 3 Закону України «Про охорону праці» (далі – Закону) законодавство про охорону праці складається з Закону, Кодексу законів про працю України, Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності" та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ, директиви Європейської Ради).

4.1.2 Організаційно-технічні заходи з безпеки праці

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [2].

Також впроваджені організаційні заходи з пожежної безпеки - навчання і перевірку знань відповідно до вимог Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 29.09.2003 N 368, зареєстрованого в Міністерстві юстиції України 11.12.2003 за N 1148/8469 [3].

4.2 Аналіз стану умов праці

Робота над створенням системи управління продажами комп'ютерних компонент в мережі Інтернет проходитиме в приміщенні приватного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	5
Висота, м	3
Площа, м ²	25
Об'єм, м ³	75

Згідно з [4] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [5] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 □ 800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 □ 500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 □ 800

У кабінеті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом.

4.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00.-1.28-10 «Правил охорони праці під час експлуатації електронно-обчислювальних машин», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга $U=+220\text{В} \pm 5\%$;
- робочий струм $I=2\text{А}$;
- споживана потужність $P=350\text{ Вт}$.

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
1	2	3	4
фізичні			
- підвищена температура поверхонь обладнання	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[10]
- підвищений рівень шуму на робочому місці	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[14]
- підвищений рівень вібрації	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[14] [15]
- підвищена або знижена вологість повітря	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[10]
- підвищена або знижена рухливість повітря	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	1	[10]
- підвищений рівень електромагнітного випромінення	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[16]
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	4	[18]
- підвищена напруженість електричного поля	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[16]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[13]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[13]
- підвищена яскравість світла	порушення умов праці (організації місця праці-налагодження моніторів)	1	[17]

Продовження таблиці 4.3

хімічні:			
- загазованість повітря робочої зони, яка впливає на організм людини через органи дихання та надає токсичну і канцерогенну дію	від експлуатації сканерів, принтерів для роботи – O ₃ , оплавлення електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів, транзисторів й інше в ЕОМ та системах кондиціонування повітря - CO, CO ₂ , SO ₂ , P ₂ O ₅ , H ₂ S, HCl, H, NH ₃ , ClF ₃ , F ₂ O ₂ , F ₂ O ₃ , SeO ₂ , SeF ₆ , TeF ₆ , COCl ₂ , SO ₂ F ₂ , інш.	3	[14] [12]
психофізіологічні:			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[11]
- фізичні (статичне –	порушення умов праці робочого часу - безперервна робота)	2	[11]

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [11]. За умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, забруднення повітря шкідливими речовинами (джерелом, яких можуть бути: принтер, сканер та інші джерела виділення багатьох хімічних речовин - напр., озону, оксидів азоту та аерозолів високодисперсних частинок тонера), шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

4.3.2 Пожежна безпека

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування. Небезпека загоряння пов'язана з особливістю комп'ютерів - із значною кількістю щільно розташованих на монтажній платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Надійна робота окремих елементів і мікросхем в цілому забезпечується тільки в певних інтервалах температури, вологості і при заданих електричних параметрах. При відхиленні реальних умов експлуатації від розрахункових можуть виникнути пожежонебезпечні ситуації.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важко горючі) не надається технічно можливим. Тому проектом передбачаються способи і засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання, таких як: застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонам відповідно до ПУЕ;

Згідно [6] таке приміщення, площею 25 м², відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ

(правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три-провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт Ia. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [4].

4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Основний потік природного світла при цій повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працівника на ПЕОМ.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IVв розряду точності (середня точність). При цьому нормована освітленість на робочому місці (E_n) рівна 200 лк. Джерелом природного освітлення є сонячне світло.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає [6]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше $1/8$, в побутових – $1/10$:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10} \right) \times S_n \quad (4.1)$$

де S_b – площа віконних прорізів,

m^2 ; S_n – площа підлоги, m^2 .

$$S_n = a \cdot b = 5 \cdot 5 = 25 \text{ м}^2,$$

$$S = 1/8 \cdot 25 = 3,125 \text{ м}^2.$$

Приймаємо 2 вікна площею $S=1,6 \text{ м}^2$ кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \times S \times Z \times K}{F \times U \times M} \quad (4.2)$$

де E – нормована освітленість робочої поверхні, визначається

нормами – 300 лк; S – освітлювана площа, m^2 ; $S = 25 \text{ м}^2$;

Z – поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2; F – світловий потік

лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \cdot 25 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 2,0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.4.3 Шум та вібрація, електромагнітне випромінювання

Рівень шуму, що супроводжує роботу користувачів персональних комп'ютерів (зумовлений як роботою системних блоків, клавіатури, так і друкуванням на принтерах, а також зовнішніми чинниками), коливається у межах 50–65 дБА [7]. Шум такої інтенсивності на тлі високого ступеня напруженості праці негативно впливає на функціональний стан користувачів. Тому на практиці рекомендують знижувати фактичний рівень шуму у приміщеннях, де створюють комп'ютерні програми, виконують теоретичні та творчі роботи, проводять навчання до 40 дБА, а в приміщеннях, де виконують роботу, що потребує зосередженості, — до 55 дБА. У залах опрацювання інформації та комп'ютерного набору рівні шуму не повинні перевищувати 65 дБА.

4.4.4 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення $> 40 \text{ м}^3$ на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря;
- зниження рівня шуму та вібрації.

Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;

- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;
- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;
- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

Вимоги безпеки при надзвичайних ситуаціях:

- а) при раптовому припиненні подачі електричної енергії вимкнути всі пристрої ПК в такій послідовності: периферійні пристрої, ВДТ, системний блок, стабілізатор (або блок безперервного живлення). Витягнути вилки з розеток. При наявності ознак горіння (дим, запах горілого) необхідно вимкнути всі пристрої ПК, знайти місце загоряння і виконати всі можливі заходи для його ліквідації, попередивши терміново про це керівництво. У випадку виникнення пожежі негайно попередити про це пожежну частину та керівництво, виконати усі можливі заходи по евакуації людей з приміщення і розпочати гасіння пожежі первинними засобами пожежогасіння;
- б) при замиканні, перевантаженні електричного струму на електричному обладнанні, внаслідок ураження грозової блискавки та ймовірної небезпеки ураженням електричним струмом, приймають наступне:
 - 1) попередження замикання здійснюється правильним вибором, монтажем експлуатації мереж;
 - 2) застосування захисту схем у вигляді швидкодіючих реле, а також вимикачів, плавких запобіжників, автоматичних вимикачів.

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [8], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Під час роботивикористовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контура заземлення повинен мати не більше 4 Ом.

Розрахунок проводять за допомогою методу коефіцієнта використання (екранування) електродів. Коефіцієнт використання групового заземлювача η – це відношення діючої провідності цього заземлювача до найбільш можливої його провідності за нескінченно великих відстаней між його електродами.

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_d * R_{пр.з.}}{R_{пр.з.} - R_d} \quad (4.3)$$

де $R_{пр.з.}$ – опір природних заземлювачів; R_d – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_d$.

Підставивши числові значення у формулу (4.3), отримуємо:

$$R_{шт.з.} = \frac{4 * 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м. Приблизне значення питомого опору глини приймаємо $\rho = 40$ Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту, $\rho_{розр.}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в.}$, і горизонтальних $\rho_{розр.г.}$, Ом·м за формулою:

$$\rho_{розр.} = \Psi * \rho \quad (4.4)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів і кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{\text{розр.в}} = 1,7$ і горизонтальних $\rho_{\text{розр.г}} = 5,5$ Ом·м.

$$\rho_{\text{розр.в}} = 1,7 * 40 = 68 \text{ Ом} \cdot \text{м}$$

$$\rho_{\text{розр.г}} = 5,5 * 40 = 220 \text{ Ом} \cdot \text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача $R_{\text{в}}$, Ом, за (4.5).

$$R_{\text{в}} = \frac{\rho_{\text{розр.в}}}{2 * \pi * l_{\text{в}}} * \left(\ln \frac{2 * l_{\text{в}}}{d_{\text{ст}}} + \frac{1}{2} * \ln \frac{4 * t + l_{\text{в}}}{4 * t - l_{\text{в}}} \right) \quad (4.5)$$

де $l_{\text{в}}$ – довжина вертикального заземлювача (для труб - 2–3 м; $l_{\text{в}}=3$ м); $d_{\text{ст}}$ – діаметр стержня (для труб - 0,03–0,05 м; $d_{\text{ст}}=0,05$ м);

t – відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.6):

$$t = h_{\text{в}} + \frac{l_{\text{в}}}{2} \quad (4.6)$$

де $h_{\text{в}}$ – глибина закладання вертикальних заземлювачів (0,8 м); тоді t 0,8м

$$R_{\text{в}} = \frac{68}{2 * \pi * 3} * \left(\ln \frac{2 * 3}{0,05} + \frac{1}{2} * \ln \frac{4 * 2,3 + 3}{4 * 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання $\eta_{\text{в}}$:

$$n = \frac{2 * R_{\text{г}}}{R_{\text{д}}} = \frac{2 * 18,5}{4} = 9,25 \quad (4.7)$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_{\text{в}} = 0,57$ (табличне значення).

б) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання $\eta_{\text{в}}$, шт:

$$n_{\text{г}} = \frac{2 * R_{\text{г}}}{R_{\text{д}} * \eta_{\text{в}}} = \frac{2 * 18,5}{4 * 0,57} = 16,2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача $l_{\text{с}}$, м:

$$l_{\text{с}} = 1,05 * L_{\text{в}} * (n_{\text{г}} - 1) \quad (4.9)$$

де $L_{\text{в}}$ – відстань між вертикальними заземлювачами, (прийняти за $L_{\text{в}} = 3\text{м}$); $n_{\text{г}}$ – необхідна кількість вертикальних заземлювачів.

$$l_{\text{с}} = 1,05 * 3 * (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) $R_{\text{г}}$, Ом:

$$R_{\text{г}} = \frac{\rho_{\text{розр.г}}}{2 * \pi * l_{\text{с}}} * \ln \frac{2 * l_{\text{с}}}{d_{\text{см}} * h} \quad (4.10)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15 \text{ м}$;

h – глибина закладання горизонтальних заземлювачів (0,5 м);

$l_{\text{с}}$ – довжина з'єднувальної стрічки горизонтального заземлювача $l_{\text{с}}$, м

$$R_{\text{г}} = \frac{220}{2 * \pi * 48} * \ln \frac{2 * 48}{0,95 * 0,15 * 0,5} = 8,1 \text{ Ом}$$

- 9) Визначається коефіцієнт використання горизонтального заземлювача η_c відповідно до необхідної кількості вертикальних заземлювачів пв. Коефіцієнт використання з'єднувальної смуги $\eta_c=0,3$ (табличне значення).
- 10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{заг} = \frac{R_г * R_2}{R_г * \eta_c + R_2 * n_г * \eta_г} \leq R_д \quad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{заг} < 4 \text{ Ом}$, а саме:

$$R_{заг} = \frac{18,5 * 8,1}{18,5 * 0,3 + 8,1 * 16 * 0,57} = 1,9 \leq R_д$$

Важливу увагу слід звернути на пожежну безпеку підприємства в цілому і окремих його приміщень. В приміщеннях не повинно накопичуватися сміття, непотрібний папір, мотлох та ін. речі, які не використовуються у виробничому процесі. Наявний вільний аварійний вихід за межі приміщення в разі пожежі, бути передбачені вогнегасники. Вони повинні бути в робочому стані і перевірятися згідно з нормами. У приміщеннях повинна бути пожежна сигналізація, вогнегасник.

У разі виникнення пожежі необхідно повідомити в найближчу пожежну частину, убезпечити інших працівників і по можливості прийняти кроки по запобіганню можливих наслідків та усуненню пожежі.

ВИСНОВКИ

У дипломному проекті був розроблений онлайн словник для ОС Amdroid. Розроблений додаток задовольняє всім вимогам поставленого завдання.

Було проведено порівняння традиційних словників та онлайн-словників. За багатьма параметрами домінують саме онлайн-додатки.

Для кожної мовної пари створена своя база даних яка додається при першому запуску додатку.

Була розроблена загальна структура системи, на основі якої сформований вміст кожної сторінки додатку. Розроблений дизайн є повністю адаптивним під будь-які пристрої.

Була розроблена структура бази даних, якою можна користуватися в режимі оффлайн, якщо завантажити відповідний пакет на пристрій.

В розділі «охорона праці та безпека в надзвичайних ситуаціях» був зроблений аналіз шкідливих виробничих факторів, запропоновані заходи щодо техніки безпеки, виробничої санітарії і гігієни праці.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. НПАОП 0.00-4.12-05 Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці.
2. НПАОП Б.02.005-2003 Про інструктаж, спецнавчання з питань пожежної безпеки.
3. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих.
4. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
5. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою.
6. ДБН В.2.5-28:2015 Природне і штучне освітлення.
7. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку.
8. НПАОП 40.1-1.01-97 Правила безпечної експлуатації електроустановок.
9. ДСТУ ГОСТ 12.1.012-90 ССБТ. Вибрационная безопасность.
10. ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот.
11. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
12. Android для программистов. Создаем приложения Автор: П. Дейтел, Х. Дейтел, Э. Дейтел, М. Моргано Издательство: Питер Год: 2013
13. Android. Программирование для профессионалов Филлипс Б., Стюарт К., Марсикано К., Питер, 2017, 688 с.
14. Планшет на ANDROID 5 Шпее В.В. , Наука и Техника, 2016, 320 с
15. Android. Программирование для профессионалов Филлипс Б., Стюарт К., Марсикано К., Питер, 2016, 640 с Ретабоуил С., ДМК Пресс, 2016, 518 с.
16. Android NDK. Руководство для начинающих

17. Head First. Программирование для Android Дон Гриффитс, Дэвид Гриффитс, Питер, 2016, 704 с.
18. Новейший самоучитель. Андроид для планшетов и смартфонов Виталий Леонтьев, Эксмо, 2015, 288 с
19. Матеріал з електронного ресурсу <https://www.java.com>
20. Матеріал з електронного ресурсу www.oracle.com
21. Матеріал з електронного ресурсу <https://developer.android.com>

ДОДАТОК А

Лістинг системи онлайн-словника

Файл app_bar_main.xml

```

<?xml version="1.0" encoding="utf-8"?>

/**
 * Внешний вид главной страницы
 */
<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"
"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">
        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay"
            app:contentInsetStartWithNavigation="0dp">
            <EditText
                android:id="@+id/edit_search"
                android:layout_width="match_parent"
                android:layout_height="40dp"

                android:background="@drawable/radius_edit_text"
                android:drawablePadding="3dp"
                android:paddingLeft="5dp"
            android:drawableLeft="@drawable/ic_search"
                android:textColor="#000000"

```

```

        android:textSize="20dp"

        android:hint="Search"
        android:textColorHint="#9b9a9a"/>
    </android.support.v7.widget.Toolbar>
</android.support.design.widget.AppBarLayout>
    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>

```

Файл activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
/**
 * Внешний вид навигационного меню
 */
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">
    <include
        layout="@layout/app_bar_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:menu="@menu/activity_main_drawer" />
</android.support.v4.widget.DrawerLayout>

```

Файл `bookmark_layout_item.xml`

```

<?xml version="1.0" encoding="utf-8"?>
/**
 * Внешний вид раздела “Bookmark”
 */
<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android
"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="16dp">
    <TextView
        android:id="@+id/word"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="word"
        android:layout_gravity="center"/>
    <ImageView
        android:id="@+id/btnDelete"
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_delete"/>
</LinearLayout>

```

Файл `fragment_detail.xml`

```

<?xml version="1.0" encoding="utf-8"?>
/**
 * Внешний вид внутренней страницы слова
 */
<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android
"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"

```

```

        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
android:layout_width="match_parent"
        android:layout_height="50dp"
        android:background="#FDFDFD"
        android:elevation="1.46dp"
        android:orientation="horizontal"
        android:paddingLeft="16dp"
        android:paddingRight="16dp">
        <TextView
            android:id="@+id/tvWord"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:gravity="center_vertical"
            android:text="about"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="20sp" />
        <ImageView
            android:id="@+id/btnVolume"
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:background="?actionBarItemBackground"
            android:src="@drawable/ic_volume" />
        <ImageView
            android:id="@+id/btnBookmark"
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:background="?actionBarItemBackground"
            android:src="@drawable/ic_bookmark_border" />
    </LinearLayout>
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#fff">
        <WebView
            android:id="@+id/tvWordTranslate"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="@string/hello_blank_fragment"
            android:background="#fff"/>
    </ScrollView>
</LinearLayout>

```

Файл active_main_drawer.xml

```

<?xml version="1.0" encoding="utf-8"?>

/**
 * Внешний вид навигационного меню
 */
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">
    <group android:checkableBehavior="none">
        <item
            android:id="@+id/nav_bookmark"
            android:icon="@drawable/ic_bookmark"
            android:title="Bookmark" />
    </group>
    <item android:title="NETWORK">
        <menu>
            <item
                android:id="@+id/nav_rate"
                android:icon="@drawable/ic_thumb_up"
                android:title="Rate" />
            <item
                android:id="@+id/nav_share"
                android:icon="@drawable/ic_menu_share"
                android:title="Share" />
        </menu>
    </item>
    <item android:title="OTHER">
        <menu>
            <item
                android:id="@+id/nav_help"
                android:icon="@drawable/ic_help"
                android:title="Help" />
            <item
                android:id="@+id/nav_about"
                android:icon="@drawable/ic_info"
                android:title="About" />
        </menu>
    </item>
</menu>

```


Файл main.xml

```
<?xml version="1.0" encoding="utf-8"?>
/**
 * Внешний вид выбора языковой пары
 */
<menu
xmlns:android="http://schemas.android.com/apk/res/android
"
xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/action_settings"
    android:orderInCategory="100"
    android:title="ENG-UA"
    app:showAsAction="always">
    <menu>
      <item
        android:id="@+id/action_eng_ua"
        android:title="ENG-UA"
        android:orderInCategory="100"
        app:showAsAction="never" />
      <item
        android:id="@+id/action_ua_eng"
        android:title="UA-ENG"
        android:orderInCategory="100"
        app:showAsAction="never" />
      <item
        android:id="@+id/action_ua_ua"
        android:title="UA-UA"
        android:orderInCategory="100"
        app:showAsAction="never" />
    </menu>
  </item>
</menu>
```

Файл menu_clear.xml

```
<?xml version="1.0" encoding="utf-8"?>
/**
 * Внешний вид кнопки удаления содержимого
 */
<menu
xmlns:android="http://schemas.android.com/apk/res/android
"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/action_clear"
        android:title="CLEAR"
        android:orderInCategory="100"
        app:showAsAction="always" />
</menu>
```

Файл MainActivity.java

```
package com.example.dimarozkin.diplomdict;
// **
 * Класс MainActivity с расширением
 */
public class MainActivity extends AppCompatActivity
// **
 * Реализация интерфейса
 */
implements
NavigationView.OnNavigationItemSelectedListener {
    MenuItem menuSettings;
    DictionaryFragment dictionaryFragment;
    BookmarkFragment bookmarkFragment;
    Toolbar toolbar;
    DBHelper dbHelper;
// **
 * Переопределение метода базового класса
 */
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.activity_main);
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        dbHelper = new DBHelper(this);
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new
ActionBarDrawerToggle(
            this, drawer, toolbar,
R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();
        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);

navigationView.setNavigationItemSelectedListener(this);
        dictionaryFragment = new DictionaryFragment();
        bookmarkFragment =
BookmarkFragment.getNewInstance(dbHelper);
        goToFragment(dictionaryFragment, true);
        dictionaryFragment.setOnFragmentListener(new
FragmentListener() {
            @Override
            public void onItemClick(String value) {
                String id =
Global.getState(MainActivity.this, "dic_type");
                int dicType = id == null ?
R.id.action_eng_u : Integer.valueOf(id);

                goToFragment(DetailFragment.getNewInstance(value,
dbHelper, dicType), false);
            }
        });
        bookmarkFragment.setOnFragmentListener(new
FragmentListener() {
            @Override
            public void onItemClick(String value) {
                String id =
Global.getState(MainActivity.this, "dic_type");
                int dicType = id == null ?
R.id.action_eng_u : Integer.valueOf(id);

                goToFragment(DetailFragment.getNewInstance(value,
dbHelper, dicType), false);
            }
        });

```

```

        }
    });
    EditText edit_search =
findViewById(R.id.edit_search);
    edit_search.addTextChangedListener(new
TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence
charSequence, int i, int i1, int i2) {
        }
        @Override
        public void onTextChanged(CharSequence charSequence,
int i, int i1, int i2) {
dictionaryFragment.filterValue(charSequence.toString());
        }
        @Override
        public void afterTextChanged(Editable
editable) {
        }
    });
}
@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);

    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    // Inflate the menu; this adds items to the
action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    menuSettings =
menu.findItem(R.id.action_settings);
    String id = Global.getState(this, "dic_type");
    if (id != null) {

```

```

onOptionsItemSelected(menu.findItem(Integer.valueOf(id)))
;
    }
    else {
        ArrayList<String> source =
dbHelper.getWord(R.id.action_eng_ua);
        dictionaryFragment.resetDataSource(source);
    }
    return true;
}
@Override
    public boolean onOptionsItemSelected(MenuItem item) {

        int id = item.getItemId();
        //noinspection SimplifiableIfStatement
        if (id == R.id.action_eng_ua) {
            Global.saveState(this, "dic_type",
String.valueOf(id));
            ArrayList<String> source =
dbHelper.getWord(id);
            dictionaryFragment.resetDataSource(source);
            menuSettings.setTitle("ENG-UA");
            return true;
        } else if (id == R.id.action_ua_eng) {
            Global.saveState(this, "dic_type",
String.valueOf(id));
            ArrayList<String> source =
dbHelper.getWord(id);
            dictionaryFragment.resetDataSource(source);
            menuSettings.setTitle("UA-ENG");
            return true;
        } else if (id == R.id.action_ua_ua) {
            Global.saveState(this, "dic_type",
String.valueOf(id));
            ArrayList<String> source =
dbHelper.getWord(id);

            dictionaryFragment.resetDataSource(source);
            menuSettings.setTitle("UA-UA");
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
    @SuppressWarnings("StatementWithEmptyBody")

```

```

@Override
public boolean onNavigationItemSelected(MenuItem
item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();
    if (id == R.id.nav_bookmark) {
        String activeFragment =
String.valueOf(getSupportFragmentManager().findFragmentBy
Id(R.id.fragment_container)).getClass().getSimpleName();
        if
(!activeFragment.equals(BookmarkFragment.class.getSimpleN
ame())) {
            goToFragment(bookmarkFragment, false);
        }
    }
    DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
void goToFragment(Fragment fragment, boolean isTop) {
    FragmentManager fragmentManager =
getSupportFragmentManager();
    FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();

fragmentTransaction.replace(R.id.fragment_container,
fragment);
    if (!isTop)
        fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
}
@Override
public boolean onPrepareOptionsMenu(Menu menu) {
    String activeFragment =
String.valueOf(getSupportFragmentManager().findFragmentBy
Id(R.id.fragment_container)).getClass().getSimpleName();
    if
(activeFragment.equals(BookmarkFragment.class.getSimpleNa
me())) {
        menuSettings.setVisible(false);

toolbar.findViewById(R.id.edit_search).setVisibility(View
.GONE);
        toolbar.setTitle("Bookmark");
    }
}

```

```

        } else {
            menuSettings.setVisible(true);

toolbar.findViewById(R.id.edit_search).setVisibility(View
.VISIBLE);

toolbar.setTitle(" ");
    }
    return true;
}

```

Файл DBHelper.java

```

package com.example.dimarozkin.diplomdict;
// **
* Импорт нужных пакетов
*/
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import com.example.dimarozkin.diplomdict.R;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
// **
* Расширение класса DBHelper для работы с базой данных SQLite
*/
public class DBHelper extends SQLiteOpenHelper {
// **
* Внутренние классы класса DBHelper
*/
    private Context mContext;
        public static final String DATABASE_NAME =
"my_dic.db";
        public static final int DATABASE_VERSION = 1;
        private String DATABASE_LOCATION = "";

```

```

private String DATABASE_FULL_PATH = "";
private final String TBL_ENG_UA = "eng_ua";
private final String TBL_UA_UA = "ua_ua";
private final String TBL_UA_ENG = "ua_eng";
private final String TBL_BOOKMARK = "bookmark";
private final String COL_KEY = "key";
private final String COL_VALUE = "value";
public SQLiteDatabase mDB;
// **
* Класс хранения базы данных и проверки на ее наличие
*/
public DBHelper(Context context) {
    super(context, DATABASE_NAME, null,
DATABASE_VERSION);
    mContext = context;
    DATABASE_LOCATION = "data/data/" +
mContext.getPackageName() + "/database/";
    DATABASE_FULL_PATH = DATABASE_LOCATION +
DATABASE_NAME;
    if (!isExistingfDB()) {
        try {
            File dbLocation = new
File(DATABASE_LOCATION);
            dbLocation.mkdirs();

extractAssetsToDatabaseDirectory(DATABASE_NAME);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    mDB =
SQLiteDatabase.openOrCreateDatabase(DATABASE_FULL_PATH,
null);
}
boolean isExistingfDB() {
    File file = new File(DATABASE_FULL_PATH);
    return file.exists();
}
public void extractAssetsToDatabaseDirectory(String
fileName)
        throws IOException {
    int length;
    InputStream sourceDatabase =
this.mContext.getAssets().open(fileName);

```



```

        File destinationPath = new
File(DATABASE_FULL_PATH);
        OutputStream destination = new
FileOutputStream(destinationPath);
        byte[] buffer = new byte[4096];
        while ((length = sourceDatabase.read(buffer)) >
0) {
            destination.write(buffer, 0, length);
        }
        sourceDatabase.close();
        destination.flush();
        destination.close();
    }
    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
}

@Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase,
int i, int il) {
    }
    public ArrayList<String> getWord(int dicType) {
        String tableName = getTableName(dicType);
        String q = " SELECT * FROM " + tableName;
        Cursor result = mDB.rawQuery(q, null);
        ArrayList<String> source = new ArrayList<>();
        while (result.moveToNext()) {

source.add(result.getString(result.getColumnIndex(COL_KEY
)));
        }
        result.close();
        return source;
    }
    public Word getWord(String key, int dicType) {
        String tableName = getTableName(dicType);
        String q = "SELECT * FROM " + tableName + " WHERE
upper([key]) = upper (?);";
        Cursor result = mDB.rawQuery(q, new
String[]{key});
        Word word = new Word();
        while (result.moveToNext()) {
            word.key =
result.getString(result.getColumnIndex(COL_KEY));

```

```

        word.value =
result.getString(result.getColumnIndex(COL_VALUE));
    }
    result.close();
    return word;
}
// **
* Работа с таблицами базы данных
*/
public void addBookmark(Word word) {
    try {
        String q = "INSERT INTO bookmark ([" +
COL_KEY + "], [" + COL_VALUE + "]) VALUES (?, ?)";
        mDB.execSQL(q, new Object[]{word.key,
word.value});
    } catch (SQLException ex) {
    }
}
public void removeBookmark(Word word) {
    try {
        String q = "DELETE FROM bookmark WHERE upper
([" + COL_KEY + "]) = upper (?) AND [" + COL_VALUE + "] =
'?'";
mDB.execSQL(q, new Object[]{word.key, word.value});
    } catch (SQLException ex) {
    }
}
public void removeBookmark(String key) {
    try {
        String q = "DELETE FROM bookmark WHERE upper
([" + COL_KEY + "]) = upper (?)";
        mDB.execSQL(q, new Object[]{key});
    } catch (SQLException ex) {
    }
}
public ArrayList<String> getAllWordFromBookmark() {
    String q = "SELECT * FROM bookmark ORDER BY
[date] DESC";
    Cursor result = mDB.rawQuery(q, null);
    ArrayList<String> source = new ArrayList<>();
    while (result.moveToNext()) {
source.add(result.getString(result.getColumnIndex(COL_KEY
)));
}
}

```

```

        }
        result.close();
        return source;
    }
    public boolean IsWordMark(Word word) {
        String q = "SELECT * FROM bookmark WHERE upper
([key]) = upper(?) AND [value] = ?;";
        Cursor result = mDB.rawQuery(q, new
String[]{word.key, word.value});
        result.close();
        return result.getCount() > 0;
    }
    public Word getWordFromBookmark(String key) {
        String q = "SELECT * FROM bookmark WHERE upper
([key]) = upper(?);";
        Cursor result = mDB.rawQuery(q, new
String[]{key});
        Word word = new Word();
        while (result.moveToNext()) {
            word = new Word();
            word.key =
result.getString(result.getColumnIndex(COL_KEY));
            word.value =
result.getString(result.getColumnIndex(COL_VALUE));
        }
        result.close();
        return word;
    }
    public String getTableName(int dicType) {
        String tableName = "";
        if (dicType == R.id.action_ua_eng) {
tableName = TBL_UA_ENG;
        } else if (dicType == R.id.action_ua_ua) {
            tableName = TBL_UA_UA;
        } else if (dicType == R.id.action_eng_ua) {
            tableName = TBL_ENG_UA;
        }
        return tableName;
    }
    public void clearBookmark() {
        try {
            String q = "DELETE FROM bookmark;";
            mDB.execSQL(q);
        } catch (SQLException ex) {

```



```

        return inflater.inflate(R.layout.fragment_detail,
            container, false);
    }
    @Override
    public void onCreateView(View view, @Nullable Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        tvWord = view.findViewById(R.id.tvWord);
        tvWordTranslate =
view.findViewById(R.id.tvWordTranslate);
        btnBookmark =
view.findViewById(R.id.btnBookmark);
        btnVolume = view.findViewById(R.id.btnVolume);
        final Word word = mDBHelper.getWord(value,
mDicType);
        tvWord.setText(word.key);
        tvWordTranslate.loadDataWithBaseURL(null,
word.value, "text/html", "utf-8", null);
        Word bookmarkWord =
mDBHelper.getWordFromBookmark(value);
        int isMark = bookmarkWord == null ? 0 : 1;
        btnBookmark.setTag(isMark);
        int icon = bookmarkWord == null ?
R.drawable.ic_bookmark_border :
R.drawable.ic_bookmark_fill;
        btnBookmark.setImageResource(icon);
        btnBookmark.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                int i = (int) btnBookmark.getTag();
                if (i == 0) {

btnBookmark.setImageResource(R.drawable.ic_bookmark_fill)
;

                btnBookmark.setTag(1);
                mDBHelper.addBookmark(word);
            } else if (i == 1) {

btnBookmark.setImageResource(R.drawable.ic_bookmark_borde
r);

                btnBookmark.setTag(0);
                mDBHelper.removeBookmark(word);
            }
        }
    }
}

```

});

```

    }
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
    }
    @Override
    public void onDetach() {
        super.onDetach();
    }
}

```

Файл DictionaryFragment.java

```

package com.example.dimarozkin.diplomdict
public class DictionaryFragment extends Fragment {
    private FragmentListener listener;
    ArrayAdapter<String> adapter;
    ListView dicList;
    private String value = "Hello everyone!";
    private ArrayList<String> mSource = new
ArrayList<String>();
    public DictionaryFragment() {
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater,
    ViewGroup container,
                                Bundle savedInstanceState) {
        return
inflater.inflate(R.layout.fragment_dictionary, container,
false);
    }
    @Override
    public void onViewCreated(View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
    }
}

```

```

        dicList = view.findViewById(R.id.dictionaryList);
        adapter = new ArrayAdapter<String>(getContext(),
android.R.layout.simple_list_item_1,mSource);

dicList.setAdapter(adapter);
        dicList.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?>
parent, View view, int position, long id) {
                if ( listener != null )

listener.onItemClick(mSource.get(position));
            }
        });
    }
    public void resetDataSource(ArrayList<String>
source){
        mSource = source;
        adapter = new ArrayAdapter<String>(getContext(),
android.R.layout.simple_list_item_1,source);
        dicList.setAdapter(adapter);
    }
    public void filterValue(String value){
        // adapter.getFilter().filter(value);
        int size = adapter.getCount();
        for (int i = 0 ; i<size;i++){
            if(adapter.getItem(i).startsWith(value)){
                dicList.setSelection(i);
                break;
            }
        }
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
    }
    @Override
    public void onDetach() {
        super.onDetach();
    }
    public void setOnFragmentManager(FragmentListener
listener){
        this.listener = listener;
    }

```

```

    }
}

```

Файл BookmarkAdapter.java

```

package com.example.dimarozkin.diplomdict
public class BookmarkAdapter extends BaseAdapter {
    private ListItemListener listener;
    private ListItemListener listenerBtnDelete;
    Context mContext;
    ArrayList<String> mSource;
    public BookmarkAdapter(Context context,
ArrayList<String> source) {
        this.mSource = source;
        this.mContext = context;
    }
    @Override
    public int getCount() {
        return mSource.size();
    }
    @Override
    public Object getItem(int i) { return mSource.get(i);
    }
    @Override
    public long getItemId(int i) {
        return 0;
    }
    @Override
    public View getView(final int i, View view, ViewGroup
viewGroup) {
        ViewHolder viewHolder;
        if (view == null) {
            viewHolder = new ViewHolder();
            view =
LayoutInflater.from(mContext).inflate(R.layout.bookmark_l
ayout_item, viewGroup, false);
            viewHolder.textView =
view.findViewById(R.id.word);

```



```

        viewHolder.btnDelete =
view.findViewById(R.id.btnDelete);
        view.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) view.getTag();
    }
    viewHolder.textView.setText(mSource.get(i));
    view.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (listener != null)
                listener.onItemClick(i);
        }

    });
    viewHolder.btnDelete.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if ( listenerBtnDelete != null )
                listenerBtnDelete.onItemClick(i);
        }
    });
    return view;
}
public void removeItem(int position){
    mSource.remove(position);
}
public void clear() {
    mSource.clear();
}
class ViewHolder {
    TextView textView;
    ImageView btnDelete;
}
public void setOnItemClick(ListItemListener
listItemListener){
    this.listener = listItemListener;
}
public void setOnItemDeleteClick(ListItemListener
listItemListener){
    this.listenerBtnDelete = listItemListener;
}

```

ДОДАТОК Б

Комп'ютерна презентація

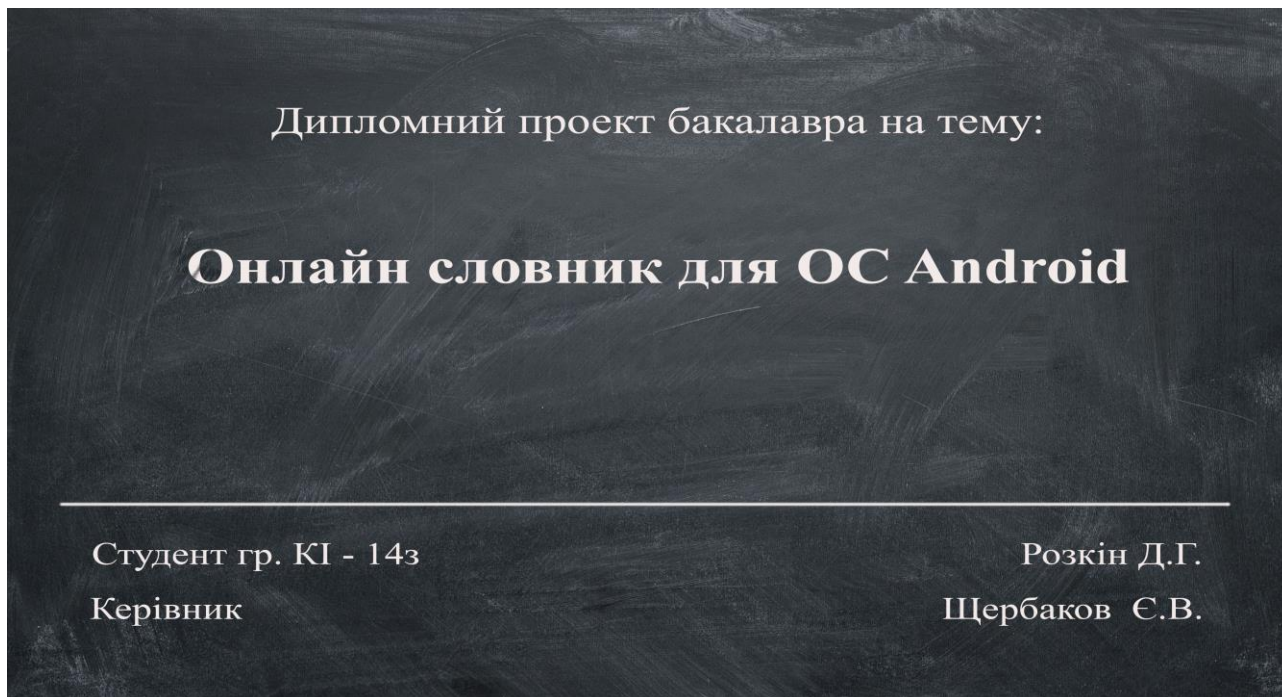


Рисунок Б.1 – Титульний лист

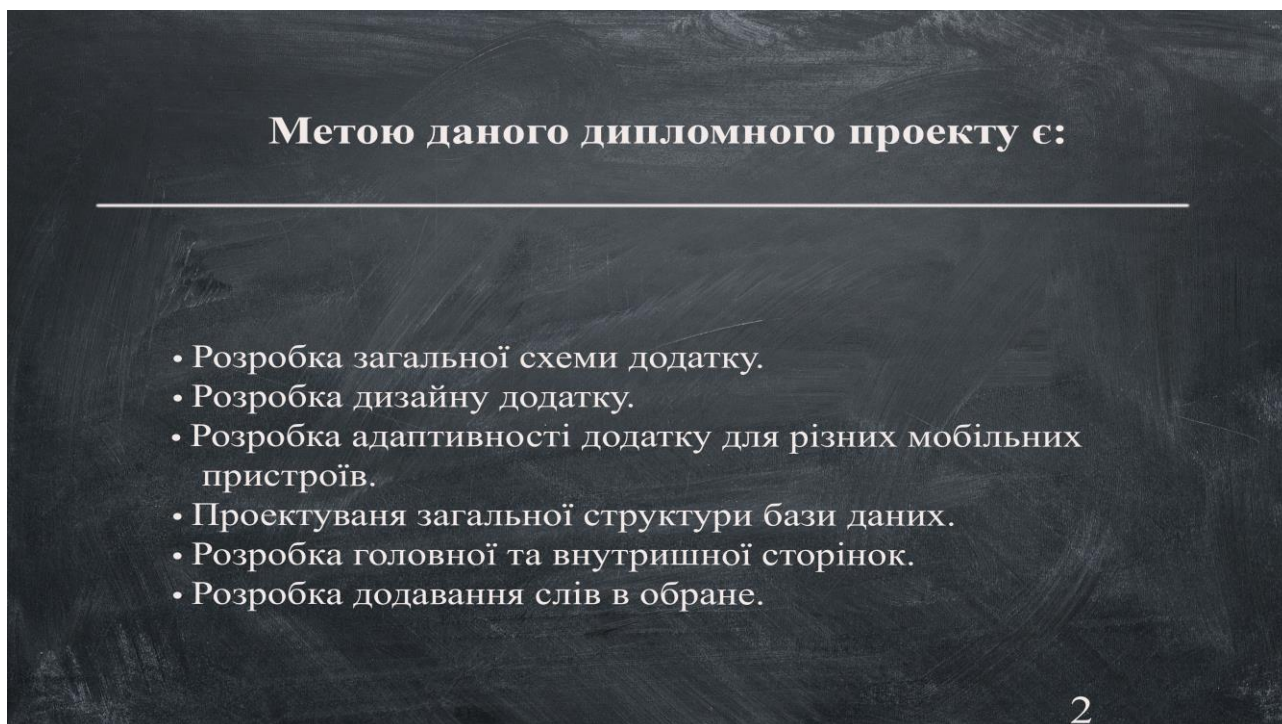
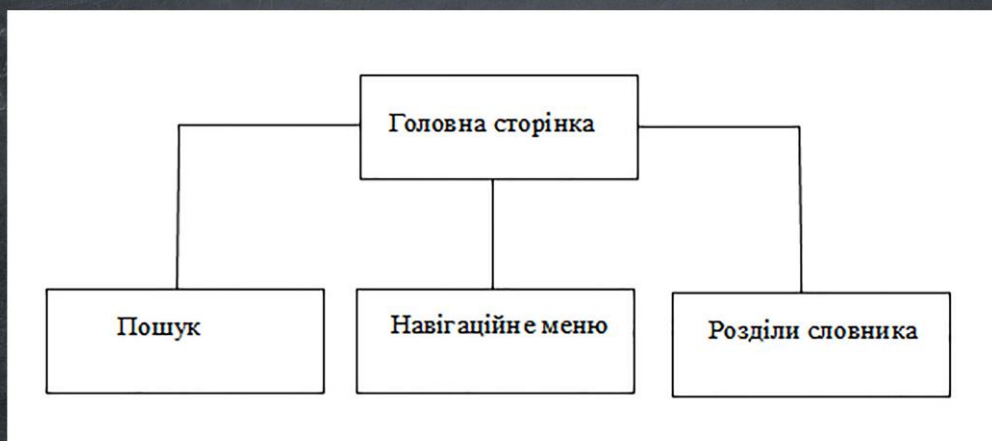


Рисунок Б.2 – Мета дипломного проекту

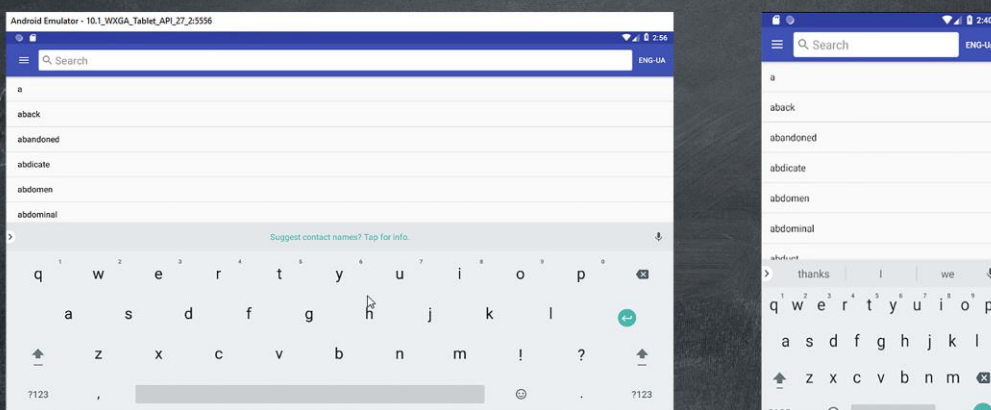
Загальна структура онлайн-словника



3

Рисунок Б.3 – Загальна структура онлайн словника

Дизайн системи



Дизайн системи адаптивний під будь-які пристрої

4

Рисунок Б.4 – Дизайн системи

Загальна структура бази даних

The screenshot shows the SQLite database structure with four tables. Each table has a primary key and specific data types for its columns.

Table name	Name	Data type	Primary Key	Foreign Key	Unique
eng_ua	1 key	VARCHAR	Yes	No	No
	2 value	TEXT	No	No	No
ua_ua	1 key	VARCHAR	Yes	No	No
	2 value	TEXT	No	No	No
ua_eng	1 key	VARCHAR	Yes	No	No
	2 value	TEXT	No	No	No
bookmark	1 key	VARCHAR	Yes	No	No
	2 value	TEXT	No	No	No
	3 date	DATETIME	No	No	No

В якості СУБД використовується SQLite. База даних складається з декількох таблиць, які впорядковано зберігають інформацію.

5

Рисунок Б.5 – Структура бази даних

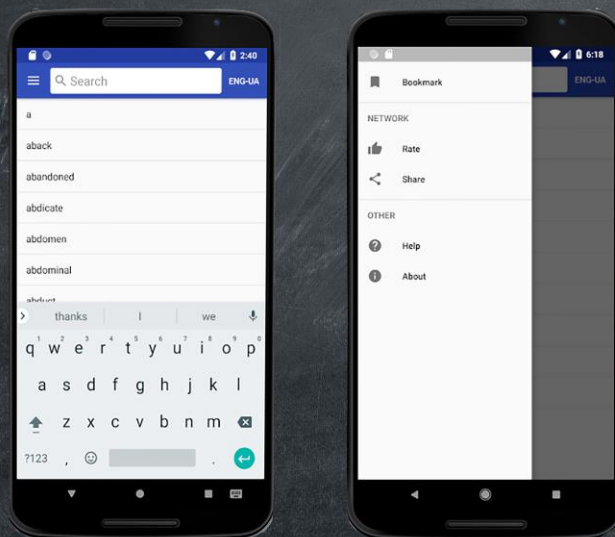
Схема взаємодії компонентів



6

Рисунок Б.5 – Схема взаємодії компонентів

Головна сторінка



Зовнішній
вигляд головної
сторінки додатку
та навігаційного
меню

7

Рисунок Б.5 – Головна сторінка додатку

Внутрішня сторінка

aback

ə'bæk, /ə'bæk/; Type:
noun, adverb;

(rare) Towards the back or
rear; backwards; away.

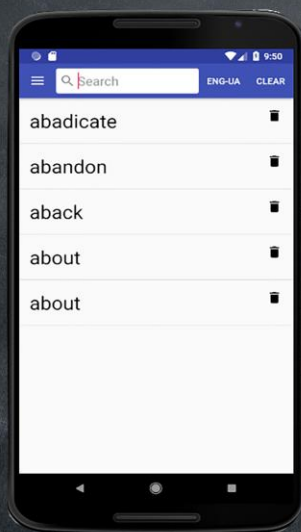
назад, за, поза, тому,
давно

Внутрішня сторінка
слова з транскрипцією,
частиною мови,
поясненням
англійською та
перекладом

8

Рисунок Б.5 – Зображення внутрішньої сторінки додатку

Сторінка обраного



Сторінка обраного з функціями повного очищення («CLEAR») та видалення окремого слова.

9

Рисунок Б.9 – Сторінка обраного

Висновки

- У дипломному проекті був розроблений онлайн словник для ОС Android.
- Розроблений додаток задовольняє всім вимогам поставленого завдання.
- За багатьма параметрами домінують саме онлайн-додатки.
- Для кожної мовної пари створена своя база даних яка додається при першому запуску додатку.
- Була розроблена загальна структура системи, на основі якої сформований вміст кожної сторінки додатку. Розроблений дизайн є повністю адаптивним під будь-які пристрої.
- Була розроблена структура бази даних, якою можна користуватися в режимі оффлайн, якщо завантажити відповідний пакет на пристрій.

10

Рисунок Б.10 – Висновки

