

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Мобільний додаток Індустріально-педагогічного технікуму
Конотопського інституту Сумського державного університету з
Telegram-ботом»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ.м-82 Поволоцький Богдан Олегович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«___» грудня 2019 р.

Науковий керівник

(підпис)

к.т.н., доц., Парфененко Ю. В.

Голова комісії

(підпис)

Шифрін Д. М.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2019

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2019 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Поволоцький Богдан Олегович

(прізвище, ім'я, по батькові)

1 Тема проекту *Мобільний додаток Індустріально-педагогічного технікуму Конопського інституту Сумського державного університету з Telegram-ботом*

затверджена наказом по університету від «19» листопада 2019 р. №2305-III

2 Термін здачі студентом закінченого проекту « 10 » грудня 2019 р.

3 Вхідні дані до проекту модуль з розкладом навчального процесу ІІТ КІ Сум ДУ, чат-бот на сервері Dialogflow

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *аналіз предметної області, постановка задачі та методи дослідження, моделювання мобільного додатку, реалізація мобільного додатку*

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) *Актуальність проекту, мета і задачі проекту, дослідження аналогів, порівняння аналогів, вимоги до мобільного додатку, контекстна діаграма розробки додатку, діаграма декомпозиції розробки додатку, діаграма варіантів використання, діаграма послідовності роботи чат-бота, засоби реалізації, демонстрація додатку, висновки*

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускного проекту	Термін виконання етапів проекту	Примітка
1	Огляд існуючих мобільних додатків	15.09.19 – 18.09.19	
2	Огляд можливостей чат-ботів	21.09.19 – 24.09.19	
3	Мета та задачі	27.09.19 – 28.09.19	
4	Вибір засобів реалізації	05.10.19 – 10.10.19	
5	Моделювання мобільного додатку	11.10.19 – 23.10.19	
6	Розробка мобільного додатку з чат - ботом	25.10.19 – 25.11.19	
7	Тестування додатку	26.11.19 – 28.11.19	
8	Формування документації	05.11.19 – 05.12.19	

Магістрант _____

Поволоцький Б.О.

Керівник роботи _____

к.т.н., доц. Парфененко Ю.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Мобільний додаток Індустріально-педагогічного технікуму Конотопського інституту Сумського державного університету з Telegram-ботом».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 30 найменувань, додатків. Загальний обсяг роботи – 76 сторінок, у тому числі 53 сторінки основного тексту, 3 сторінки списку використаних джерел, 23 сторінки додатків.

Кваліфікаційну роботу магістра присвячено розробці мобільного додатку Індустріально-педагогічного технікуму Конотопського інституту Сумського державного університету з Telegram ботом.

В ході даної роботи був виконаний аналіз і формування вимог до спроектованого мобільного додатку.

На основі проектувальних даних було розроблено макет спроектованого мобільного додатку. Розробка проводилася в Android Studio з використанням Firebase як інструмент адміністрування бази даних та сервісу Dialogflow як помічник в розробці чат-бота. Розроблений додаток орієнтований на студентів ІІТ КІ СумДУ, оскільки це значно заощадить їхній час.

Практичне значення роботи полягає в отриманні студентами та абітурієнтами зручного та простого у використанні мобільного додатку з чат-ботом, який допоможе швидко знайти потрібну інформацію.

Ключові слова: ІІТ КІ СумДУ, технікум, мобільний додаток, студент, android studio, firebase, dialogflow, чат-бот.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Аналіз застосування інформаційних технологій в діяльності технікуму.....	8
1.2 Огляд існуючих мобільних додатків	11
1.3 Огляд можливостей чат-ботів	17
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	20
2.1 Мета та задачі.....	20
2.2 Алгоритм роботи чат-бота	22
2.3 Вибір засобів реалізації мобільного додатку для ЗВО	23
3 МОДЕЛЮВАННЯ МОБІЛЬНОГО ДОДАТКУ	29
3.1 Моделювання процесу розробки мобільного додатку.....	29
3.2 Моделювання роботи з мобільним додатком	31
3.3 Розробка макетів екранів мобільного додатку	34
4 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ	38
4.1 Розробка чат-бота в сервісі Dialogflow.....	38
4.2 Реалізація бази даних чат-бота.....	42
4.3 Розробка модулів програмного продукту	44
4.4 Тестування програмного продукту	48
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТОК А.....	54
ДОДАТОК Б.....	66

ВСТУП

Протягом останніх декількох років мобільні технології стрімко розвиваються за рахунок вдосконалення бездротового зв'язку, розширення функціональності мобільних пристроїв і появи мобільних платформ з відкритим вихідним кодом. В даний час мобільні пристрої можуть використовуватися не тільки для комунікації, а й в освітніх і комерційних цілях. Для підтримки цих можливостей розробляються різні мобільні додатки.

На сьогоднішній день все більше людей віддають перевагу мобільному і швидкому доступу до інформації. З кожним роком відсоток користувачів мобільних версій сайтів зростає, а, отже, комп'ютерні версії стають все менш популярними.

Навчальні заклади мають офіційні сайти, що містять необхідну студентам інформацію. На даний момент користування сайтом з мобільного пристрою створює ряд незручностей. Потрібно запам'ятовувати адресу сайту або шукати його в пошуковій мережі. На самому сайті серед дрібного тексту шукати потрібний розділ, серед величезної кількості посилань, різного ступеня корисності.

Отже, використання мобільного додатку, присвяченого Індустріально-педагогічному технікуму КІ СумДУ, не тільки спростить сприйняття інформації на мобільному пристрої, але і дозволить користувачеві оперативно взаємодіяти з інтерактивними елементами сайту.

Оскільки більшість сучасних мобільних пристроїв працюють на базі операційної системи Android, то метою даної роботи є реалізація мобільного додатку на базі цієї операційної системи, що забезпечує своєчасне доведення актуальної інформації до учасників освітнього процесу за допомогою розроблено мобільного додатку з чат-ботом. Також передбачається, що реалізований додаток дозволить привернути додаткову увагу студентів до подій з життя навчального закладу.

Для досягнення даної мети були поставлені наступні завдання:

1. Проаналізувати предметну область і аналогічні проекти в мережі Інтернет.
2. Проаналізувати вихідні дані з сайту ЗВО та підготувати ескізи і проект продукту.
3. Реалізувати мобільний додаток з чат-ботом під управлінням обраної операційної системи.
4. Провести тестування мобільного додатку.
5. Наповнити додаток даними, взятими з офіційного сайту Індустріально-педагогічному технікуму КІ СумДУ, розмістити на сайті і в офіційному магазині додатків «Google Play» і забезпечити можливість завантажування додатку.

Практичне значення роботи полягає в отриманні студентами та абітурієнтами зручного та простого у використанні мобільного додатку з чат-ботом, який допоможе швидко знайти потрібну інформацію.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз застосування інформаційних технологій в діяльності технікуму

На сьогодні інформаційні технології(ІТ) широко використовуються закладами вищої освіти (ЗВО) для підвищення якості навчання й освіти . Вони дозволяють здійснювати дистанційну взаємодію викладачів і студентів, іншими словами - здобувати освіту дистанційно. За допомогою інформаційних технологій студент може отримати потрібну йому інформацію зайшовши на сайт ЗВО. Не виключенням є технікум, де ІТ застосовується у вигляді сайту, на якому є вся необхідна інформація для студентів, абітурієнтів та вчителів.

Технікум [1] – заклад вищої освіти першого рівня акредитації, який проводить освітню діяльність, пов'язану зі здобуттям певної вищої освіти.

Індустріально-педагогічний технікум Конотопського інституту СумДУ має свою матеріальну базу, яка [2] забезпечує у повному обсязі належну підготовку фахівців.

На сайті технікуму [3] знаходиться інформація про навчання студентів, методичну роботу, кадрове забезпечення, історію технікуму, інформацію для абітурієнтів. Також сайт має сторінки з розклад занять та замінами в розкладі.

Реалізація сайту відбувалася за допомогою WordPress [4].

WordPress – це система управління контентом, заснована на PHP і MySQL, яка зазвичай використовується з серверами баз даних MySQL.

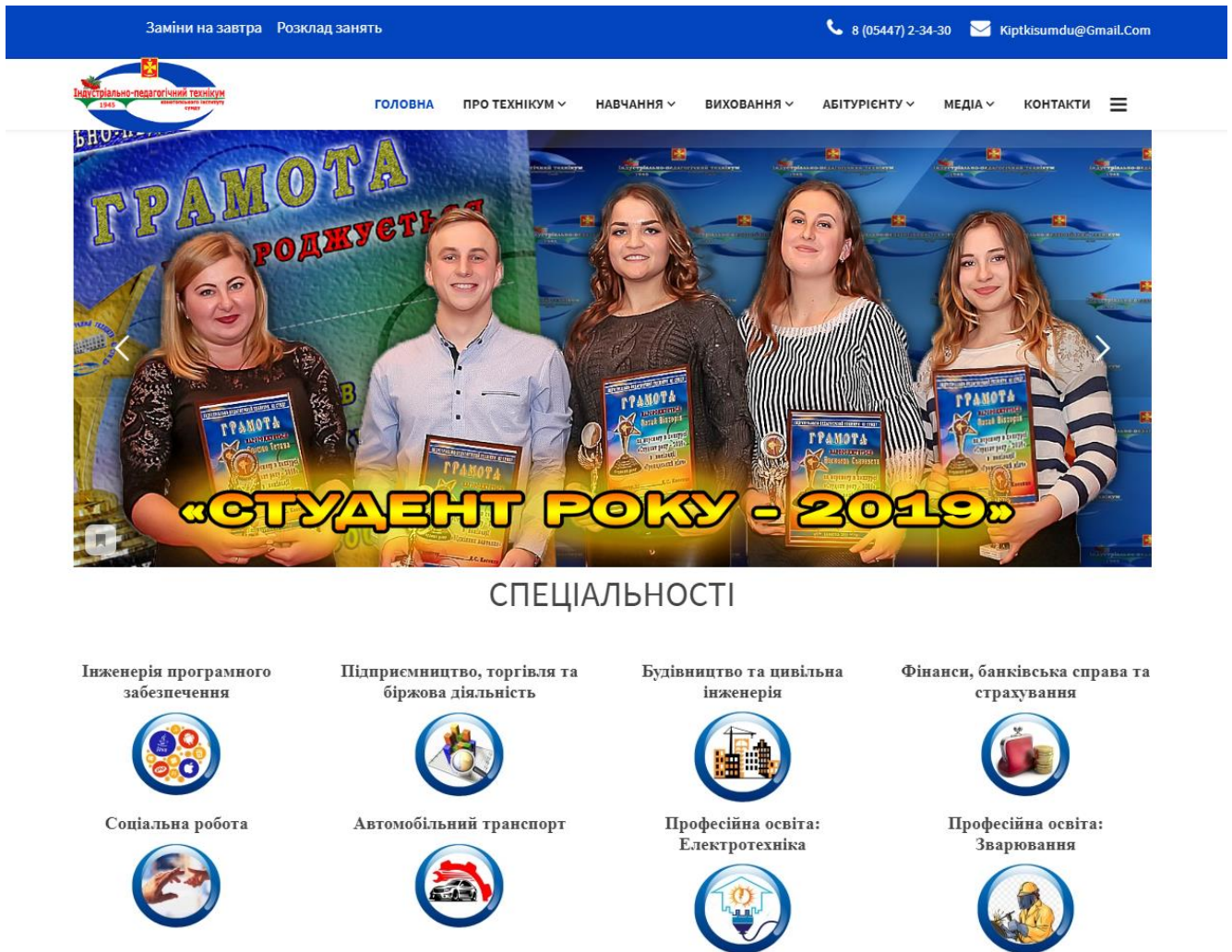


Рисунок 1.1 – Головна сторінка сайту технікуму

Розділи сайту: головна, про технікум, навчання, виховання, абітурієнту, медіа, контакти. Кожен розділ поділений на декілька підрозділів, які містять відповідні інформаційні сторінки.

У розділі «Про технікум» можна знайти інформацію про історію технікуму, матеріальну базу та кадрове забезпечення технікуму. Розділ «Навчання» включає в себе інформацію про стипендіальне забезпечення, інформацію для викладачів та майстрів. Розділ «Виховання» направлений на діяльність студентів, він містить інформацію про різні виховні та волонтерські заходи, фізичне виховання студентів.

Розділ «Абітурієнту» містить інформацію для вступників: список спеціальностей, умови прийому, банківські реквізити та інше. Розділ «Медіа» містить фото та відео матеріалу з життя студентів технікуму.

Сайт реалізований двома мовами: українська, англійська.



ГОЛОВНА ПРО ТЕХНІКУМ НАВЧАННЯ ВИХОВАННЯ АБІТУРІЄНТУ МЕДІА КОНТАКТИ

741

Понеділок

Регламент пар	№ пари	Назва	Аудиторія
8 ³⁰ - 9 ⁵⁰	1	ОПІ Зимовець В.І.	222
10 ⁰⁰ - 11 ²⁰	2	ІЗВП Осадчий А.С.	314 (Б)
З обідом 12 ¹⁰ - 13 ³⁰	3	ОКМ Смаглюк П.С.	222
13 ⁴⁰ - 15 ⁰⁰	4		

Рисунок 1.2 – Сторінка з розкладом

Зміни в розкладі

учбових занять ІІТ КІ Сум ДУЗ

зміни в розкладі
на 25 жовтня 2019р.

Група	№ пари	Назва предмета	Назва предмета	Аудиторія
321	2	МТКМ	ФВ	
	3	ФВ	Психологія	112
221	3	-----	УМ	221
931	1	-----	ОП	229
741	1	КПЗ	ІЗВП	314 (Б)
	2	ЛМІ	ПМА	314 (Б)
	3	-----	ПМА	314 (Б)
521\621	3	-----	УЛ	216

Рисунок 1.3 – Сторінка з замінами



Косенко Дмитро Сергійович

*посада: Директор
контактна інформація:*

тел: +(05447) 2-34-30

e-mail: dm.kosenko02@gmail.com



Кравченко Ольга Олександрівна

посада: Заступник директора з навчальної роботи

контактна інформація:

тел: +(05447) 2-34-30

e-mail: o.kravchenko@kprt.sumdu.edu.ua

+

Рисунок 1.4 – Сторінка адміністрації технікуму

Проаналізувавши сайт, можна зробити висновок, що розробка інформативного мобільного додатку для технікуму буде вирішенням проблеми сприйняття інформації, тому що практично кожен у наш час має мобільний пристрій з доступом до мережі Інтернет, а чат-бот допоможе швидко знайти потрібну інформацію як студентам, так і абітурієнтам та вчителям.

1.2 Огляд існуючих мобільних додатків

При розробці нового програмного продукту необхідно розглянути вже готові рішення. Даний аналіз допоможе виявити переваги і недоліки вже представлених реалізацій, щоб врахувати їх при власній розробці.

На даний момент спеціалізованих програм для ЗВО існує досить багато. Однак, більшість додатків на практиці виявляються вузькоспеціалізованими. Як правило додаток орієнтований або на виконання однієї-двох функцій, або на певний ЗВО.

При пошуку були виділені кілька категорій додатків:

– Розклад. Основна частина мобільних додатків для ЗВО являють собою розклад занять. Як правило, вони робляться навіть не самим ЗВО, а стороннім розробником, зацікавленим в розміщенні реклами з метою отримання прибутку.

- Система дистанційної освіти.
- Додатки з набором різних функцій.

Остання категорія викликає найбільший інтерес, так як в ній сконцентровані додатки, які вирішують головні завдання ЗВО. При виконанні огляду були виділені кілька додатків з магазину додатків GooglePlay.

Мобільний університет [5]

"Мобільний університет" - це мобільний додаток, який зацікавить як студентів, так і викладачів, завдяки можливості отримання через мобільний телефон або планшет інформації про розклад навчальних занять, навчальних планів і успішності.

Функції додатку:

- розклад навчальних занять;
- навчальний план;
- успішність.

До переваг даного додатку можна віднести:

- масштабованість;
- можливість авторизації;
- простий дизайн;
- доступ до розкладу офлайн.

Недоліками даного додатку є:

- відсутня можливість перегляду розкладу на потрібний період;
- передбачається обов'язкове використання вузом програмного продукту 1С:Університет ПРОФ;
- малий функціонал;

Даний мобільний додаток призначений для операційної системи Android версій 4.1 та вище. Додаток можна завантажити в магазині Google Play.

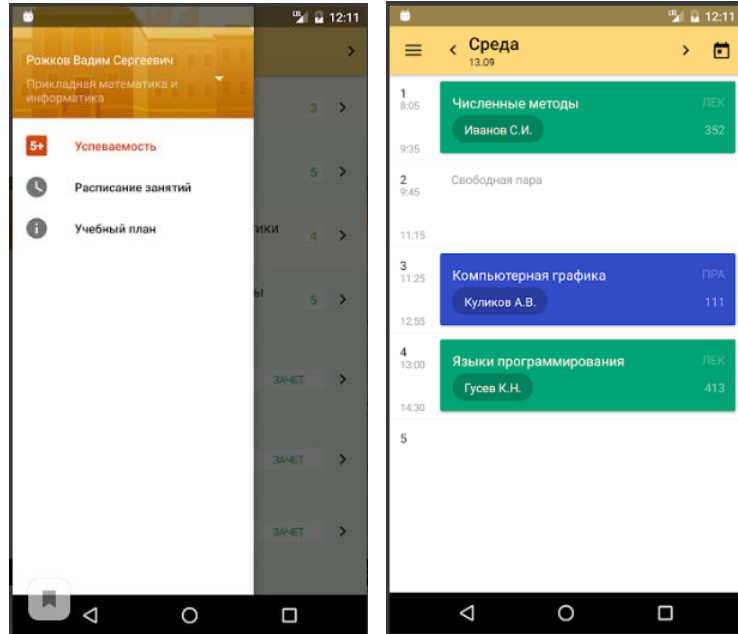


Рисунок 1.5 – Додаток «Мобільний університет»

Мій університет [6]

Даний додаток забезпечує обмін інформацією між студентами і викладацьким складом, дозволяє відстежувати зміни в розкладі.

Функції додатку:

- розклад навчальних занять;
- формує рейтинг успішності;
- надає доступ до навчальних матеріалів;
- містить актуальні вакансії потенційних роботодавців;
- містить відомості про поточну успішність;
- обмін інформацією між студентами і викладачами у вигляді чату.

Переваги додатку:

- хороший функціонал;
- доступний розклад для студента та викладача;

— система інформування про новини.

Недоліки додатку:

- відсутність пошуку по предметам та вчителям;
- відсутність відображення розкладу на тиждень вперед;
- немає автономного доступу;
- дані не оновлюються.

Даний мобільний додаток призначений для операційної системи Android версій 4.1 та вище. Додаток можна завантажити в магазині Google Play.

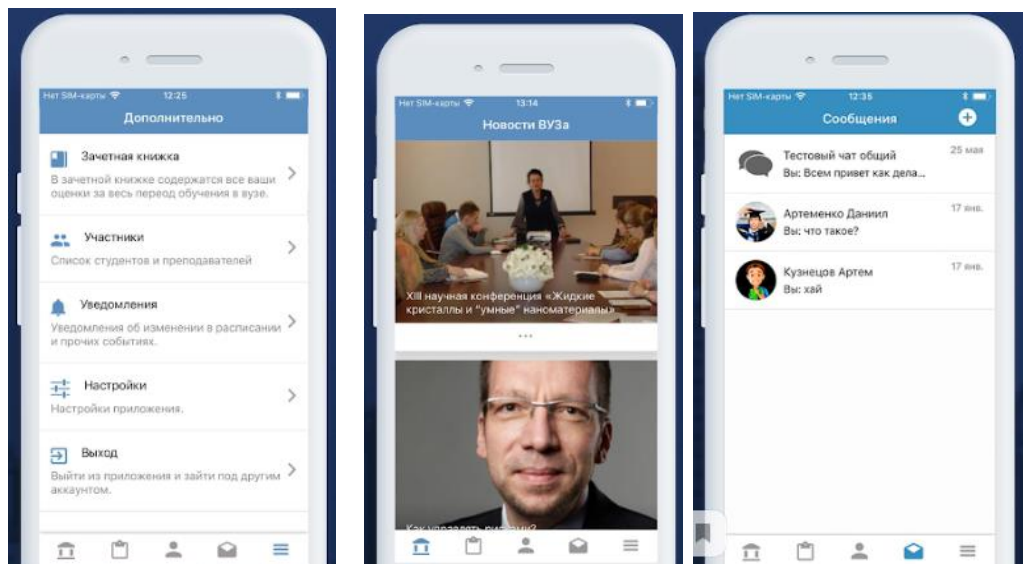


Рисунок 1.6 – Функціонал додатку «Мій університет»

Універ - розклад занять [7]

Даний додаток забезпечує обмін інформацією між студентами і викладацьким складом у вигляді чату, містить розклад для понад 440 університетів та інститутів, понад 35 000 груп.

Функції додатку:

- актуальний розклад навчальних занять для студентів та викладачів;
- має власний віджет;
- обмін інформацією між студентами і викладачами у вигляді чату.

Переваги додатку:

- хороший функціонал;
- обмін повідомленнями;
- автономність;
- доступний розклад для студента та викладача.

Недоліки додатку:

- відсутність пошуку по предметам та вчителям;
- мало довідкової інформації;
- відсутність актуальних новин та заходів.

Даний мобільний додаток призначений для операційної системи Android версій 4.1 та вище. Додаток можна завантажити в магазині Google Play.

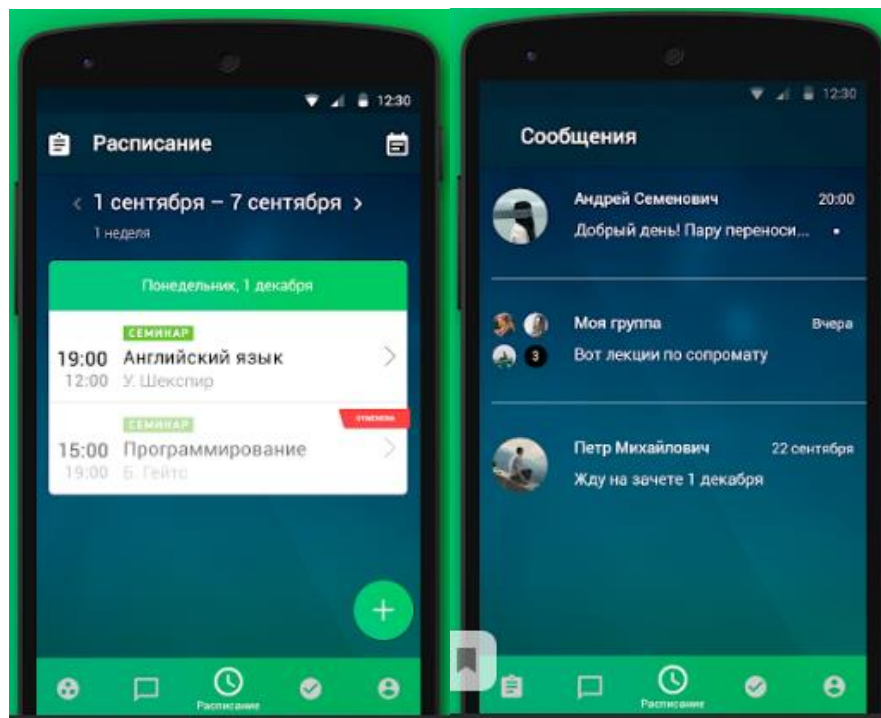


Рисунок 1.7 – Функціонал додатку «Універ - розклад занять»

За результатами проведеного аналізу можна зробити висновок: щоб реалізувати даний додаток, необхідно мати певний стандарт, якого будуть

дотримуватися більшість ЗВО. Зараз у багатьох ЗВО є власні системи дистанційної освіти, часто це обумовлено відсутністю необхідної функціональності.

Таким чином, на даний момент доцільно створювати додатки, націлені на конкретні завдання, у вирішенні яких зацікавлений ЗВО. Усі вищеразглянуті додатки мають певні недоліки, деякі з них мають істотні недоліки, які роблять додатки малоприсадибними для використання. Отже, на основі розглянутих додатків було прийнято рішення створити власний інформативний додаток технікуму з чат-ботом. Після аналізу даних була побудована наступна порівняльна таблиця.

Таблиця 1.1 Порівняльний аналіз існуючих аналогів

Функція	Назва мобільного додатку		
	Мобільний університет	Мій університет	Універ - розклад занять
Можливість перегляду спеціальностей ЗВО	-	-	-
Можливість перегляду новин та заходів	-	+	-
Автономність	+	-	+
Зручний і простий інтерфейс	+	+	+
Сучасний дизайн	-	+	+
Перегляд основних відомостей	-	+	-
Функція обміну повідомленнями у вигляді простого чату або чат-боту	-	+	+
Розклад занять	+	+	+

Після опрацювання табл.1.1 можна дійти висновку, що жоден з обраних для аналізу мобільних додатків не має всі обрані функції. Найбільш оптимальним рішенням у цьому випадку є розробка власного мобільного додатку, який буде реалізовувати всі заплановані функції.

1.3 Огляд можливостей чат-ботів

З кожним днем штучний інтелект розвивається ще більше. Боти покликані зробити наше життя простішим, більш пізнавальним і цікавішим, а можливість машинного навчання робить їх багатообіцяючою технологією в галузі освіти [8]. База знань чат-ботів буде тільки зростати, а самі боти зможуть навчатися разом зі студентами.

При великій кількості діючих онлайн-сервісів в сегменті електронного навчання чат-боти можуть супроводжувати кожного слухача індивідуально, відповідно до його рівня і обраним темпом освоєння матеріалу, роблячи навчання доступним практично для будь-якої людини, що має доступ до Wi-Fi.

Останнім часом в месенджерах набирають популярність такі сервіси, як чат-боти. На сьогоднішній момент існує величезна різноманітність чат-ботів.

Деякі з можливих варіантів представлені нижче:

- ігрові чат-боти (квести, рольові ігри);
- рекламні чат-боти (Coca-Cola);
- чат-боти новин (Комерсант, Meduza);
- чат-боти для доставок, магазинів, сервісів (таксі «Максим», піцерія «Папа Джонс»)
- чат-боти для консультації і підтримки клієнтів (банк «Точка»).

Боти [9] - це програми, які виконують різні завдання для користувача, що знаходиться в месенджері. Бот виглядає як звичайний чат, однак спілкування відбувається не з людиною, а з програмою, яка може прийняти замовлення на

виклик машини, якщо це - бот таксі, або надіслати свіжі статті, якщо це - новинний бот, або замовити доставку їжі додому, якщо це - ресторанний бот.

Не так давно чат-боти здобули велику популярність, перевтілившись з розваги в більш серйозну річ, так як вони, в основному, стали використовуватися для вирішення серйозних бізнес-завдань. В епоху інформаційних технологій - це нормальне явище, а тим більше - мережі Інтернет. По-перше, чат-боти - це «платформи» для вирішення бізнес-завдань. По-друге, чат-бот - це програма, яка підтримує діалог з користувачем, вибираючи відповіді з бази даних: ви запитуєте, де пообідати і тут же отримуєте миттєву відповідь. Крім того, чат-боти виконують безліч корисних функцій у виконанні рутинних операцій, пошуку інформації, об'єднання даних, роботі з клієнтурою.

Чат-бот як віртуальний співрозмовник має базу знань, яка являє собою набори можливих питань користувача і відповідних їм відповідей. Найбільш поширеними варіантами для отримання потрібної відповіді є ключові слова, збіг фрази, збіг контексту. Завжди існують якісь прості і легкі по виконанню справи, на які не хочеться витратити час. Тут на допомогу завжди можуть прийти чат-боти. Тепер для збору інформації можна використовувати чат-ботів. Звичайно, це робиться за допомогою спілкування з людьми, але аж ніяк не завжди.

Чат - боти не зможуть замінити людину в аудиторії найближчим часом, але вже можуть взяти частину його робочого навантаження на себе. Перспективи для використання чат-ботів на всіх рівнях освіти практично безмежні.

Для чіткого розуміння сутності, переваг і недоліків окремих чат-ботів розглянуто їх класифікацію (табл. 1.2).

Таблиця 1.2 – Класифікація чат-ботів

Ознака класифікації	Вид чат-бота
Користувач	Персональний Бізнесовий
Інтерфейс	Кнопковий Текстовий
Доступ	Доданий в групу За підпискою Вбудований в діалог
Призначення	Комунікаційний Функціональний
Принцип роботи	Шаблонний Який навчається

Після аналізу всіх можливостей чат ботів можна виділити деякі їх переваги над сайтами:

- персональна увага до кожного клієнта;
- гнучкість і швидкість відповіді;
- не вимагають установки і авторизації;

Отже виявивши всі переваги та можливості чат – ботів було прийнято рішення, що для технікуму доцільною є розробка такого чат-боту.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Дослідивши інформаційну систему ППТ КІ СумДУ, можна зробити висновок, що для студентів та абітурієнтів становлять інтерес тільки офіційний сайт технікуму та різні групи в соціальних мережах. І якщо переглядати інформацію у групах через спеціальні мобільні додатки для соціальних мереж зручно, то сайт не адаптований для перегляду на мобільних пристроях.

Проаналізувавши вищесказане, ставиться завдання створити повноцінний зручний і швидкий мобільний додаток для студентів та абітурієнтів ППТ КІ СумДУ, який дозволив би їм отримувати актуальну та необхідну інформацію за допомогою розроблених інформативних сторінок та швидкого чат-бота.

Реалізований додаток надасть зручний доступ до таких важливих розділів:

1. Розклад занять.
2. Чат-бот.
3. Заміни в розкладі.
4. Розклад дзвінків.
5. Новини технікуму.
6. Спеціальності для вступу.
7. Основні відомості про технікум.
8. Відомості про додаток.

Чат-бот допоможе автоматизувати отримання інформації студентами та абітурієнтами, а також познайомитися з новим методом комунікації зі студентами за допомогою месенджера Telegram. Основною задачею даної роботи буде розробка та впровадження чат-бота в мобільний додаток. Даний бот має можливість виконувати такі функції:

- симуляція живого спілкування з користувачем;
- текстовий пошук інформації;

— голосовий пошук інформації.

Студент може в будь-який час і в будь-якому місці дізнатися необхідну йому інформацію. Інформація надається з офіційного сайту технікуму. Чат-бот надає таку довідкову інформацію користувачу:

— інформацію про матеріальну базу, історію технікуму, кадрове забезпечення;

— інформацію для абітурієнтів: програми вступних іспитів, спеціальності, вступ;

— рейтингові списки для отримання стипендії та банківські реквізити для оплати навчання;

— доступ до медіа даних: фото та відео матеріалів;

— контактну інформацію.

Відповідь чат-бота залежить від того, що введе користувач. На стандартні привітання бот так само повинен відповісти привітанням. Основний принцип чат-бота – надавати відповідь по ключовим словам для отримання швидкої відповіді, але якщо треба отримати більш широку відповідь, то бот може задавати уточнюючі запитання і розмова переростає в діалог.

Перед розробкою додатку треба визначити основні вимоги до функціональності та вибрати платформу для розробки. Додаток повинен забезпечувати студентів необхідною інформацією максимально швидко і в максимально зручній формі.

Кінцевий продукт буде являти повноцінний мобільний додаток, основними функціями якого є надання розкладу занять та основної довідкової інформації про технікум ПТ КІ СумДУ.

Цей додаток призначений не тільки для студентів ПТ КІ СумДУ, але може бути дуже корисним абітурієнтам для отримання інформації вступ та спеціальності. Студенти повинні оцінити роботу швидкого чат-бота, який допоможе технікуму йти в ногу з часом.

2.2 Алгоритм роботи чат-бота

Опрацювавши відповідний матеріал, можна зробити висновок, що переважна більшість чат-ботів мають спільну структуру. Щоб побудувати будь-який чат-бот треба розглянути їх загальну структуру (рис. 2.1).

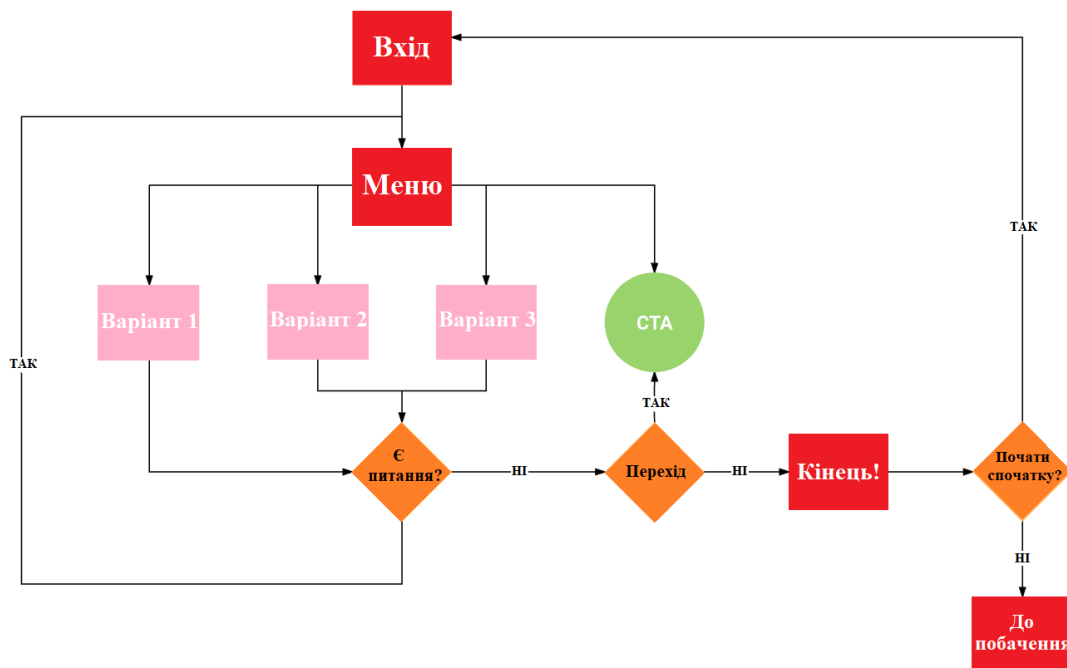


Рисунок 2.1 – Алгоритм роботи чат-бота

На рис. 2.1 можна побачити, як виглядає більшість чат-ботів зсередини. Кожен чат-бот починає вітати користувачів звичайною, реальною бесідою. Цей вітальний блок зазвичай включає в себе рекомендації по використанню бота (введіть що-небудь, натисніть кнопки, відправте голосові замітки). Далі переходимо до меню. Меню зазвичай надає користувачам деякі опції, які вони можуть вибрати, і додатковий Call To Action (СТА).

Заклик до дії (СТА) [10] - це запрошення на веб - сайті, в якому користувачеві пропонується виконати певну дію. Заклик до дії зазвичай записується у вигляді команди або фрази дії, наприклад "zareєstrуватися" або "купити зараз", і зазвичай приймає форму кнопки або гіперпосилання.

Переважна більшість чат-ботів мають чітку мету, і додавання СТА зазвичай є необхідним. Наприклад, якщо бот хоче, щоб ви підписалися на розсилку новин, він запропонує варіанти 1, 2 і 3, бот намагатиметься переконати вас, що підписатися це найкраще можливе рішення, але вам не потрібно проходити через ці варіанти, якщо ви вже ввійшли.

Після кожного блоку опцій бот пропонує користувачам вирішити, чи потрібна додаткова інформація, яка приведе їх до попереднього меню (приховуючи вже вибраний варіант). Якщо користувач не хоче більше ніякої інформації, діалог закінчується.

2.3 Вибір засобів реалізації мобільного додатку для ЗВО

Важливим фактором при створенні мобільного додатку служить платформа, під яку додаток буде розроблятися. На даний момент найпопулярнішими мобільними операційними системами є Android компанія Google та iOS компанія Apple [11]. Більш докладні дані наведені на рис. 2.2.

Одна з головних переваг платформи Android - її відкритість. Це дозволяє розробникам отримати доступ до вихідного коду Android і зрозуміти, яким чином реалізовані властивості і функції додатків. Будь-який користувач може взяти участь в удосконаленні операційної системи Android, для чого досить відправити звіт про виявлені помилки.

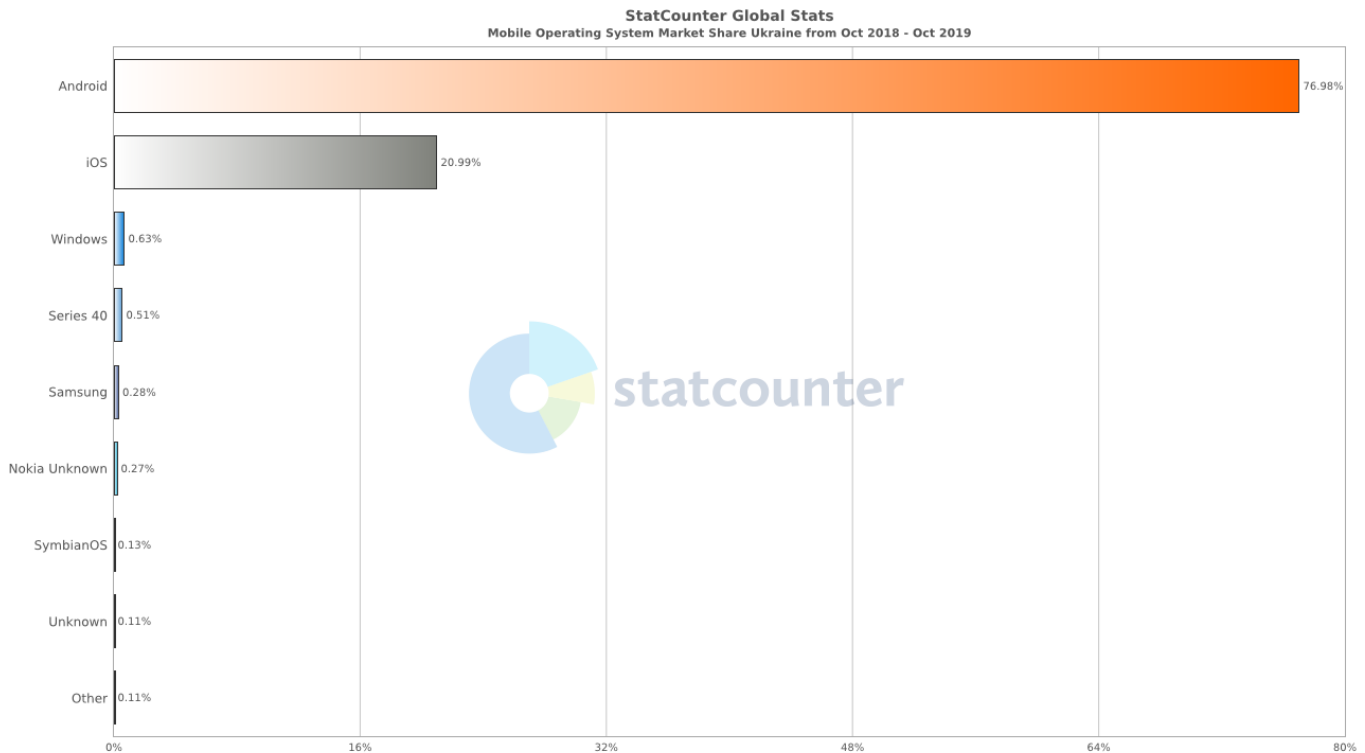


Рисунок 2.2 – Статистика мобільних операційних систем в Україні

Згідно з рис. 2.2 найпоширенішою мобільною операційною системою є Android. Отже, в результаті проведеного аналізу можна зробити висновок, що популярні мобільні платформи дуже схожі одна на одну, у зв'язку з чим, виділити вагомі переваги досить складно. Однак з огляду на більшу поширеність і відсутність необхідності початкових вкладень, для створення мобільного додатку був зроблений вибір на користь операційної системи Android.

Так як для реалізації даного проекту була обрана ОС Android, а офіційною мовою програмування для даної системи є Java, розглянемо середовища розробки, які доступні для даної операційної системи.

1. Eclipse [12] - це безкоштовне середовище розробки від некомерційної організації Eclipse Foundation. Завдяки високим темпам розвитку та підтримки з боку компанії можна виділити наступні переваги середовища Eclipse:

- можливість групової розробки;
- відмінна продуктивність на слабких машинах;
- велике число доповнень (для роботи з сервером, базою даних);

— можливість підключення модулів.

Хоча середовище розробки Eclipse мало велику популярність кілька років тому, після виходу Android Studio компанія Google перестала підтримку Eclipse як основне середовище для розробки мобільних додатків під Android.

2. IntelliJ IDEA. [13] Розробкою та підтримкою даного середовища програмування займається компанія JetBrains. Остання версія IntelliJ IDEA забезпечує значні поліпшення продуктивності і зручності використання, включаючи більш швидкий запуск, більш легку установку плагінів теми і клавіатури, поліпшені робочі процеси VCS, а також додає підтримку фреймворків мікросервісів, MongoDB і багато іншого. Як і Eclipse, це середовище розробки є кросплатформним. Головним недоліком даного IDE є наявність платної версії.

3. Android Studio [14] - це інтегроване середовище розробки (IDE) для роботи з платформою Android. Середовище Android Studio, засноване на програмному забезпеченні IntelliJ IDEA і є офіційним засобом розробки Android додатків на даний момент. Дане середовище розробки є кросплатформним і доступне на операційних системах Windows, OS X і Linux. Даним середовищем підтримується офіційна мова програмування для розробки додатків для платформи Android - Java, до того ж 17 травня 2017 року на Google I/O було оголошено про підтримку мови Kotlin в Android Studio 3.0, яка заслужила багато похвальних відгуків від досвідчених розробників.

За результатами проведеного аналізу було прийнято рішення використовувати офіційну середу розробки мобільних додатків для платформи Android - Android Studio.

Інтерфейс Android Studio [15] являє собою панель інструментів у верхній частині екрану, структуру проекту і консоль для виведення. Розробнику доступний текстовий редактор для роботи з програмним кодом, а для побудови графічного інтерфейсу додатку є можливість використовувати конструктор інтерфейсу. Файлами, що відповідають за візуальне відображення компонентів є XML-файли, які зручно редагувати вручну, тому робота з ними доступна так само і в текстовому

режимі. При редагуванні файлів інтерфейсу в режимі конструктора, праворуч стає доступна панель, що відображає властивості обраних об'єктів і їх значення, а також ієрархію і приналежність об'єктів.

Для розробки чат-ботів існує багато сервісів, але при виборі було розглянуто 2 сервіси, до яких в подальшому можна підключитися в Android Studio.

Dialogflow [16]

Dialogflow – це сервіс взаємодії людини і комп'ютера, заснований на розмові на природній мові. Це дозволяє розробникам додатків надавати своїм користувачам нові способи взаємодії з їх інтерфейсами через голосові і текстові обміни повідомленнями, що базуються на штучному інтелекті.

З допомогою Dialogflow розробники отримують доступ до вбудованих технологій машинного навчання та обробки природної мови. Це дозволяє їм зосередитися на інших невід'ємних частинах створення додатків, а не на детальному описі правил граматики.

Dialogflow також має вбудовані агенти, які дозволяють користувачам створювати свої додатки швидше. Додатково до всього, оновлений API.ai має вбудований редактор коду, що допомагає розробнику перевіряти роботу чат-бота прямо в консолі.

Dialogflow має такі переваги:

— Швидке кодування. Кодування не повинно займати багато часу. З Dialogflow, пов'язані з кодом завдання можуть бути виконані швидко, так як платформа має вбудований редактор коду. Це надає розробникам інструменти для зв'язку їх агентів з додатками через хмарні функції для Firebase.

— Машинне навчання. Dialogflow тепер підтримується технологіями машинного навчання Google. Це дає розробникам можливість навчати своїх агентів розуміти запити користувача шляхом вилучення даних з діалогу.

— Природні розмови. Хоча користувачі будуть в основному говорити з машиною і набором алгоритмів, як і раніше, важливо забезпечити їх природними розмовами, щоб вони відчували, що підтримка є персональною і

адаптованої. Розробники можуть зробити це за допомогою Dialogflow, так як платформа підтримує створення чат-ботів, які можуть вести природні розмови.

— Легка бесіда. З допомогою Dialogflow розробники можуть створювати алгоритми для своїх агентів, щоб зробити невелику розмову з користувачами. Платформа дозволяє їм визначати набір фраз для різних тем або ліній розмови, таких як емоції, підтвердження і багато іншого. Це дозволяє розробникам створювати інтерактивні діалоги з користувачами, які не пов'язані із запитом на обслуговування і запитом.

IBM Watson [17]

IBM Watson - це корпоративна платформа штучного інтелекту, яка надає компаніям необхідну потужність для прискорення досліджень, прогнозування збоїв. Платформа надає підприємствам повний контроль над тим, що має значення для бізнесу, і дозволяє їм зберігати право власності на свої дані, навчання, інтелектуальну власність та ідеї. Хмарна платформа IBM допомагає організаціям приймати більш обґрунтовані рішення і призначена для підприємств з різних секторів, включаючи: фінанси, охорону здоров'я, енергетику, взаємодію із споживачами, освіту, роботу і транспорт. Платформа здатна використовувати і розуміти різні типи даних, що дозволяє компаніям отримувати важливі і дієві відомості.

IBM Watson дозволяє проводити чіткі та більш швидкі дослідження і допомагає підприємствам відкривати нові можливості, використовуючи передові технології штучного інтелекту і машинного навчання.

Отже, після порівняння цих сервісів з розробки чат-ботів, перевагу було віддано сервісу Dialogflow. На відміну від IBM Watson, Dialogflow є безкоштовним сервісом та має підтримку української та російської мов. Також перевага Dialogflow над IBM Watson в тому, що не потрібно створювати новий акаунт користувача, а треба просто мати акаунт Google, який надає доступ сервісу Dialogflow.

В якості середовища для читання та збереження даних було обрано хмарну базу даних Firebase .

Firestore Realtime Database [18] – це хмарна база даних, що підтримує кілька платформ Android, iOS і Web. Всі дані зберігаються у форматі JSON, і будь-які зміни в даних негайно відображаються шляхом синхронізації на всіх платформах і пристроях. Це дозволяє легко створювати більш гнучкі додатки в реальному часі з мінімальними зусиллями. Дані синхронізуються між усіма клієнтами в реальному часі і залишаються доступними, коли база даних переходить в автономний режим.

Нижче наведені переваги використання бази даних Firestore в реальному часі проєкті:

— Realtime: дані, що зберігаються в базі даних Firestore Realtime, будуть відображені в реальному часі, тобто якщо відбудеться зміна значень в базі даних, то ця зміна буде відображена всім користувачам відразу.

— Велика доступність: база даних Firestore в реальному часі може бути доступна з різних платформ, таких як Android, iOS, Web. Таким чином, вам не потрібно писати той самий код для різних платформ кілька разів.

— Автономний режим: це найкраща перевага використання бази даних Firestore в реальному часі. Якщо ви не підключені до інтернету, і ви змінили щось в своєму додатку, то ця зміна буде відображена у вашому додатку відразу, але в базі даних Firestore, зміна буде оновлена, як тільки ви будете в мережі.

— Контроль доступу до даних: за замовчуванням нікому не дозволено редагувати дані у базі даних Firestore в реальному часі, але ви можете контролювати доступ до даних, тобто ви можете встановити, який користувач може отримати доступ до даних.

3 МОДЕЛЮВАННЯ МОБІЛЬНОГО ДОДАТКУ

3.1 Моделювання процесу розробки мобільного додатку

Функціональне моделювання (IDEF0) [19] - це метод моделювання бізнес-систем та інженерний підхід для аналізу потреб. Метод IDEF0 використовує поле для представлення функцій в рамках процесу і показує відносини з дочірніми і батьківськими системами. Цей метод забезпечує загальний план для розуміння систем організації.

Переваги:

- забезпечує покращений системний аналіз;
- забезпечує поліпшені методи зв'язку;
- допомагає в розумінні системи та її взаємозв'язків.

Ефективні моделі IDEF0 допомагають організувати аналіз системи і сприяють хорошій комунікації між аналітиком і клієнтом. IDEF0 корисний при визначенні обсягу аналізу, особливо для функціонального аналізу. Як комунікаційний інструмент IDEF0 посилює участь експертів у предметній області та прийняття консенсусних рішень за допомогою спрощених графічних пристроїв. В якості інструменту аналізу IDEF0 допомагає розробнику моделей визначити, які функції виконуються, що необхідно для виконання цих функцій, що поточна система робить правильно, а що поточна система робить неправильно. Таким чином, модель IDEF0 часто створюється як одне з перших завдань розробки системи.

Діаграму IDEF0 наведено на рис. 3.1. Дана діаграма демонструє основні процеси розробки програмного продукту. Так на вхід ми отримуємо потребу навчального закладу в програмному продукті, текстові документи для інформаційного наповнення, програмний модуль з розкладом навчального

процесу, технічне завдання для розробки чат-боту та технічну документацію Android, а на виході отримуємо програмний продукт у вигляді мобільного додатку.

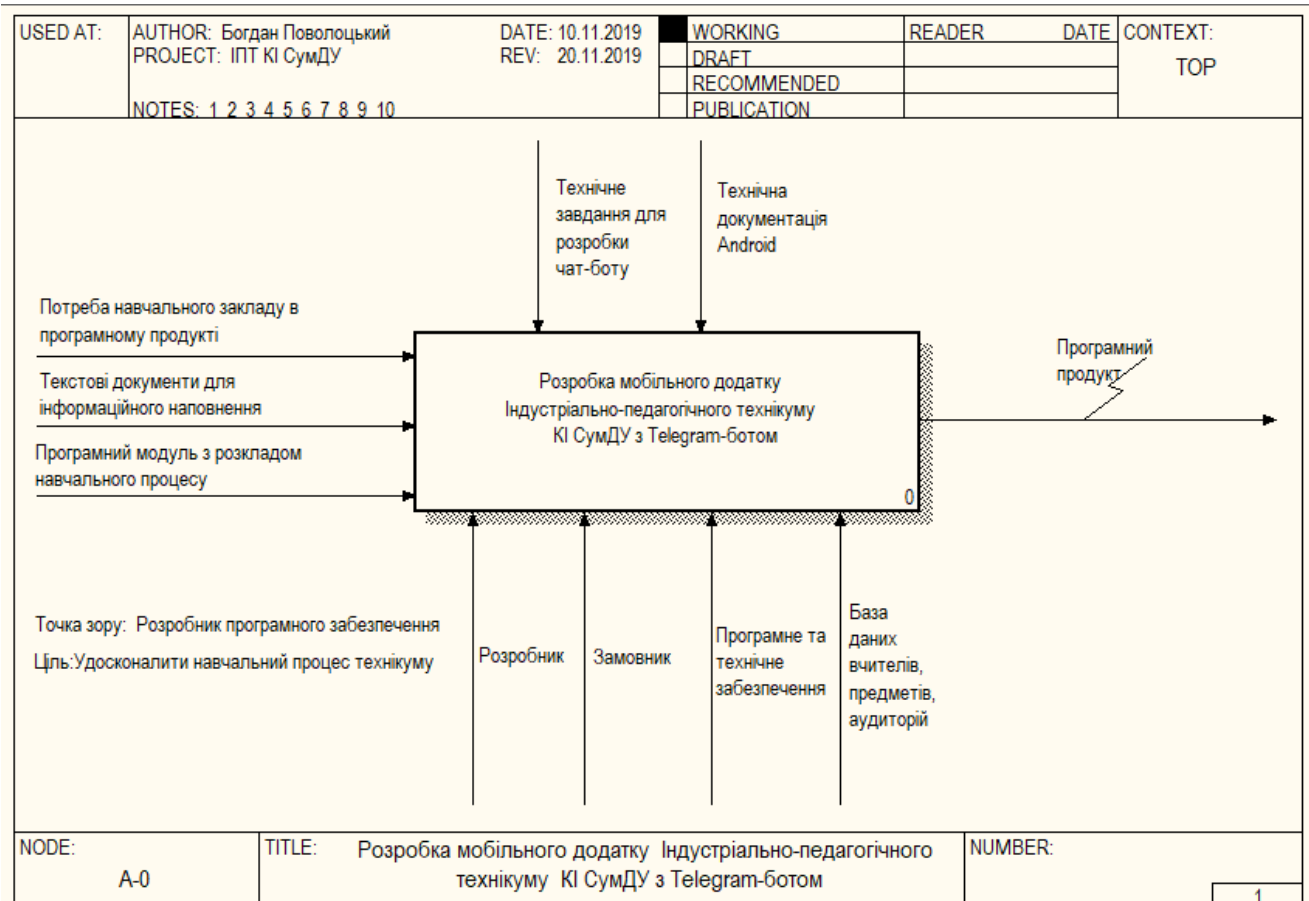


Рисунок 3.1 – Контекстна діаграма IDEF0

Декомпозиція першого рівня у нотації IDEF0 відображає взаємодію процесів розробки мобільного додатку (рис. 3.2).

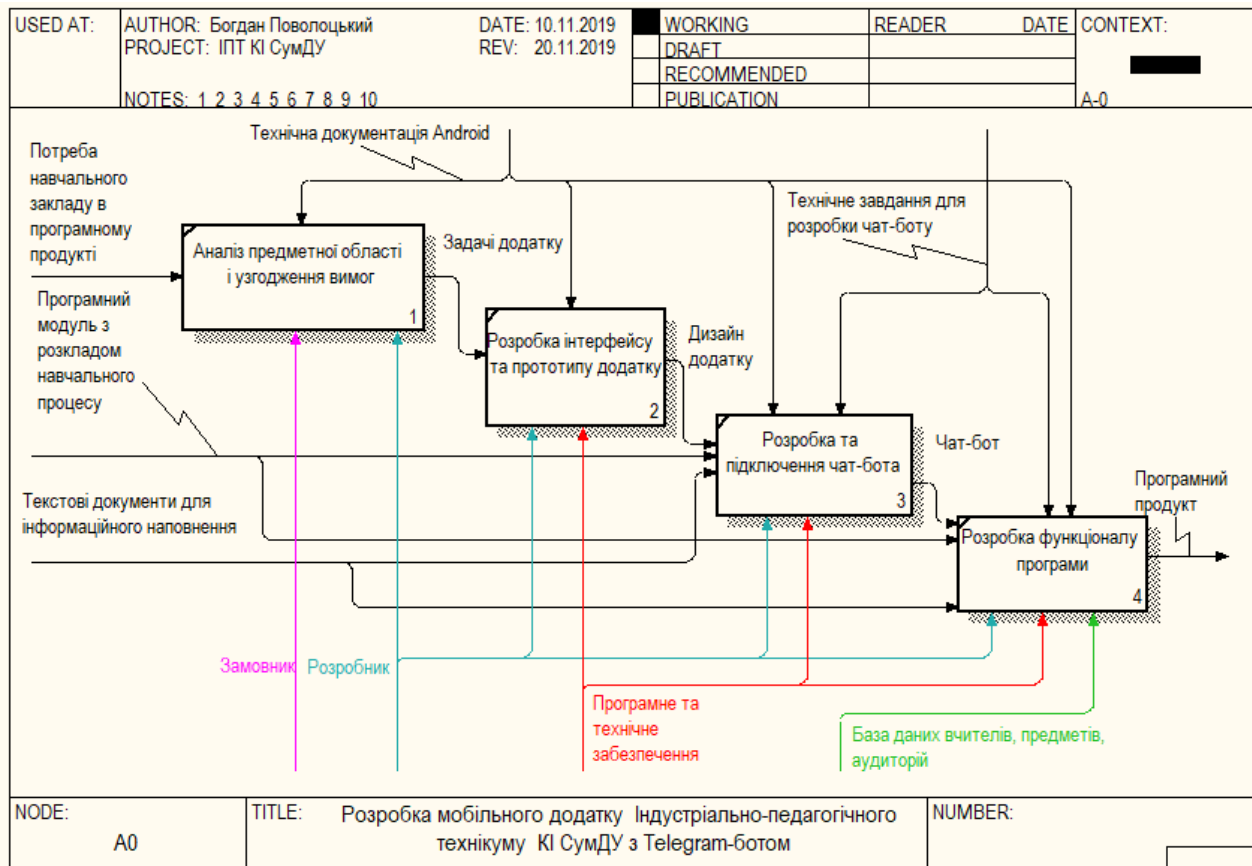


Рисунок 3.2 – Перший рівень декомпозиції IDEF0

Процес розробки мобільного додатку розділяється на такі функціональні блоки: аналіз предметної області і узгодження вимог, розробка інтерфейсу та прототипу додатку, розробка та підключення чат-бота, розробка функціоналу програми. Всі вищезгадані блоки зв'язані між собою, а виконання кожного блоку дає певний результат, який передається в інший блок. На виході отримуємо програмний продукт у вигляді мобільного додатку.

3.2 Моделювання роботи з мобільним додатком

Моделювання є важливим етапом при розробці будь-якого додатку, воно допомагає представити досліджуваний об'єкт та описати структуру. Перед розробкою будь-якого програмного продукту необхідно визначити вимоги до

нього, показати та зафіксувати вимоги зручно у формі діаграм варіантів використання (ВВ) програмного продукту.

Unified Modeling Language (UML) [20] – це мова моделювання загального призначення, яка використовується для візуалізації системи. Це графічна мова, яка є стандартною для індустрії програмного забезпечення для визначення, візуалізації, побудови та документування артефактів програмних систем, а також для бізнес-моделювання.

Переваги UML:

- спрощує складну конструкцію програмного забезпечення;
- зменшує тисячі слів пояснення в декількох графічних діаграмах, які можуть зменшити час, витрачений на розуміння.

Діаграми варіантів використання моделюють, як користувачі, які відображаються у вигляді фігур, званих "акторами", які взаємодіють з системою. Цей тип UML-діаграми повинен бути високорівневим оглядом відносин між учасниками та системами, тому він може бути відмінним інструментом для пояснення системи цільовій аудиторії.

Діаграму варіантів використання зображено на рис. 3.3

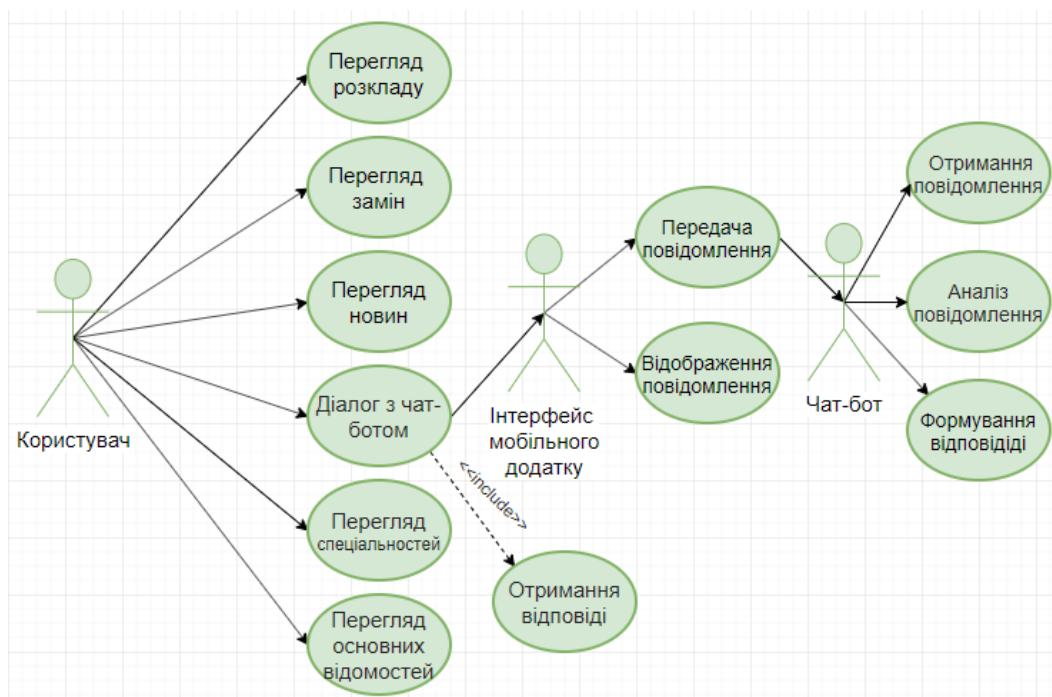


Рисунок 3.3 – Діаграма варіантів використання

Провівши аналіз було виділено наступні варіанти використання мобільного додатку:

Для користувача:

- ВВ1 – Перегляд розкладу
- ВВ2 – Перегляд замін
- ВВ3 – Перегляд новин
- ВВ4 – Діалог з чат-ботом
- ВВ5 – Перегляд спеціальностей
- ВВ6 – Перегляд основних відомостей.

Для інтерфейсу мобільного додатку:

- ВВ1 – Передача повідомлення
- ВВ2 – Відображення повідомлення

Для чат-бота:

- ВВ1 – Пересилання повідомлення для обробки
- ВВ2 – Формування повідомлення
- ВВ3 – Отримання відповіді.

Діаграма послідовностей UML [21] – це схема взаємодії, яка деталізує, як виконуються операції. Вони фіксують взаємодію між об'єктами в контексті спільної роботи. Діаграми послідовностей є тимчасовими і вони показують порядок взаємодії візуально, використовуючи вертикальну вісь діаграми для представлення часу, які повідомлення відправляються і коли.

Діаграма послідовностей зображує взаємодію [22] між об'єктами в послідовному порядку, тобто порядок, в якому ці взаємодії відбуваються. Діаграми послідовностей описують, як і в якому порядку об'єкти в системі функціонують. Ці схеми широко використовуються бізнесменами та розробниками програмного забезпечення для документування та розуміння вимог до нових та існуючих систем.

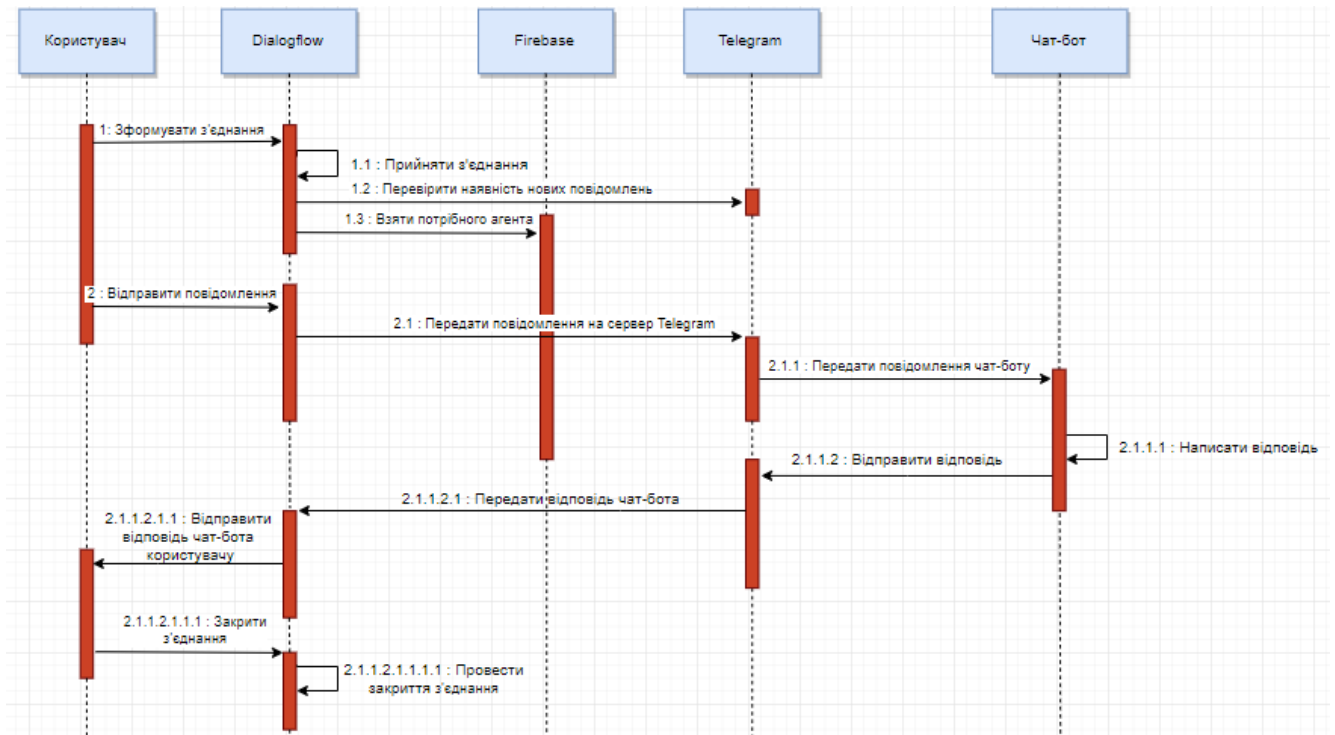


Рисунок 3.4 – UML діаграма послідовностей роботи користувача з чат-ботом мобільного додатку

На рис. 3.4 можна побачити, як працює програма. Спочатку користувач вводить якийсь текст, потім відбувається з'єднання з сервісом Dialogflow, де потрібний нам агент шукає в базі даних ключові слова, знайшовши їх дані зберігаються в хмарній базі даних Firebase, а через мобільний додаток чат-бот передає відповідь на екран смартфона.

3.3 Розробка макетів екранів мобільного додатку

Згідно визначення [23], дизайн – це діяльність по проектуванню естетичних властивостей промислових виробів, а також результат цієї діяльності.

При розробці макетів та дизайну додатку першим етапом є вибір цільової аудиторії. Далі варто продумати дизайн вікон додатку, розташування елементів інтерфейсу, їх стиль оформлення (колір, розмір, шрифт) тощо. Орієнтуючись на

цільову аудиторією, якою є студенти, абітурієнти та вчителі та навчальний заклад, для якого призначений даний мобільний додаток, треба сформулювати певні задачі та вимоги до дизайну.

Під час розробки макету екранів додатку були поставлені такі задачі:

- дотримуватися сучасних стилів дизайну, таких як Material Design;
- використати обмежену кількість кольорів, не більше 3-4;
- розробити простий, зручний інтуїтивно зрозумілий інтерфейс;
- адаптивність під різні розміри екранів;
- використати читабельні шрифти;

В результаті роботи створено макет екранів мобільного додатку. Він представлений на рис. 3.5 - 3.8.

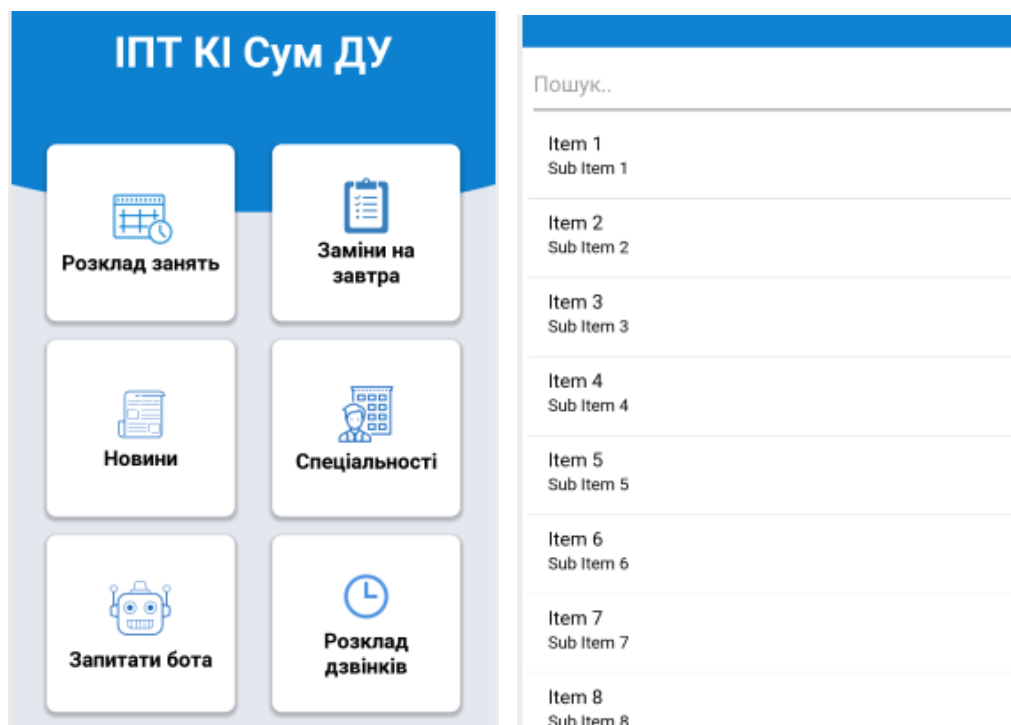


Рисунок 3.5 – Макети дизайну головного меню та вікна пошуку групи

Меню мобільного додатку складається з 6 пунктів: розклад занять, заміни на завтра, новини, спеціальності, запитати у бота, розклад дзвінків.

Після натискання на пункт меню «Розклад занять» користувач перейде на сторінку із списком груп, він може скористатися пошуком, щоб знайти потрібну групу або вибрати групу зі списку. Вибравши групу, користувач отримає актуальний розклад. Натиснувши пункт меню «Заміни на завтра», користувачу буде запропоновано переглянути актуальні заміни в розкладі на наступний день занять.

Пункти меню: розклад дзвінків, новини, спеціальності містять в собі інформаційні сторінки з відповідним інформаційним наповненням. Пункт меню новини відображає актуальні новини технікуму.

Пункт меню «Запитати у бота» відповідає за перехід на сторінку спілкування з ботом, де користувач у формі діалогу зможе отримати потрібну йому інформацію.

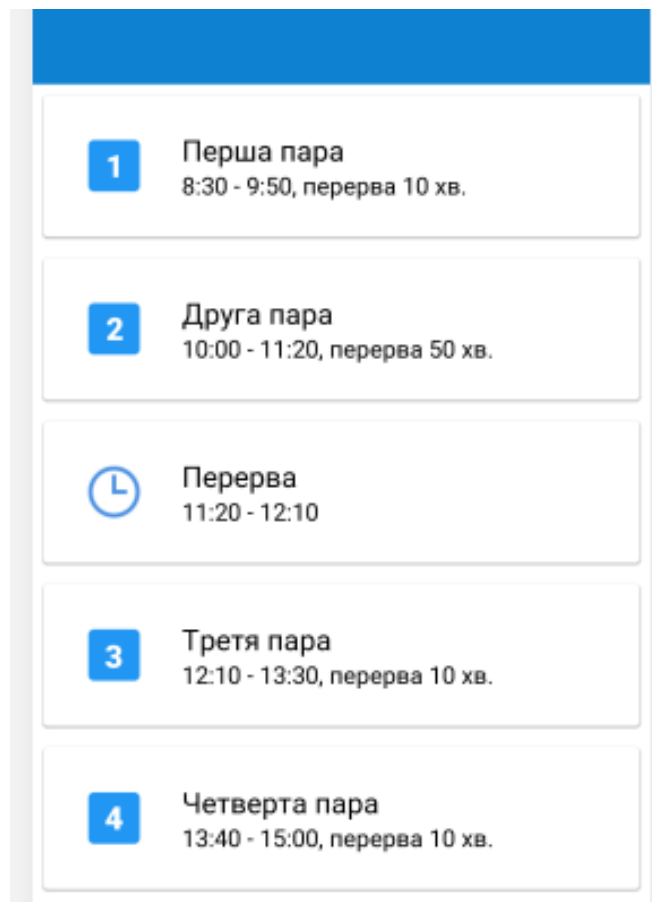


Рисунок 3.6 – Макет вікна розкладу дзвінків

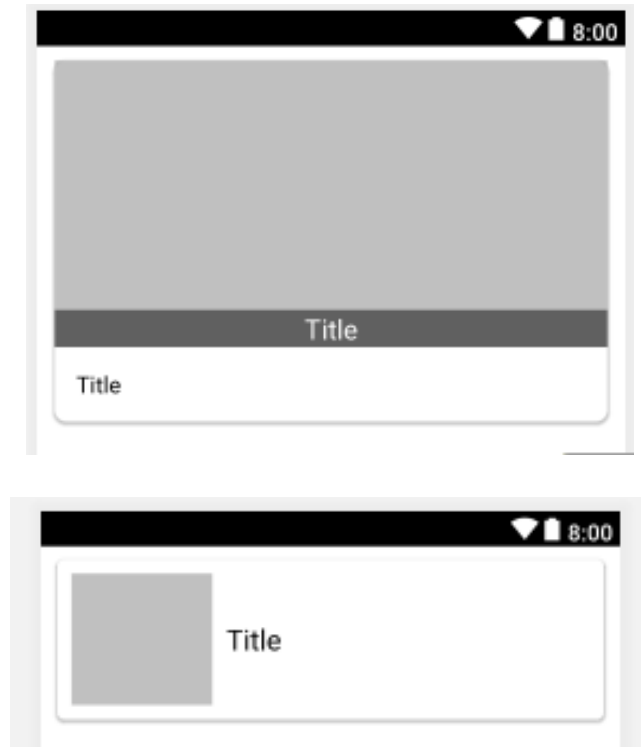


Рисунок 3.7 – Макети вікон новин та спеціальностей

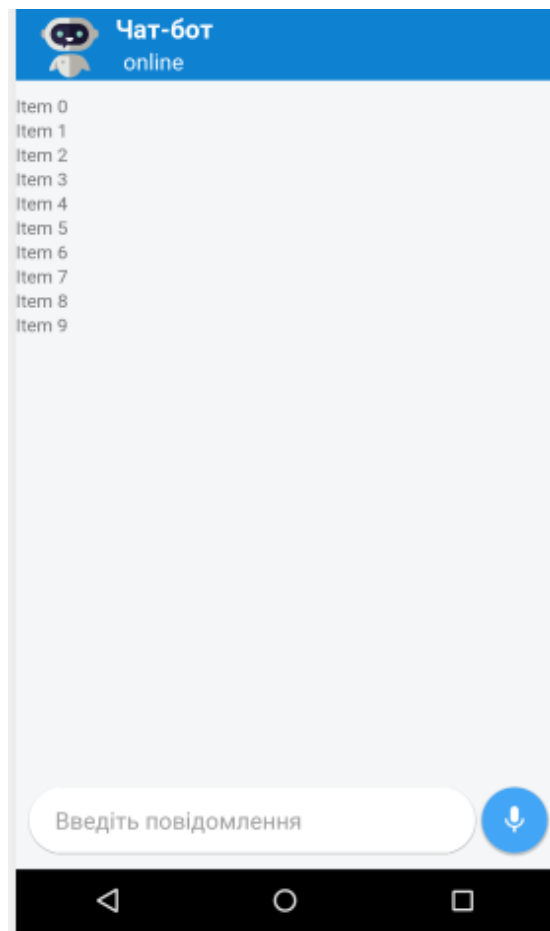


Рисунок 3.8 – Макет вікна чат-бота

4 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

4.1 Розробка чат-бота в сервісі Dialogflow

Щоб створити чат-бота, спочатку потрібно зайти на сайт Dialogflow [24] і натиснути на кнопку «Sign in» для авторизації. Маючи авторизований обліковий запис Google, можна увійти через нього.

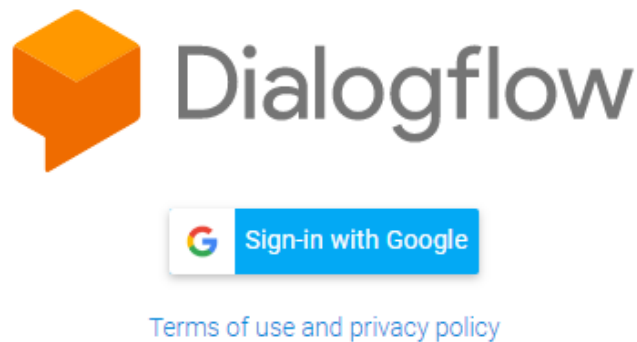


Рисунок 4.1 – Авторизація на сайті Dialogflow

Ввійшовши в систему, можна перейти до створення чат-бота. В наступному вікні треба дати ім'я боту та вибрати мову, на якій бот буде давати відповіді користувачу. Dialogflow постійно навчається і розпізнає текст з урахуванням особливостей, тому правильно вибравши мову, можна підвищити точність розпізнавання. Якщо мову не вибрати, бот все одно буде працювати, тільки стандартні відповіді будуть англійською, а не українською, і доведеться редагувати відповіді бота вручну. Тому краще вибрати відразу потрібну мову.

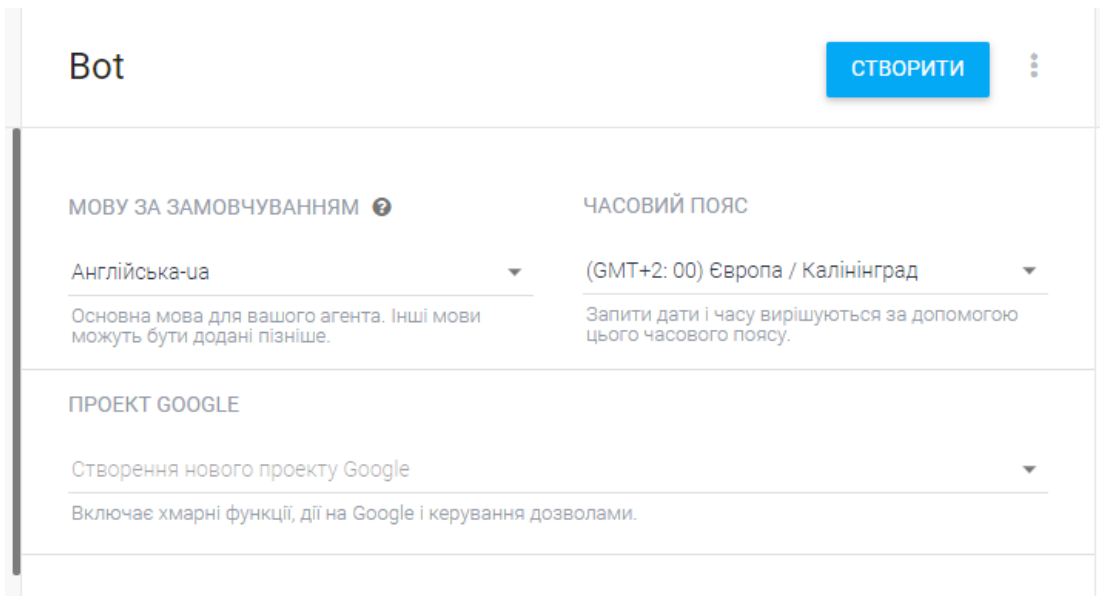


Рисунок 4.2 – Створення нового бота

Після створення бота можна побачити екран, на якому вже створено 2 intent: Default Fallback і Default Welcome. На цей момент бот уже вміє відповідати на «Привіт» і інші стандартні вітання і говорити, що він нас не розуміє, якщо жоден intent не підійшов. Intent – це співставлення фрази, яку ввів користувач з ключовими словами, на які реагуватиме бот для отримання відповіді.

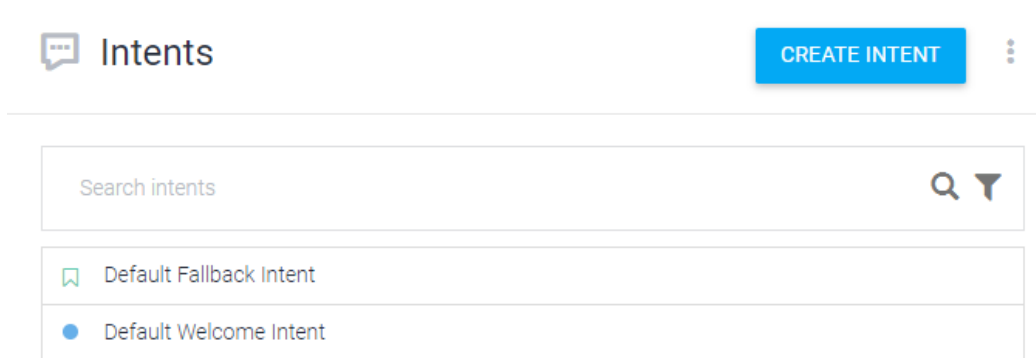


Рисунок 4.3 – Стандартні наміри

Протестувати бота можна, використовуючи веб-версію. Для цього треба вибрати розділ Integrations і там перший блок.

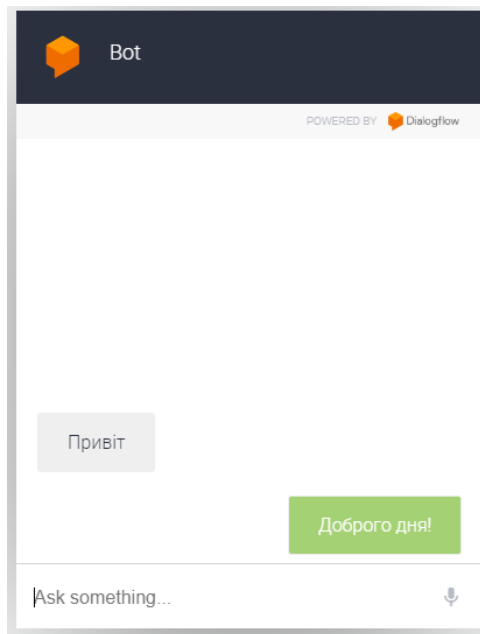


Рисунок 4.4 – Тестування роботи чат-бота у веб-версії

Для того, щоб налаштувати отримання відповідей, треба зайти в пункт меню Intent і налаштувати його під свої потреби. Для отримання бажаного результату треба між різними intent налаштовувати послідовності і взаємозв'язки.

Intent, який видає інформацію, якщо бот не розуміє співрозмовника, - Default Fallback Intent. Для його відкриття треба в меню повернутися на пункт Intents і натиснути на Default Fallback Intent.

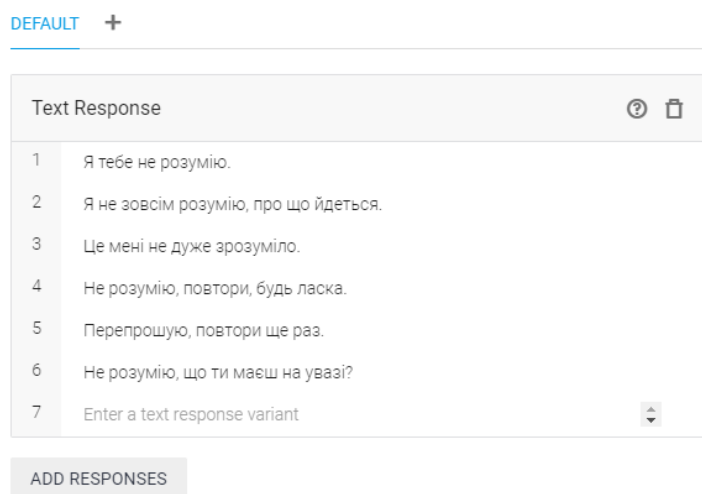


Рисунок 4.5 – Default Fallback Intent

Також в Dialogflow є Default Welcome Intent, який містить слова вітання з користувачем. При бажанні можна замінити тренувальні фрази, тобто, ключові слова, на які реагуватиме бот на свої та додати власні відповіді у вигляді привітань.

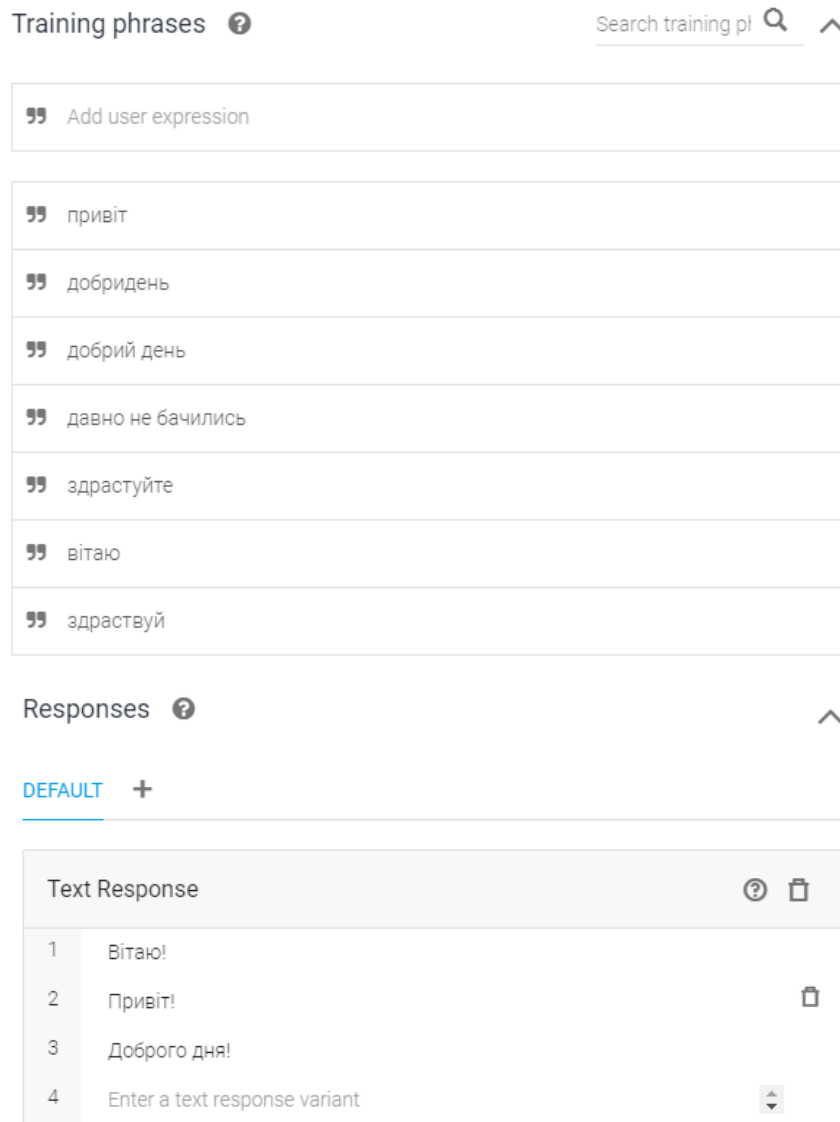


Рисунок 4.6 – Default Welcome Intent

Перейдемо до створення потрібних нам intents(намірів), для цього створюємо намір, задаємо його ім'я, ключові слова, на які реагуватиме чат-бот та відповіді. Створені в даному проєкті наміри показані на рисунку.

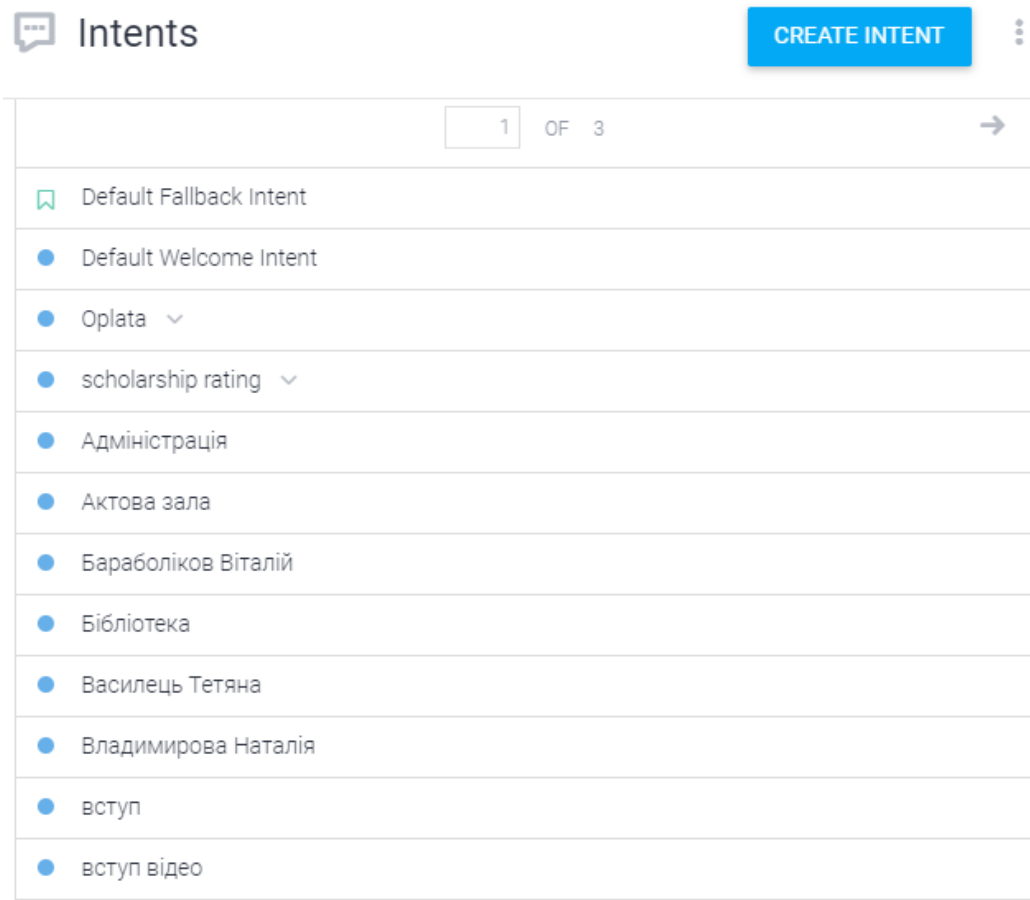


Рисунок 4.7 – Наміри в проекті

Dialogflow інтегрується з багатьма популярними платформами для спілкування, такими як Google Assistant, Facebook Messenger, Telegram та іншими.

4.2 Реалізація бази даних чат-бота

Для реалізації поставлених завдань в проекті була використана noSQL база даних Firebase. База даних Firebase Realtime дозволяє створювати програми, забезпечуючи безпечний доступ до бази даних безпосередньо з клієнтського коду. Дані зберігаються локально, навіть в автономному режимі. Коли пристрій відновлює з'єднання, база даних реального часу синхронізує локальні зміни даних з віддаленими оновленнями, які сталися під час автономної роботи клієнта,

автоматично оновлюючи всі дані. Щоб використовувати базу даних треба створити на сайті базу і в самому проекті Android Studio підключитися до неї. Структура бази даних представлена на рис. 4.8.

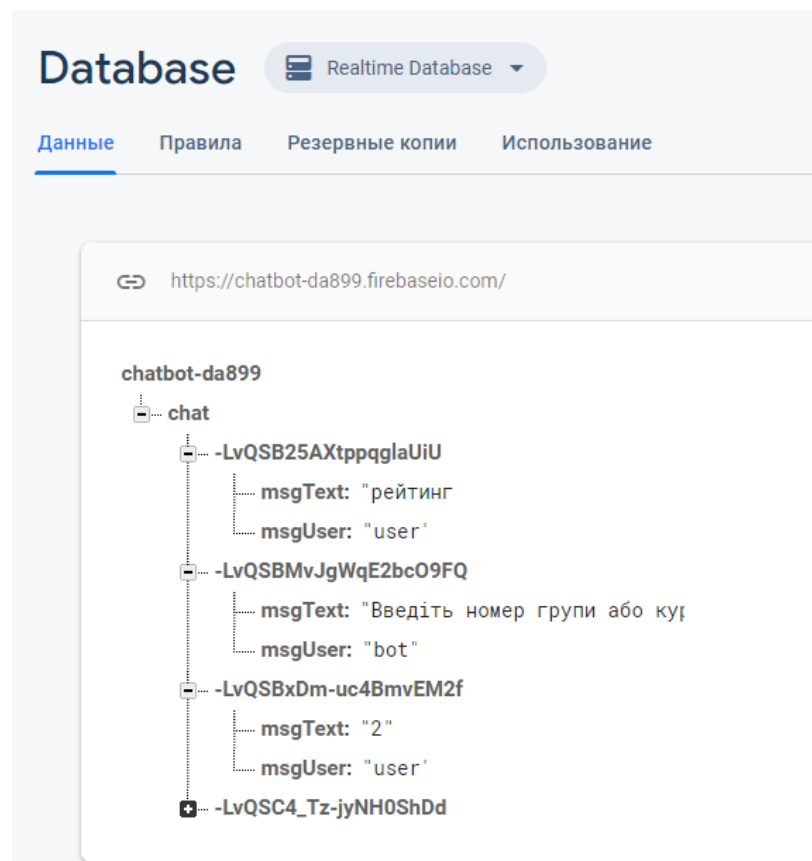


Рисунок 4.8 – Структура бази даних Firebase

База даних складається з вузла «chat», який містить в собі 2 поля:

- msgText – текстове поле, яке зберігає в собі текст повідомлення;
- msgUser – текстове поле, яке показує приналежність повідомлення користувачу чи боту.

На рисунку можна побачити, що дані заносяться в таблицю у формі діалогу. За допомогою коду в базі даних створюються відповідні поля для відображення діалогу між користувачем та чат-ботом. Після завершення сеансу дані з таблиці автоматично видаляються.

4.3 Розробка модулів програмного продукту

Створена таблиця 4.1, в якій вказані основні класи програми в Android Studio та їх призначення.

Таблиця 4.1 – Основні класи програми

Клас	Призначення
BotActivity	Клас призначений для підключення до бази даних Firebase та сервісу Dialogflow. Також для відображення діалогу між користувачем та чат-ботом
NewsActivity	Клас відповідає за парсинг списку новин з сайту технікуму
FirstActivity	Відображення головного меню додатку
SpecActivity	Клас відповідає за парсинг списку спеціальностей з сайту технікуму
DetailActivity	Клас для парсингу детальної інформації вибраної новини
DetailActivity1	Клас для парсингу детальної інформації вибраної спеціальності
DzvinkiActivity	Клас для відображення розкладу дзвінків

Кожен клас має набір методів, які відповідають за певну дію. За візуальне відображення інформації в класах відповідають файли розмітки, тобто .xml файли в папці layout, які викликають в розглянутих вище класах.

Таблиця 4.2 – Основні методи класів та їх призначення

№	Клас	Метод	Призначення
1	BotActivity	onTextChanged ()	Анімація зміни зображення на кнопці відправки повідомлення
2	BotActivity	populateViewHolder ()	Відображення діалогу користувача з ботом
3	DetailActivity	onPageStarted ()	Відображення спливаючого повідомлення
4	DetailActivity	onPageFinished ()	Завантаження html-сторінки
5	FirsrActivity	setSingleEvent ()	Обробка натискання на пункти головного меню, перехід на відповідну активність
6	NewsActivity	doInBackground ()	Парсинг списку новин з сайту технікуму
7	SpecActivity	doInBackground ()	Парсинг спеціальностей з сайту технікуму

Далі представлені вікна з результатами роботи мобільного додатку (рис. 4.9 – 4.14).

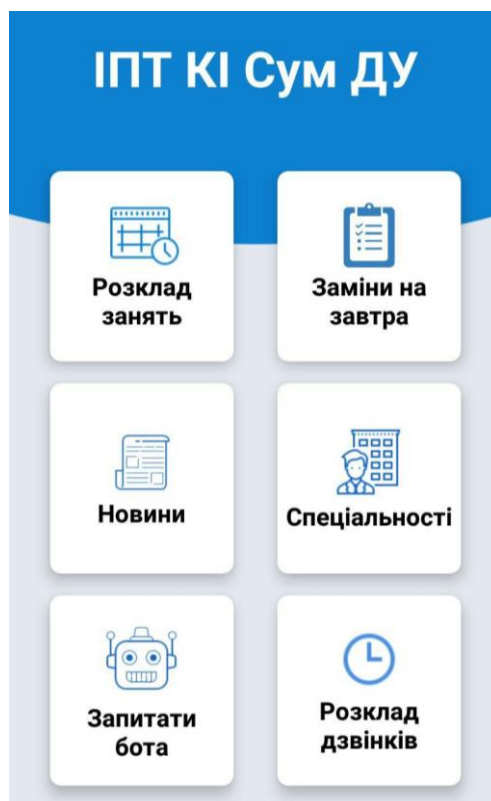


Рисунок 4.9 – Головне меню додатку

Група	№ пари	Назва предмета	Назва предмета	Аудиторія
111/311/811	4	-----	Географія	227
511/911	1	ЗВ	Географія	227
611/711	1	-----	УМ	213;215
	4	ЗВ	ВІ	112
322	2	Екологія	ІКГ	115
721	2	Математ	Культурол	319
	4	Фізика	-----	
121	1	-----	БК	218
	2	ФВ	Інжгеодез	218
	4	Психол	-----	
122	2	ВдоС	Інжгеодез	218
	3	Ін мова	ІКГ	317
321	3	ІКГ	МТКМ	115
	4	-----	Ін мова	221;217
421	4	-----	МТОМЗЗ	113
521/621	2	Оснправоз	Ін мова	318
821	2	ВдоС	УМ	213
	4	-----	ОФЗ	321
921	1	УМ	МТКМ	115
231	2	Ін мова	ЕППЦС	104
	3	-----	ОМВР	208
	4	-----	Педагогіка	208
131	3	ФВ	ОМВР	208
	4	-----	Педагогіка	208
731	1	ОС	ДМ	120

Рисунок 4.10 – Вікно відображення заміні

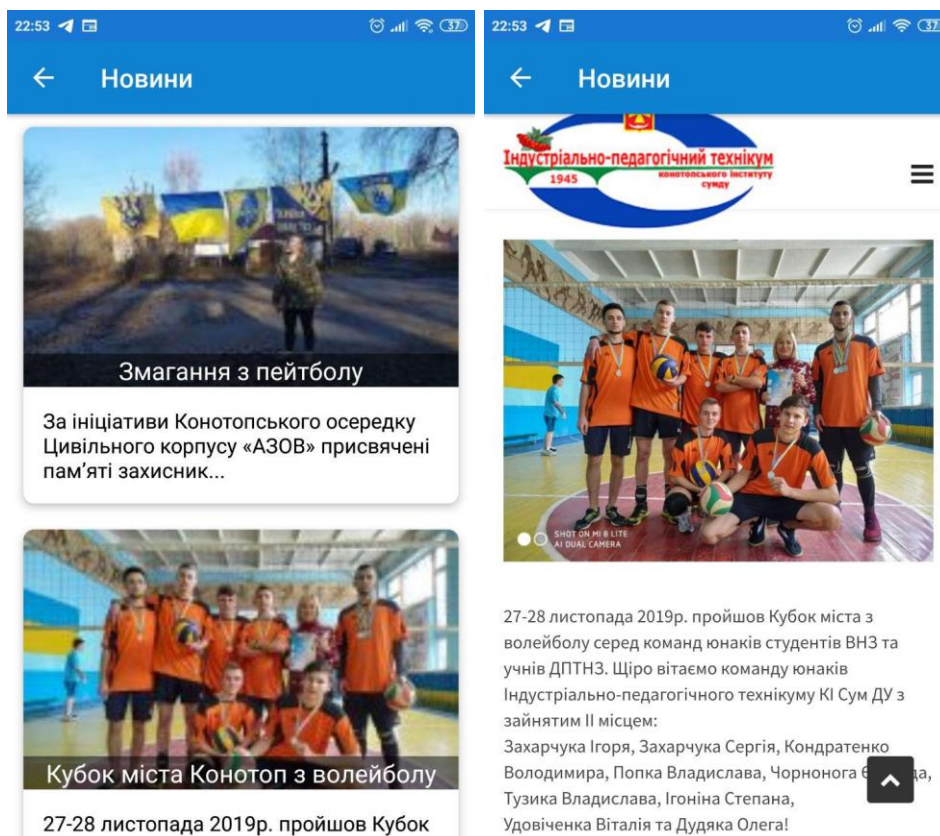


Рисунок 4.11 – Вікна перегляду списку новин та однієї новини

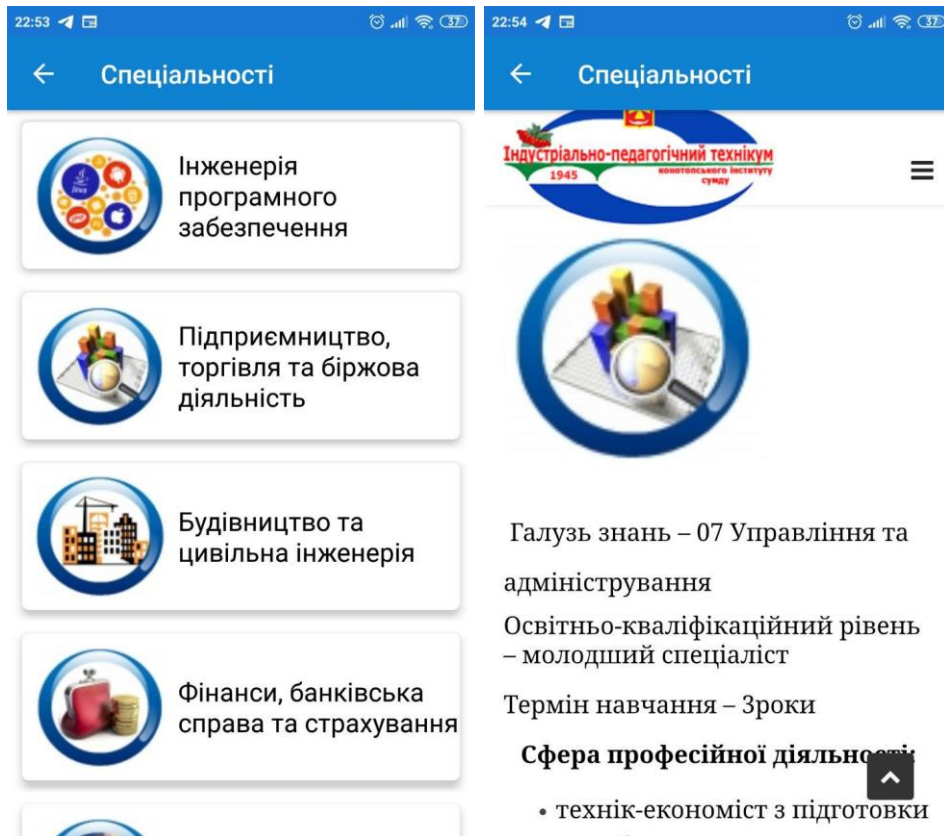


Рисунок 4.12 – Вікна перегляду списку спеціальностей та однієї спеціальності

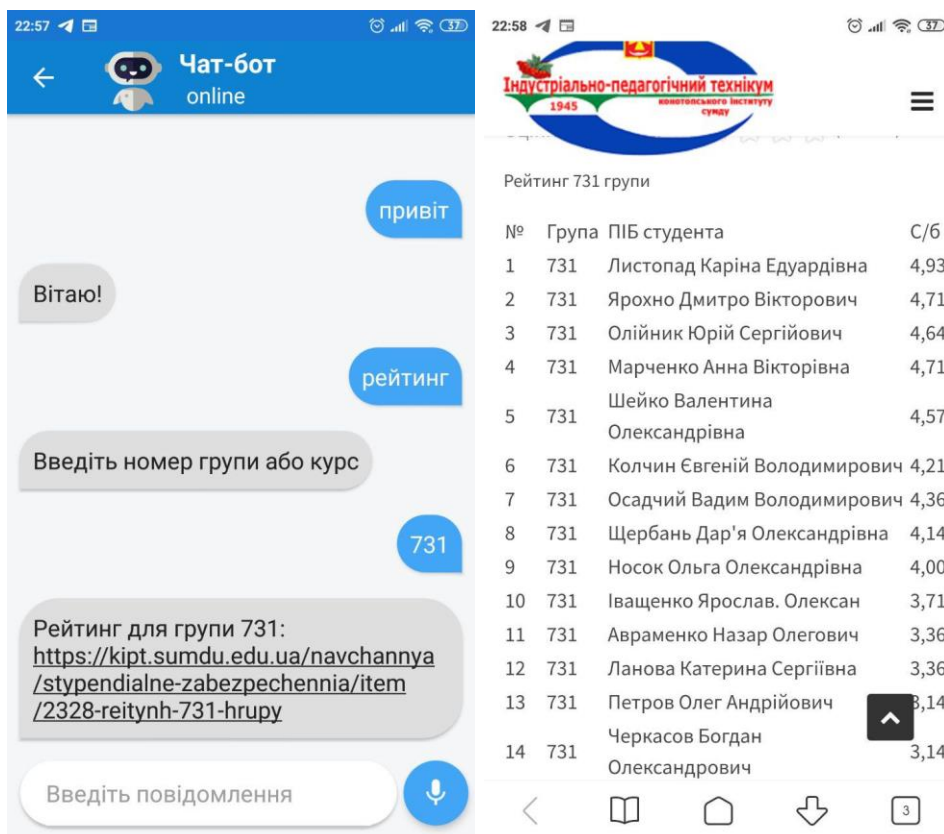


Рисунок 4.13 – Вікно діалогу з чат-ботом

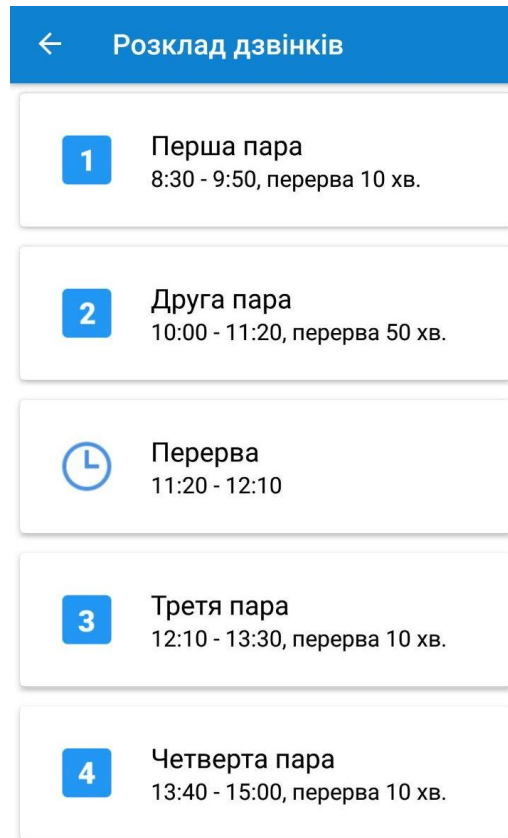


Рисунок 4.14 – Вікно відображення розкладу дзвінків

4.4 Тестування програмного продукту

Тестування програмного забезпечення [25] – це дослідження, проведене з метою надання зацікавленим сторонам інформації про якість досліджуваного програмного продукту або послуги.

Таким чином, розробивши мобільний додаток можна зробити тестування. Тестування додатку здійснювалося на власному мобільному пристрої з версією android 7.0, мінімальною версією для даного мобільного додатку є android 4.1.

У даній роботі було застосоване ручне тестування програмного продукту [26]. У табл. 4.3 було обрано та представлено набір Test Cases. Основним призначенням Test Cases є перевірка основного функціоналу програмного продукту.

Test Cases являє собою набір умов, за яких тестувальник має визначити відповідність заздалегідь визначеної вимоги очікуваному результату.

Таблиця 4.3 – Test Cases для програмного продукту

№	Test Case	Послідовність кроків	Очікуваний результат	Статус (пройдено/провалено)
1	Перевірка відображення новин, спеціальностей за відсутності інтернет з'єднання	Вибрати пункт головного меню Новини, Спеціальності	Повідомлення про помилку	Пройдено
2	Перевірка відповідності коректного переходу на потрібну активність	1. Натиснути на будь-який пункт головного меню 2. Перевірити відповідність	1. Перехід проходить коректно 2. Помилки не виникло	Пройдено
3	Перевірка працездатності кнопки назад	Натиснути на кнопку Назад в одному з вікон	1. Перехід проходить коректно 2. Помилки не виникло	Пройдено
4	Перевірка видалення повідомлень після завершення сеансу чату з ботом	1. Написати повідомлення чат-боту 2. Натиснути кнопку назад або вийти з програми	1. Відбувається видалення 2. Помилки не виникло	Пройдено

Таким чином, при тестуванні розробленого мобільного додатку помилок не виявлено. Можна помітити, що інформація, задана в масивах, відображається коректно. Так само переходи між активностями працюють без затримок. Всі фрагменти замінюються і видаляються, анімації працюють правильно.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра було проведено аналіз діяльності Індустріально-педагогічного технікуму КІСумДУ. Після проведення опису та аналізу поточних процесів управління діяльності Індустріально-педагогічного технікуму КІСумДУ та вибору цільової аудиторії, було прийнято рішення створити мобільний додаток з чат-ботом.

Наступним етапом було проаналізовано методи створення мобільних додатків з чат-ботами та обрані інструменти для виконання задач. Середовищем розробки мобільного додатку було обрано Android Studio, сервісом з розробки чат-ботів – Dialogflow, а в якості засобу зберігання та читання інформації виступила хмарна база даних Firebase.

Кінцевим результатом роботи над проектом є функціональний мобільний додаток з чат-ботом для полегшення навчального процесу студентів та абітурієнтів ІІТ КІ СумДУ. Додаток здатний виконувати наступні завдання: показувати розклад занять, показувати заміни в розкладі, показувати актуальні новини технікуму та спеціальності, відображати розклад дзвінків та довідкову інформацію. Окремим розділом даної роботи був розділ з розробкою функціонального чат-бота.

Даний бот має можливість виконувати такі функції:

- текстовий пошук інформації;
- симуляція живого спілкування з користувачем;
- голосовий пошук інформації.

Студент може в будь-який зручний для нього час дізнатися необхідну йому інформацію. Інформація надається з офіційного сайту технікуму.

Розроблений мобільний додаток з чат-ботом призначений полегшити та зробити процес отримання студентами інформації більш приємним та зручним. Додаток орієнтований на студентів та абітурієнтів технікуму, він допоможе технікуму підвищити свій імідж та зацікавленість абітурієнтів у вступі саме до цього технікуму.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технікум – [Електронний ресурс]. – Режим доступу: <https://uk.nure.info/blog/307-osvita-u-technikumi-ta-pislya-technikumu.html> (дата звернення: 17.09.2019).
2. Матеріальна база ІПТ КІ СумДУ – [Електронний ресурс]. – Режим доступу : <https://kipt.sumdu.edu.ua/pro-tehnikum/materialna-baza> (дата звернення: 18.09.2019).
3. Офіційний сайт Індустріально-педагогічного технікуму КІСумДУ – [Електронний ресурс]. – Режим доступу: <https://kipt.sumdu.edu.ua/> (дата звернення: 14.10.2019).
4. WordPress – [Електронний ресурс]. – Режим доступу : <https://wpprentice.com/blog/what-is-wordpress/> (дата звернення: 14.10.2019).
5. Мобільний університет – [Електронний ресурс]. – Режим доступу : <https://play.google.com/store/apps/details?id=ru.infocom.university> (дата звернення: 18.10.2019).
6. Мій університет – [Електронний ресурс]. – Режим доступу : <https://play.google.com/store/apps/details?id=com.involta.inuniver> (дата звернення: 18.10.2019).
7. Універ - розклад занять – [Електронний ресурс]. – Режим доступу : <https://play.google.com/store/apps/details?id=com.raspisaniyevuzov.app> (дата звернення: 18.10.2019).
8. Чат-боти в освіті – [Електронний ресурс]. – Режим доступу : <https://chatbotsmagazine.com/six-ways-a-i-and-chatbots-are-changing-education-c22e2d319bbf> (дата звернення: 19.10.2019).
9. Види чат-ботів – [Електронний ресурс]. – Режим доступу : https://www.hurekatek.com/blog/types-of-chatbots_ (дата звернення: 19.10.2019).

10. Call To Action – [Електронний ресурс]. – Режим доступу : <https://www.optimizely.com/optimization-glossary/call-to-action/> (дата звернення: 19.10.2019).
11. Статистика мобільних операційних систем в Україні – [Електронний ресурс]. – Режим доступу : <https://gs.statcounter.com/os-market-share/mobile/ukraine> (дата звернення: 25.10.2019).
12. Eclipse – [Електронний ресурс]. – Режим доступу : <https://www.osp.ru/os/2005/03/185394> (дата звернення: 25.10.2019).
13. IntelliJ IDEA – [Електронний ресурс]. – Режим доступу : <https://habr.com/company/JetBrains/blog/353494/> (дата звернення: 24.10.2019).
14. Android Studio – [Електронний ресурс]. – Режим доступу : <https://android-developers.googleblog.com/2018/03/android-studio-3-1.html> (дата звернення: 28.10.2019).
15. Інтерфейс Android Studio – [Електронний ресурс]. – Режим доступу : <https://developer.android.com/studio> (дата звернення: 25.10.2019).
16. Dialogflow – [Електронний ресурс]. – Режим доступу : <https://reviews.financesonline.com/p/dialogflow/> (дата звернення: 14.11.2019).
17. IBM Watson – [Електронний ресурс]. – Режим доступу : <https://reviews.financesonline.com/p/ibm-watson/> (дата звернення: 14.11.2019).
18. Firebase Realtime Database – [Електронний ресурс]. – Режим доступу : <https://firebase.google.com/docs/database/> (дата звернення: 16.11.2019).
19. Опис стандарту IDEF0 – [Електронний ресурс]. – Режим доступу : http://syque.com/quality_tools/tools/Tools19.htm. (дата звернення: 18.11.2019).
20. Діаграма варіантів використання мови UML – [Електронний ресурс]. – Режим доступу : <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/> (дата звернення: 18.11.2019).
21. Діаграма послідовностей – [Електронний ресурс]. – Режим доступу : <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/> (дата звернення: 19.11.2019).

- 22.Зв'язки між об'єктами в діаграмі послідовностей – [Електронний ресурс]. – Режим доступу : <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>_(дата звернення: 19.11.2019).
- 23.Дизайн – [Електронний ресурс]. – Режим доступу : <http://blog.awebdesign.studio/programmirovanie-i-dizajn-dve-storony-odnoj-medali/>. (дата звернення: 24.11.2019).
- 24.Сайт Dialogflow – [Електронний ресурс]. – Режим доступу : <https://dialogflow.com/> (дата звернення: 24.11.2019).
- 25.Тестування мобільних додатків – [Електронний ресурс]. – Режим доступу : <https://jetruby.com/ru/blog/testirovanie-mobilnyh-prilozhenii/>(дата звернення: 24.11.2019).
- 26.SMART – Милошевич Д. Набір інструментів для управління ІТ-проектами: підручник. – ДМК, 2006. – 674 с. (дата звернення: 24.11.2019).
- 27.WBS – [Електронний ресурс]. – Режим доступу : <https://www.workbreakdownstructure.com/> (дата звернення: 24.11.2019).
- 28.OBS – [Електронний ресурс]. – Режим доступу : <https://uplandsoftware.com/psa/resources/glossary/organization-breakdown-structure-obs/> (дата звернення: 24.11.2019).
- 29.Діаграма Ганта – [Електронний ресурс]. – Режим доступу : <https://www.gantt.com/> (дата звернення: 25.11.2019).
- 30.Аналіз ризиків в проєкті – [Електронний ресурс]. – Режим доступу : <http://um.co.ua/14/14-2/14-20214.html> (дата звернення: 24.11.2019).

ДОДАТОК А

Планування робіт

Ідентифікація ідеї проекту

У нашій країні, так і у всьому світі велика кількість закладів вищої освіти. В Індустріально-педагогічному технікумі КІ СумДУ є сайт, доступна можливість переглянути розклад занять та потрібну інформацію онлайн, але зручніше буде переглядати потрібну інформацію в мобільному додатку.

Надати можливість зручного перегляду інформації у будь-який час було вирішено за допомогою власного мобільного додатку, оскільки це дає гнучкі можливості модифікації та масштабування під власні потреби.

Ціллю проекту є створення мобільного додатку з чат-ботом, який допоможе полегшити процес отримання інформації студентами та абітурієнтами Індустріально-педагогічного технікуму КІСумДУ.

Мета створення додатку – забезпечити швидке та якісне надання актуальної інформації студентам та абітурієнтам технікуму. Дана розробка буде актуальна серед студентів та абітурієнтів, оскільки це значно заощадить їхній час.

Дане завдання є найбільш актуальним у ЗВО з великою чисельністю студентів, тому що це допоможе полегшити життя студентів та абітурієнтів.

В мобільному додатку планується розробка чат-бота, основною метою якого буде направити студентів до правильних джерел інформації, а також повідомлення важливих новин і доставка повідомлень протягом семестру. Також планується розробка отримання відповіді від чат-бота по певних категоріях, таких як: навчання, виховання, кадрове забезпечення і т.д.

Деталізація мети проекту методом SMART

Визначення мети проекту на основі показників абрєвіатури SMART [26] (конкретна (Specific), вимірювана (Measurable), досяжна (Achievable), реалістична (Relevant), обмежена у часі (Time-framed)) наведено у табл. А.1.

Таблиця А.1 – Деталізація мети методом SMART

Specific	Створення мобільного додатку з чат-ботом для студентів Індустріально-педагогічного технікуму КІСумДУ
Measurable	Створити якісний мобільний додаток, використовуючи наявні ресурси у обсязі, що не виходять за межі можливостей
Achievable	Створити мобільний додаток на основі реально доступних ресурсних можливостей
Relevant	У наявності є всі необхідні технічні та програмні засоби (Android Studio), доступ до інтернет
Time-bound	Мобільний додаток створюється з обмеженням у часі на основі сформованого календарного плану

Після проведення аналізу методом SMART можна визначити кінцеву мету одним реченням: розробити мобільний додаток з чат-ботом для відображення начального процесу технікуму, використовуючи наявні ресурси у обсязі, що не виходять за межі реально доступних апаратних та програмних можливостей, з обмеженням у часі на основі сформованого календарного плану.

Планування змісту структури робіт IT-проекту (WBS)

Структурна декомпозиція робіт Work Breakdown Structure (WBS) [27] є загальним терміном в управлінні проектами. Поділ складних проектів на більш прості і керовані задачі-це процес, який визначається як структура декомпозиції робіт (WBS).

Зазвичай менеджери проектів використовують цей метод для спрощення виконання проекту. У WBS набагато більші завдання розбиваються на керовані відрізки. Ці відрізки можна легко контролювати і оцінювати. Концепція використовується кожен раз, коли розробляється новий проект — і створюється до введення будь-яких даних в файл Microsoft Project. Тим не менш, незважаючи на те, що WBS є важливою конструкцією для менеджерів проектів, створення і використання WBS на етапі проектування проекту не відбувається з тією регулярністю, яку можна було б очікувати. WBS – це насправді просто візуальна розбивка проекту на більш дрібні компоненти – уявна ієрархія, яка спрощує планування (створення) необхідних результатів для будь-якої проектної команди.

Переваги проектування проекту, який включає в себе WBS:

- WBS допомагає визначити ключові результати і підкомпоненти результатів

до того, як робота запланована і запущена, що призводить до більш плавного розгортання під час проекту.

- WBS дає кожному чітке, візуальне уявлення про проект, без необхідності пробиратися через дрібниці і нудьгу інших типів метрик проекту або документації.

- Завдяки використанню схеми нумерації, WBS ідентифікує частини плану чисельно, які можуть бути використані в багатьох відносинах під час виконання вашого проекту.

WBS не обмежується конкретною галуззю, коли мова заходить про застосування. Ця методологія може бути використана для будь-якого типу управління проектами.

Нижче наведено кілька причин для створення WBS у проекті:

- чітка і зрозуміла організація проекту;
- чіткий розподіл обов'язків між проектною групою;
- вказує основні етапи проекту і контрольні точки;
- допомагає оцінити вартість, час і ризику.

Ефективність структури поділу робіт може визначати успішність проекту. WBS забезпечує основу для всієї роботи з управління проектами, включаючи планування, оцінку витрат і зусиль, розподіл ресурсів і планування.

Тому створення WBS слід розглядати як найважливіший крок у процесі управління проектами. Визначення основних результатів проекту є відправною точкою для одержання структури розбивки робіт. Будуємо WBS діаграму (рис. А.1).

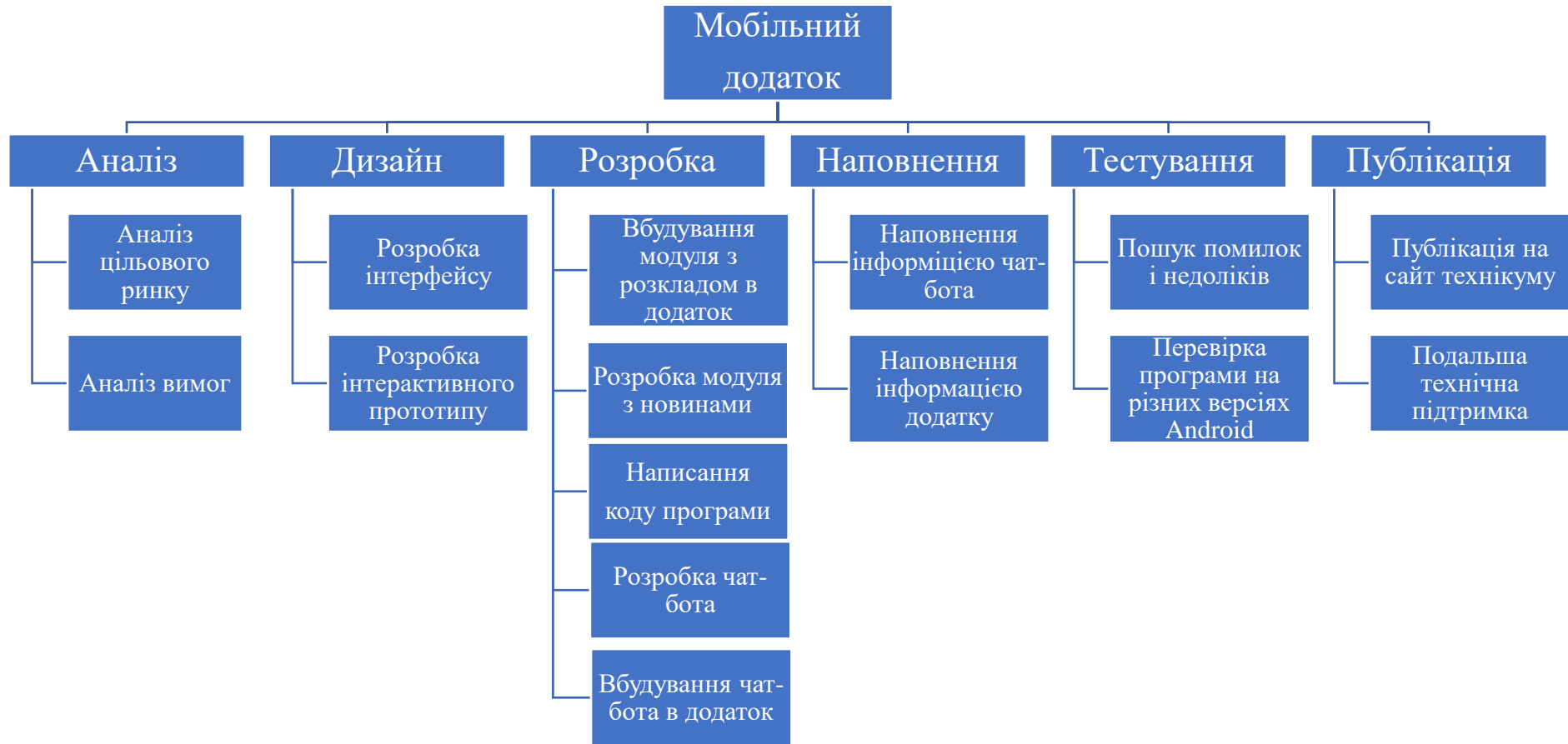


Рисунок А.1 – WBS діаграма розробки мобільного додатку

Організаційна структура проекту (OBS)

Структурна декомпозиція організації Organizational Breakdown Structure (OBS) [28] – це ієрархічна модель, що описує встановлену організаційну структуру для планування проектів, управління ресурсами, відстеження часу і витрат, розподілу витрат, звітності про доходи/прибутку і управління роботою. OBS відображає організаційні відносини і потім використовує їх для призначення ресурсів в проекті.

OBS дозволяє розбивати складні проекти, забезпечуючи більш організоване подання виконуваної роботи. У той час як WBS використовується для визначення проекту на ранніх стадіях його циклу, OBS забезпечує організаційну структуру для проекту у міру його просування до завершення. Ієрархічний характер організаційної структури дозволяє розподіляти відповідні ресурси та обов'язки.

Організаційна структура декомпозиції використовується в складних проектах і в поєднанні зі структурою декомпозиції робіт. Це дозволяє поглянути на організаційні ресурси, структуровані в ієрархічному порядку. OBS не орієнтована на вирішення завдань.

Для візуального представлення завдань в рамках проекту буде використовуватися структура декомпозиції робіт. Будуємо OBS діаграму (рис. А.2).

Будується OBS таким чином:

- на першому рівні відображається організаційна структура як єдиний елемент;
- на другому і нижчих рівнях ділиться нижча частина структури на основні організаційні елементи.

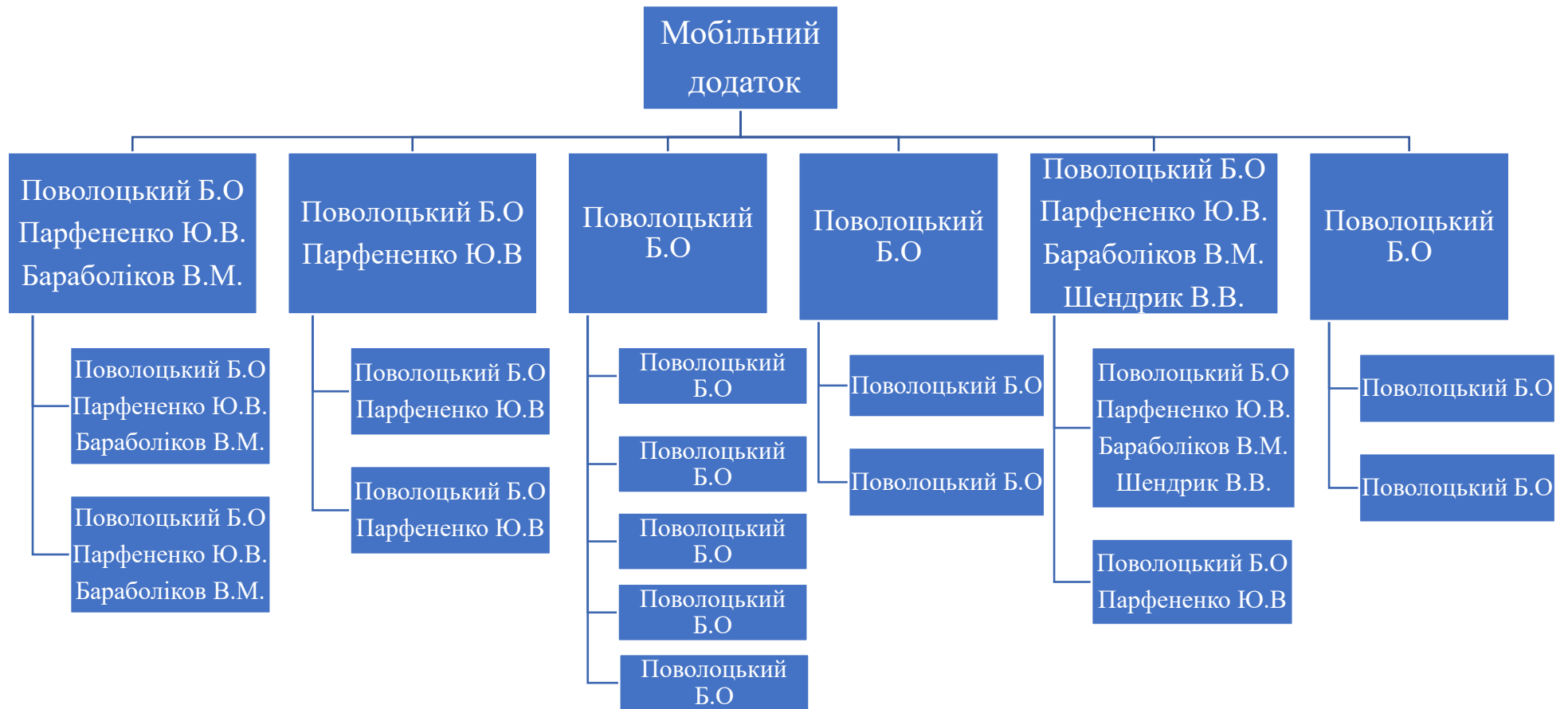


Рисунок А.2 – OBS діаграма розробки мобільного додатку

Побудова календарного графіка виконання ІТ – проекту

Діаграма Ганта [29] є одним з найбільш популярних і корисних способів відображення дій (завдань або подій), що відображаються у часі. Часто використовувана в управлінні проектами діаграма Ганта надає графічну ілюстрацію розкладу, яка допомагає планувати, координувати і відстежувати конкретні завдання в проекті. Зліва від графіка розташований список видів діяльності, а вгорі – відповідна шкала часу. Кожна дія представлено лінією, положення і довжина лінії відображає дату початку, тривалість і дату закінчення дії.

Діаграма визначає завдання, які можуть виконуватися паралельно, і ті, які не можуть бути запущені або завершені до завершення інших завдань. Діаграма Ганта може допомогти виявити потенційні важкі місця і визначити завдання, які могли бути виключені з графіка проекту.

На діаграмі показано час уповільнення виконання завдання або додатковий час для виконання завдання, яке не повинне затримувати проект, некритичні дії, які можуть бути затримані, і критичні дії, які повинні бути виконані вчасно.

Діаграму Ганта можна використовувати в управлінні проектами всіх розмірів і типів. Ця діаграма використовується у декількох галузях промисловості і для цілого ряду проектів, таких як розробка програмного забезпечення та розробка інших товарів і послуг.

Діаграма Ганта дає чітку ілюстрацію стану проекту, але її недолік полягає у тому, що вона не вказує залежність завдань - не можна сказати, як одна задача, яка відстає від розкладу, впливає на інші задачі.

За допомогою програми Microsoft Project Professional побудовано діаграму Ганта (рис. А.3).

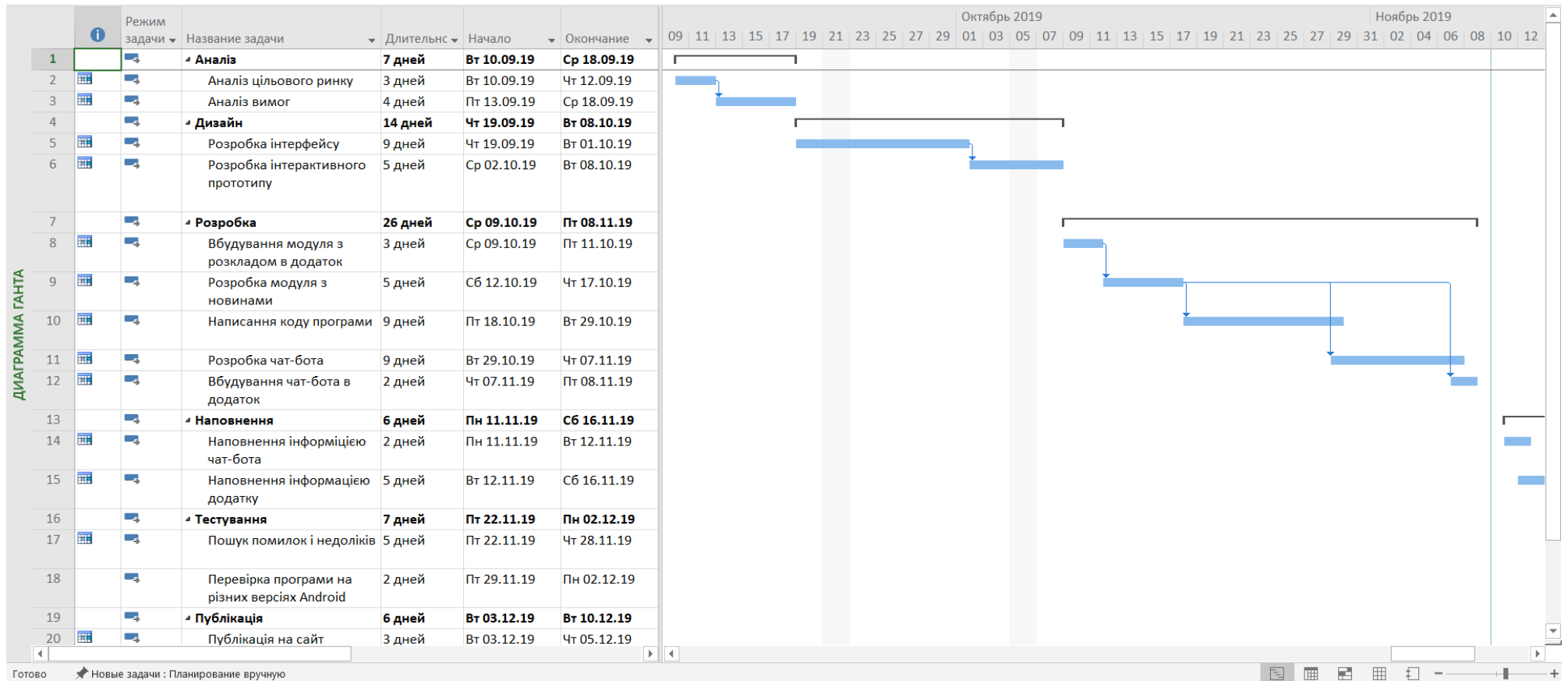


Рисунок А.3 – Діаграма Ганта розробки мобільного додатку

Управління ризиками

Вивчення ризиків [30], з якими може зустрітися проект із розробки мобільного додатку, показало, що можна виділити кілька десятків, але до основних ризиків розробки мобільного додатку і проекту загалом можна віднести наступні:

- збільшення навантаження під час реалізації проекту;
- зміна цілей у ході реалізації проекту;
- зміна строків виконання роботи;
- не знаходження спільної мови між керівником і виконавцем;
- зростання вимог до проекту;
- помилки у плануванні та проектуванні;
- виявлення помилок при функціональному тестуванні;
- відмова обладнання.

Повний список виділених у проекті ризиків представлений у табл. 1.2. Також там наведені оцінки вірогідності їх настання та очікуваний їх вплив на проект.

Таблиця 1.2. Вірогідність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Збільшення навантаження під час реалізації проекту	3	2
2	Зміна цілей у ході реалізації проекту	4	4
3	Зміна строків виконання роботи	2	4
4	Складнощі при впровадженні на місці	3	3
5	Зростання вимог до проекту	3	3
6	Помилки у плануванні та проектуванні	2	4
7	Виявлення помилок при функціональному тестуванні	2	3
8	Відмова обладнання	2	4

Матриця «Ймовірність – Втрати»

Ймовірність	-	5	5	10	15	20	25
	Зміна цілей у ході реалізації проекту	4	4	8	12	16	20
	Збільшення навантаження під час реалізації проекту Складнощі при впровадженні на місці Зростання вимог до проекту	3	3	6	9	12	15
	Відмова обладнання Зміна строків виконання роботи Помилки у плануванні та проектуванні Виявлення помилок при функціональному тестуванні	2	2	4	6	8	10
	-	1	1	2	3	4	5
			1	2	3	4	5

Збільшення навантаження під час реалізації проекту

Виявлення помилок при функціональному тестуванні
Складнощі при впровадженні на місці
Зростання вимог до проекту

Зміна цілей у ході реалізації проекту
Відмова обладнання
Зміна строків виконання роботи
Помилки у плануванні та проектуванні

Втрати

Аналізуючи ризики за ймовірністю їх виникнення, можемо їх розділити на:

— Ігноровані

1) Відсутні

— Незначні

1) Збільшення навантаження під час реалізації проекту

2) Відмова обладнання

3) Зміна строків виконання роботи

4) Виявлення помилок при функціональному тестуванні

— Помірні

- 1) Зростання вимог до проекту
- 2) Складнощі при впровадженні на місці
- 3) Помилки у плануванні та проектуванні

— Істотні

- 1) Зміна цілей у ході реалізації проекту

— Критичні

- 1) Відсутні

Класифікація ризиків за рівнем впливу:

— Прийнятні

- 1) Відсутні

— Виправдані

- 1) Збільшення навантаження під час реалізації проекту
- 2) Відмова обладнання
- 3) Зміна строків виконання роботи
- 4) Не знаходження спільної мови між керівником і виконавцем
- 5) Зростання вимог до проекту
- 6) Помилки у плануванні та проектуванні
- 7) Виявлення помилок при функціональному тестуванні

— Неприпустимі

- 1) Зміна цілей у ході реалізації проекту

ДОДАТОК Б

Лістинг

BotActivity.java :

```

package com.example.bogdan.scheduleiptkisumdu;
import android.Manifest;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import ai.api.AIDataService;
import ai.api.AIListener;
import ai.api.AIServiceException;
import ai.api.android.AIConfiguration;
import ai.api.android.AIService;
import ai.api.model.AIRequest;
import ai.api.model.AIResponse;
import ai.api.model.Result;

public class BotActivity extends AppCompatActivity implements AIListener {

    RecyclerView recyclerView;
    EditText editText;
    RelativeLayout addBtn;
    DatabaseReference ref;
    FirebaseRecyclerAdapter<ChatMessage,chat_rec> adapter;
    Boolean flagFab = true;
    Toolbar toolbar;

    private AIService aiService;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.bot_acrivity);

```

```

toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
toolbar.setNavigationOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v){
        /*Intent intent = new Intent(InfoActivity.this, MainActivity.class);
        startActivity(intent);*/
        onBackPressed();
    }
});

TypedValue typedValueColorPrimaryDark = new TypedValue();
BotActivity.this.getTheme().resolveAttribute(R.attr.colorPrimaryDark, typedValueColorPrimaryDark, true);
final int colorPrimaryDark = typedValueColorPrimaryDark.data;
if (Build.VERSION.SDK_INT >= 21) {
    getWindow().setStatusBarColor(colorPrimaryDark);
}

ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.RECORD_AUDIO},1);

recyclerView = (RecyclerView)findViewById(R.id.recyclerView);
editText = (EditText)findViewById(R.id.editText);
addBtn = (RelativeLayout)findViewById(R.id.addBtn);

recyclerView.setHasFixedSize(true);
final LinearLayoutManager layoutManager = new LinearLayoutManager(this);
linearLayoutManager.setStackFromEnd(true);
recyclerView.setLayoutManager(linearLayoutManager);

ref = FirebaseDatabase.getInstance().getReference();
ref.keepSynced(true);

final AIConfiguration config = new AIConfiguration("f73a25164ca14654ac9ee9f19db95818",
    AIConfiguration.SupportedLanguages.Ukrainian,
    AIConfiguration.RecognitionEngine.System);

aiService = AIService.getService(this, config);
aiService.setListener(this);

final AIDataService aiDataService = new AIDataService(config);

final AIRequest aiRequest = new AIRequest();

addBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        String message = editText.getText().toString().trim();

        if (!message.equals("")) {

            ChatMessage chatMessage = new ChatMessage(message, "user");
            ref.child("chat").push().setValue(chatMessage);

            aiRequest.setQuery(message);
            new AsyncTask<AIRequest,Void,AIResponse>(){

                @Override
                protected AIResponse doInBackground(AIRequest... aiRequests) {
                    final AIRequest request = aiRequests[0];

```

```

        try {
            final AIResponse response = aiDataService.request(aiRequest);
            return response;
        } catch (AIServiceException e) {
        }
        return null;
    }
    @Override
    protected void onPostExecute(AIResponse response) {
        if (response != null) {

            Result result = response.getResult();
            String reply = result.getFulfillment().getSpeech();
            ChatMessage chatMessage = new ChatMessage(reply, "bot");
            ref.child("chat").push().setValue(chatMessage);
        }
    }
    }.execute(aiRequest);
}
else {
    aiService.startListening();
}

editText.setText("");

}
});

editText.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        ImageView fab_img = (ImageView)findViewById(R.id.fab_img);
        Bitmap img = BitmapFactory.decodeResource(getResources(),R.drawable.ic_send_white_24dp);
        Bitmap img1 = BitmapFactory.decodeResource(getResources(),R.drawable.ic_mic_white_24dp);

        if (s.toString().trim().length() != 0 && flagFab){
            ImageViewAnimatedChange(BotActivity.this,fab_img,img);
            flagFab=false;
        }
        else if (s.toString().trim().length() == 0){
            ImageViewAnimatedChange(BotActivity.this,fab_img,img1);
            flagFab=true;
        }
    }
});

    @Override
    public void afterTextChanged(Editable s) {

    }
});

    adapter = new FirebaseRecyclerAdapter<ChatMessage,
chat_rec>(ChatMessage.class,R.layout.msglist,chat_rec.class,ref.child("chat")) {

```

```

@Override
protected void populateViewHolder(chat_rec viewHolder, ChatMessage model, int position) {

    if (model.getMsgUser().equals("user")) {

        viewHolder.rightText.setText(model.getMsgText());

        viewHolder.rightText.setVisibility(View.VISIBLE);
        viewHolder.leftText.setVisibility(View.GONE);
    }
    else {
        viewHolder.leftText.setText(model.getMsgText());

        viewHolder.rightText.setVisibility(View.GONE);
        viewHolder.leftText.setVisibility(View.VISIBLE);
    }
}
};

adapter.registerAdapterDataObserver(new RecyclerView.AdapterDataObserver() {
    @Override
    public void onItemRangeInserted(int positionStart, int itemCount) {
        super.onItemRangeInserted(positionStart, itemCount);

        int msgCount = adapter.getItemCount();
        int lastVisiblePosition = layoutManager.findLastCompletelyVisibleItemPosition();

        if (lastVisiblePosition == -1 ||
            (positionStart >= (msgCount - 1) &&
             lastVisiblePosition == (positionStart - 1))) {
            recyclerView.scrollToPosition(positionStart);
        }
    }
});

recyclerView.setAdapter(adapter);
}
public void ImageViewAnimatedChange(Context c, final ImageView v, final Bitmap new_image) {
    final Animation anim_out = AnimationUtils.loadAnimation(c, R.anim.zoom_out);
    final Animation anim_in = AnimationUtils.loadAnimation(c, R.anim.zoom_in);
    anim_out.setAnimationListener(new Animation.AnimationListener()
    {
        @Override public void onAnimationStart(Animation animation) {}
        @Override public void onAnimationRepeat(Animation animation) {}
        @Override public void onAnimationEnd(Animation animation)
        {
            v.setImageBitmap(new_image);
            anim_in.setAnimationListener(new Animation.AnimationListener() {
                @Override public void onAnimationStart(Animation animation) {}
                @Override public void onAnimationRepeat(Animation animation) {}
                @Override public void onAnimationEnd(Animation animation) {}
            });
            v.startAnimation(anim_in);
        }
    });
    v.startAnimation(anim_out);
}
}

```

```

@Override
public void onResult(ai.api.model.AIResponse response) {

    Result result = response.getResult();

    String message = result.getResolvedQuery();
    ChatMessage chatMessage0 = new ChatMessage(message, "user");
    ref.child("chat").push().setValue(chatMessage0);

    String reply = result.getFulfillment().getSpeech();
    ChatMessage chatMessage = new ChatMessage(reply, "bot");
    ref.child("chat").push().setValue(chatMessage);

}

@Override
public void onDestroy(){
    //ref = FirebaseDatabase.getInstance().getReference().child("chat"); ref.removeValue();
    FirebaseDatabase.getInstance().getReference().child("chat").removeValue();
    super.onDestroy();
}

@Override
public void onError(ai.api.model.AIError error) {
}

@Override
public void onAudioLevel(float level) {

}

@Override
public void onListeningStarted() {
}

@Override
public void onListeningCanceled() {
}

@Override
public void onListeningFinished() {
}
}

```

DetailActivity.java :

```

public class DetailActivity extends AppCompatActivity {

    ProgressBar progressBar;
    Toolbar toolbar;
    WebView webview;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        webview = (WebView) findViewById(R.id.detailView);
        webview.setVisibility(View.INVISIBLE);
        webview.getSettings().setJavaScriptEnabled(true);
        webview.setWebChromeClient(new WebChromeClient());
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        toolbar.setNavigationOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v){
    /* Intent intent = new Intent(DzvinkiActivity.this, MainActivity.class);
    startActivity(intent);*/
    onBackPressed();
}
});
TypedValue typedValueColorPrimaryDark = new TypedValue();
DetailActivity.this.getTheme().resolveAttribute(R.attr.colorPrimaryDark, typedValueColorPrimaryDark, true);
final int colorPrimaryDark = typedValueColorPrimaryDark.data;
if (Build.VERSION.SDK_INT >= 21) {
    getWindow().setStatusBarColor(colorPrimaryDark);
}
webView.setWebViewClient(new WebViewClient(){

@Override
public void onPageStarted(WebView view, String detailUrl, Bitmap favicon) {
    super.onPageStarted(view, detailUrl, favicon);
    Toast.makeText(DetailActivity.this, "Завантаження сторінки...", Toast.LENGTH_SHORT).show();
}

@Override
public void onPageFinished(WebView view, String detailUrl) {
    super.onPageFinished(view, detailUrl);
    progressBar.setVisibility(View.GONE);
    webView.setVisibility(View.VISIBLE);
    //Toast.makeText(DetailActivity.this, "Page Loaded", Toast.LENGTH_SHORT).show();
}
});
webView.loadUrl(getIntent().getStringExtra("detailUrl"));
}
}

```

NewsActivity.java :

```

public class NewsActivity extends AppCompatActivity {

private RecyclerView recyclerView;
private ParseAdapter adapter;
private ArrayList<ParseItem> parseItems = new ArrayList<>();
private ProgressBar progressBar;
Toolbar toolbar;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_news);

    progressBar = findViewById(R.id.progressBar);
    recyclerView = findViewById(R.id.recyclerView);

    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    adapter = new ParseAdapter(parseItems, this);
    recyclerView.setAdapter(adapter);
    toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    toolbar.setNavigationOnClickListener(new View.OnClickListener() {
        @Override

```

```

    public void onClick(View v){
        onBackPressed();
        /* Intent intent = new Intent(MainInfoActivity.this, MainActivity.class);
        startActivity(intent);*/
    }
});

TypedValue typedValueColorPrimaryDark = new TypedValue();
NewsActivity.this.getTheme().resolveAttribute(R.attr.colorPrimaryDark, typedValueColorPrimaryDark, true);
final int colorPrimaryDark = typedValueColorPrimaryDark.data;
if (Build.VERSION.SDK_INT >= 21) {
    getWindow().setStatusBarColor(colorPrimaryDark);
}

Content content = new Content();
content.execute();
}

private class Content extends AsyncTask<Void,Void,Void>{

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressBar.setVisibility(View.VISIBLE);
        progressBar.startAnimation(AnimationUtils.loadAnimation(NewsActivity.this, android.R.anim.fade_in));
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
        progressBar.setVisibility(View.GONE);
        progressBar.startAnimation(AnimationUtils.loadAnimation(NewsActivity.this, android.R.anim.fade_out));
        adapter.notifyDataSetChanged();
    }

    @Override
    protected Void doInBackground(Void... voids) {

        try {
            String url = "https://kipt.sumdu.edu.ua/";

            Document doc = Jsoup.connect(url).get();

            Elements data = doc.select("div.bt-inner");
            int size = data.size();
            for (int i = 0; i < size; i++) {
                String imgUrl = data.select("img.hovereffect")
                    .select("img")
                    .eq(i)
                    .attr("src");

                String title = data.select("div.bt-inner")
                    .select("a.bt-title")
                    .eq(i)
                    .text();
                String description = data.select("div.bt-inner")
                    .select("div.bt-introtext")
                    .eq(i)
                    .text();
                String detailUrl = url + data.select("div.bt-inner")
                    .select("a.bt-title")

```



```

        .eq(i)
        .attr("href");

        parseItems.add(new ParseItem(imgUrl, title, description, detailUrl));
        Log.d("items", "img: " + imgUrl + " . title: " + title + ".description: "+ description);
        //+ ".detailUrl: "+ detailUrl
    }

    } catch (IOException e) {
        e.printStackTrace();
    }

    return null;
}
}
}
}

```

SpecActivity.java :

```

public class SpecActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private ParseAdapter1 adapter1;
    private ArrayList<ParseItem1> parseItems1 = new ArrayList<>();
    private ProgressBar progressBar;
    Toolbar toolbar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_spec);

        progressBar = findViewById(R.id.progressBar);
        recyclerView = findViewById(R.id.recyclerView);

        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        adapter1 = new ParseAdapter1(parseItems1, this);
        recyclerView.setAdapter(adapter1);
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        toolbar.setNavigationOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v){
                onBackPressed();
                /* Intent intent = new Intent(MainInfoActivity.this, MainActivity.class);
                startActivity(intent);*/
            }
        });

        TypedValue typedValueColorPrimaryDark = new TypedValue();
        SpecActivity.this.getTheme().resolveAttribute(R.attr.colorPrimaryDark, typedValueColorPrimaryDark, true);
        final int colorPrimaryDark = typedValueColorPrimaryDark.data;
        if (Build.VERSION.SDK_INT >= 21) {
            getWindow().setStatusBarColor(colorPrimaryDark);
        }

        Content content1 = new Content();
        content1.execute();
    }
}

```

```

private class Content extends AsyncTask<Void,Void,Void>{

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressBar.setVisibility(View.VISIBLE);
        progressBar.startAnimation(AnimationUtils.loadAnimation(SpecActivity.this, android.R.anim.fade_in));
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
        progressBar.setVisibility(View.GONE);
        progressBar.startAnimation(AnimationUtils.loadAnimation(SpecActivity.this, android.R.anim.fade_out));
        adapter1.notifyDataSetChanged();
    }

    @Override
    protected Void doInBackground(Void... voids) {

        try {
            String url = "https://kipt.sumdu.edu.ua";

            Document doc = Jsoup.connect(url).get();

            Elements data = doc.select("div.clearfix");
            int size = 12;
            for (int i = 0; i < size; i++) {
                String imgUrl = url+data.select("img.sppb-img-responsive")
                    .select("img")
                    .eq(i)
                    .attr("src");

                String title = data.select("div.clearfix")
                    .select("h3.sppb-addon-title")
                    .eq(i)
                    .text();
                String detailUrl = url + data.select("div.sppb-addon-single-image-container")
                    .select("a")
                    .eq(i)
                    .attr("href");

                parseItems1.add(new ParseItem1(imgUrl, title, detailUrl));
                Log.d("items", "img: " + imgUrl + " . title: " + title);
                //+ ".detailUrl: "+ detailUrl
            }

        } catch (IOException e) {
            e.printStackTrace();
        }

        return null;
    }
}

```

ZaminiActivity.java :

```

public class ZaminiActivity extends AppCompatActivity {
    ListView listView;
    TextView textView;
    ListViewAdapter adapter;
    ProgressDialog mProgressDialog;
    ArrayList<HashMap<String, String>> arraylist;
    static String GRUPA = "grupa";
    static String SUBJ_NUMBER = "subj_number";
    static String ZA_ROZKLADOM = "za_rozkladom";
    static String ZAMINA = "zamina";
    static String CLASSROOM = "classroom";
    Toolbar toolbar;
    Boolean isInternetPresent = false;
    ConnectionDetector cd;
    // URL Address
    String url = "https://kipt.sumdu.edu.ua/zaminy-na-zavtra";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Get the view from listview_main.xml
        setContentView(R.layout.listview_main);
        // Execute DownloadJSON AsyncTask
        cd = new ConnectionDetector(this);
        //Получаем статус Интернет
        isInternetPresent = cd.ConnectingToInternet();
        if (isInternetPresent) {
            new JsoupListView().execute();
        } else {
            Toast.makeText(ZaminiActivity.this, "Перевірте з'єднання з інтернетом", Toast.LENGTH_LONG).show();
        }
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        toolbar.setNavigationOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v){
                onBackPressed();
                /* Intent intent = new Intent(ZaminiActivity.this, MainActivity.class);
                startActivity(intent);*/
            }
        });

        TypedValue typedValueColorPrimaryDark = new TypedValue();
        ZaminiActivity.this.getTheme().resolveAttribute(R.attr.colorPrimaryDark, typedValueColorPrimaryDark, true);
        final int colorPrimaryDark = typedValueColorPrimaryDark.data;
        if (Build.VERSION.SDK_INT >= 21) {
            getWindow().setStatusBarColor(colorPrimaryDark);
        }
    }

    // Title AsyncTask
    private class JsoupListView extends AsyncTask<Void, Void, Void> {

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            // Create a progressdialog
            mProgressDialog = new ProgressDialog(ZaminiActivity.this,R.style.AppCompatAlertDialogStyle);
            // Set progressdialog title
            mProgressDialog.setTitle("Зачекайте");
        }
    }
}

```

```

    // Set progressdialog message
    mProgressDialog.setMessage("Завантаження замін...");
    mProgressDialog.setIndeterminate(false);
    // Show progressdialog
    mProgressDialog.show();
}
@Override
protected Void doInBackground(Void... params) {
    // Create an array
    arraylist = new ArrayList<HashMap<String, String>>();
    try {
        // Connect to the Website URL
        Document doc = Jsoup.connect(url).get();
        // Identify Table Class "worldpopulation"

        for (Element table : doc.select("table")) {
            // Identify all the table row's(tr)
            for (Element row : table.select("tr")) {
                HashMap<String, String> map = new HashMap<String, String>();

                // Identify all the table cell's(td)
                Elements tds = row.select("td");

                // Retrive Jsoup Elements
                // Get the first td
                map.put("grupa", tds.get(0).text());
                // Get the second td
                map.put("subj_number", tds.get(1).text());
                // Get the third td
                map.put("za_rozkladom", tds.get(2).text());

                map.put("zamina", tds.get(3).text());

                map.put("classroom", tds.get(4).text());

                // Set all extracted Jsoup Elements into the array
                arraylist.add(map);
            }
        }
    } catch (IOException e ){
        // TODO Auto-generated catch block
        e.printStackTrace();
        Log.e("Помилка", e.getMessage());
    }
    return null;
}
@Override
protected void onPostExecute(Void result) {
    // Locate the listview in listview_main.xml
    listview = (ListView) findViewById(R.id.listview);
    // textView = (TextView) findViewById(R.id.textView);

    // Pass the results into ListViewAdapter.java
    adapter = new ListViewAdapter(ZaminiActivity.this, arraylist);
    // Set the adapter to the ListView
    listview.setAdapter(adapter);
    // Close the progressdialog
    mProgressDialog.dismiss();
}
}
}

```