

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**на тему: «Мобільний додаток супроводження
навчального процесу НВК №11»**

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

Виконавець роботи: студент групи ІТ-81 Сітало Максим Євгенійович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2022 р.

Науковий керівник

(підпис)

к.т.н., доц., Ващенко С. М.
(науковий ступінь, вчене звання, прізвище та ініціали)

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

Студент _____
(підпис)

Суми-2022

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

ЗАТВЕРДЖУЮ
Зав. кафедрою ІТ

В. В. Шендрик
«___» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Сітало Максим Євгенійович

1 Тема роботи Мобільний додаток супроводження навчального процесу НВК №11

керівник роботи Ващенко С. М., к.т.н., доц.,
затверджені наказом по університету від «27» квітня 2022 р. № 0301-IV

2 Строк подання студентом роботи «10» червня 2022 р.

3 Вхідні дані до роботи технічне завдання на розробку мобільного додатку супроводження навчального процесу НВК №11

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування мобільного-додатку, розробка мобільного додатку

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Актуальність роботи. Постановка задачі. Аналіз предметної області. Структурно-функціональна модель. Діаграма варіантів використання. Схема бази даних. Демонстрація роботи. Висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання: 07.10.2021

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Оформлення планування робіт	01.12.2021- 08.12.2021	виконано
2	Оформлення технічного завдання	09.12.2021- 16.12.2021	виконано
3	Проведення аналізу предметної області	17.12.2021- 05.01.2022	виконано
4	Проведення проектування мобільного додатку	06.01.2022- 21.01.2022	виконано
5	Розробка мобільного додатку	22.01.2022- 14.04.2022	виконано
6	Наповнення мобільного додатку контентом	15.04.2022- 27.04.2022	виконано
7	Тестування мобільного додатку	28.04.2022- 16.05.2022	виконано
8	Оформлення пояснювальної записки	17.05.2022- 10.06.2022	виконано

Студент

(підпис)

Сітало М. Є.

Керівник роботи

(підпис)

к.т.н., доц., Ващенко С. М.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Мобільний додаток супроводження навчального процесу НВК №11».

Пояснювальна записка складається зі вступу, трьох розділів, висновку, списку використаних джерел із 16 найменувань, трьох додатків. Загальний обсяг пояснювальної записки складає 90 сторінок, у тому числі 32 сторінки основного тексту, 2 сторінки списку використаних джерел, 57 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці мобільного додатку супроводження навчального процесу НВК №11.

У першому розділі проведено аналіз предметної області – визначено актуальність теми проекту, виконано огляд існуючих мобільних додатків з подібним функціоналом. На основі отриманих даних було поставлено мету проекту, і обрано засоби реалізації.

У другому розділі проведено структурно-функціональне моделювання додатку, аналіз можливих варіантів взаємодії користувача з модулями програми, створено схему бази даних. В результаті написання розділу було отримано діаграму IDEF0, її деталізацію, діаграми Use Case та Entity Relationship.

У третьому розділі описано структуру додатку, програмну реалізацію ключових модулів та приклади використання мобільного додатку.

Ключові слова: **МОБІЛЬНИЙ ДОДАТОК, БАЗА ДАНИХ, ANDROID, JAVA, FIREBASE, FRAGMENT, РОЗРОБКА, НАВЧАЛЬНИЙ ПРОЦЕС.**

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Визначення актуальності дослідження.....	8
1.2 Аналіз існуючих продуктів-аналогів	9
1.3 Постановка задачі	14
2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ	15
2.1 Структурно-функціональне моделювання	15
2.2 Моделювання варіантів використання	17
2.3 Проектування мобільного додатку.....	19
3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ	21
3.1 Архітектура мобільного додатку.....	21
3.2 Програмна реалізація.....	22
3.3 Використання програмного додатку.....	26
ВИСНОВОК.....	33
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	34
ДОДАТОК А.....	36
ДОДАТОК Б	44
ДОДАТОК В.....	56

ВСТУП

Інформаційні технології стрімко розвиваються і проникають в усі сфери людської діяльності. Галузь освіти не стала винятком. Сучасне навчання значною мірою змінилося у порівнянні з тим, що було десять чи двадцять років тому. Однією з найбільш важливих змін є активне впровадження інформаційних технологій в навчальний процес. Колись було важко уявити, що навчатися повноцінно можна без відвідування закладів освіти, просто сидячи вдома, а зараз це – буденна справа. Міністерство освіти наразі активно працює над реформою нової української школи. Одним із головних принципів НУШ є запровадження в освітній галузі використання інформаційно-комунікаційних технологій на постійній основі для розширення можливостей педагога [1]. Використання спеціалізованих інструментів дозволяє значною мірою оптимізувати більшість навчальних процесів, тому їх актуальність найближчим часом тільки зростатиме.

Мобільний додаток супроводження навчального процесу якраз і є одним із вищеописаних спеціалізованих інструментів. На відміну від більшості технологій, що зараз використовуються у навчальному процесі, він буде направлений лише на використання в навчанні. Такий підхід означає що він не матиме зайвих та відволікаючих функцій, що допоможе більш якісно опрацьовувати інформацію. Крім того, мобільний додаток позбавлений деяких незручностей, що можуть виникати при роботі з веб-сторінками. Він має зручний та зрозумілий інтерфейс, і його відкриття займає менше часу ніж пошук необхідного сайту у браузері.

Для досягнення мети проекту необхідно виконати наступні задачі:

- визначити актуальність роботи, дослідити предметну область та провести аналіз аналогів мобільних додатків;
- провести моделювання роботи майбутнього додатку;

- розробити та реалізувати структуру та функціонал мобільного додатку;
- виконати тестування мобільного додатку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення актуальності дослідження

Смартфони та інші мобільні пристрої вже давно стали невід'ємною частиною життя будь-якої сучасної людини. Згідно з результатами дослідження DataPortal, більше 31 мільйона українців постійно користуються мережею Інтернет [2]. Без сумніву, переважна частина з них використовує для цього саме мобільний телефон. Маючи під рукою такий потужний та багатофункціональний пристрій, було б дуже недоцільно використовувати його лише для розваг.

Спалах пандемії значною мірою пришвидшив діджиталізацію в усіх сферах людського життя. Під поняттям «діджиталізація» зазвичай розуміють переведення інформації в будь-яких її проявах у цифровий формат [3]. Коли люди втратили можливість спілкуватися віч-на-віч, вони були вимушені перенести свої зустрічі у формат відеоконференцій. Так, зокрема, навчальні заклади проводили заняття у форматі відеоконференцій, а більшість компаній перенесла туди свої ділові зустрічі. Такий формат виявився надиво зручним, що лише доводить необхідність подальшої інтеграції інформаційних технологій в повсякденне життя.

Що стосується мобільних додатків, то вони ще не набули такої шаленої популярності як веб-сайти. Наразі кожен магазин, державна установа чи будь-який інший заклад намагаються створювати та розвивати саме свій веб-сайт, а не мобільний додаток. Причиною такого вибору є те, що веб-сайт зручніший для разового використання. Наприклад, якщо людині необхідно забронювати столик в ресторані, то їй набагато простіше зробити це через сайт. Це значно швидше ніж встановлювати мобільний додаток, який потім, швидше за все, залишиться на довгий час непоміченим на екрані телефону, і лише буде займати вільну пам'ять, так як більшість людей ходять в ресторани доволі

рідко. Мобільними додатками набагато зручніше користуватись, коли його функціонал потрібно використовувати щодня. Наприклад, спілкування з друзями чи колегами в соціальній мережі. Коли необхідно бути постійно на зв'язку, набагато простіше натиснути на іконку додатку, ніж знаходити у браузері необхідний ресурс. Аналогічно, можна сказати що використання мобільних додатків в освітньому процесі є цілком виправданим. Студенти, школярі, викладачі та вчителі приймають участь у освітньому процесі майже кожного дня протягом років. І якщо вже вони вимушені робити це в цифровому просторі, то їх смартфон може стати чудовим помічником у цьому [4].

Розробка мобільного додатку супроводження навчального процесу, який би забезпечив якісну та швидку комунікацію між вчителями, учнями та батьками в рамках діяльності НВК11 є актуальною задачею в умовах поточної ситуації в галузі освіти.

1.2 Аналіз існуючих продуктів-аналогів

Мобільні додатки супроводження навчального процесу вже не новинка, і мають багатьох представників. Серед них можна знайти як офіційні державні додатки, так і користувацькі з вільним доступом. Більшість таких додатків має загальний функціонал, який необхідний будь – якому закладу, але далеко не завжди вони можуть задовільнити конкретні потреби тієї чи іншої школи. Ще одним істотним мінусом таких додатків є їх відносна ненадійність – невідомо де розташовуються їх бази даних, чи безпечно вони захищені, і хто буде відповідати якщо завтра додаток перестане працювати, тим самим зупинивши весь навчальний процес.

Тому, набагато спокійніше мати власний додаток, в якому заклад сам має можливість контролювати безпеку даних, і коректно підтримувати його

працевдатність. До того ж його можна розробити згідно до конкретних потреб навчальної установи, і за необхідності в майбутньому доповнити новим функціоналом. Для визначення відповідності існуючих додатків потребам НВК №11 було проведено дослідження існуючих продуктів зі схожим функціоналом, а саме «Oshkola», «Всеукраїнська школа онлайн» та «Єдина школа».

«Єдина школа» - мобільний додаток, що забезпечує базовий функціонал, такий як перегляд розкладу уроків, можливість здачі робіт, комунікації з викладачами та однокласниками у вигляді чату. Можна переглянути статистику у вигляді графіків. Має простий та зрозумілий дизайн (рис. 1.1) [5].

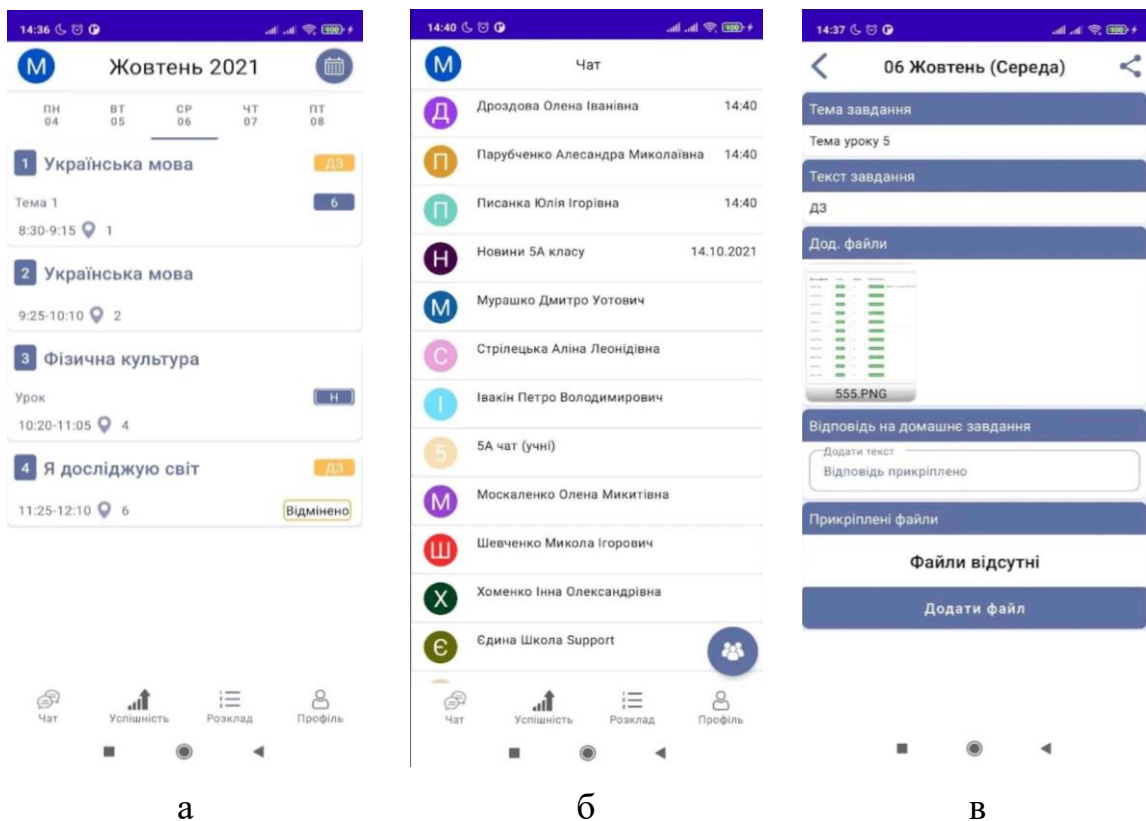


Рисунок 1.1 – Екрани додатку «Єдина школа»: екран розкладу уроків (а), екран з чатами (б), екран з призначеним домашнім завданням (в)

До недоліку можна віднести недостатній рівень безпеки і надійності.

«Oshkola» - мобільний додаток, що декілька років тому впроваджувався в навчальний процес за підтримки Міністерства освіти і

науки України. Проте, популярності він не набув через велику кількість технічних проблем. Функціонал фактично ідентичний з функціоналом «Єдиної школи». Дозволяє переглядати розклад, здавати роботи, спостерігати за успішністю навчання. Дизайн приємний і зручний (рис. 1.2).

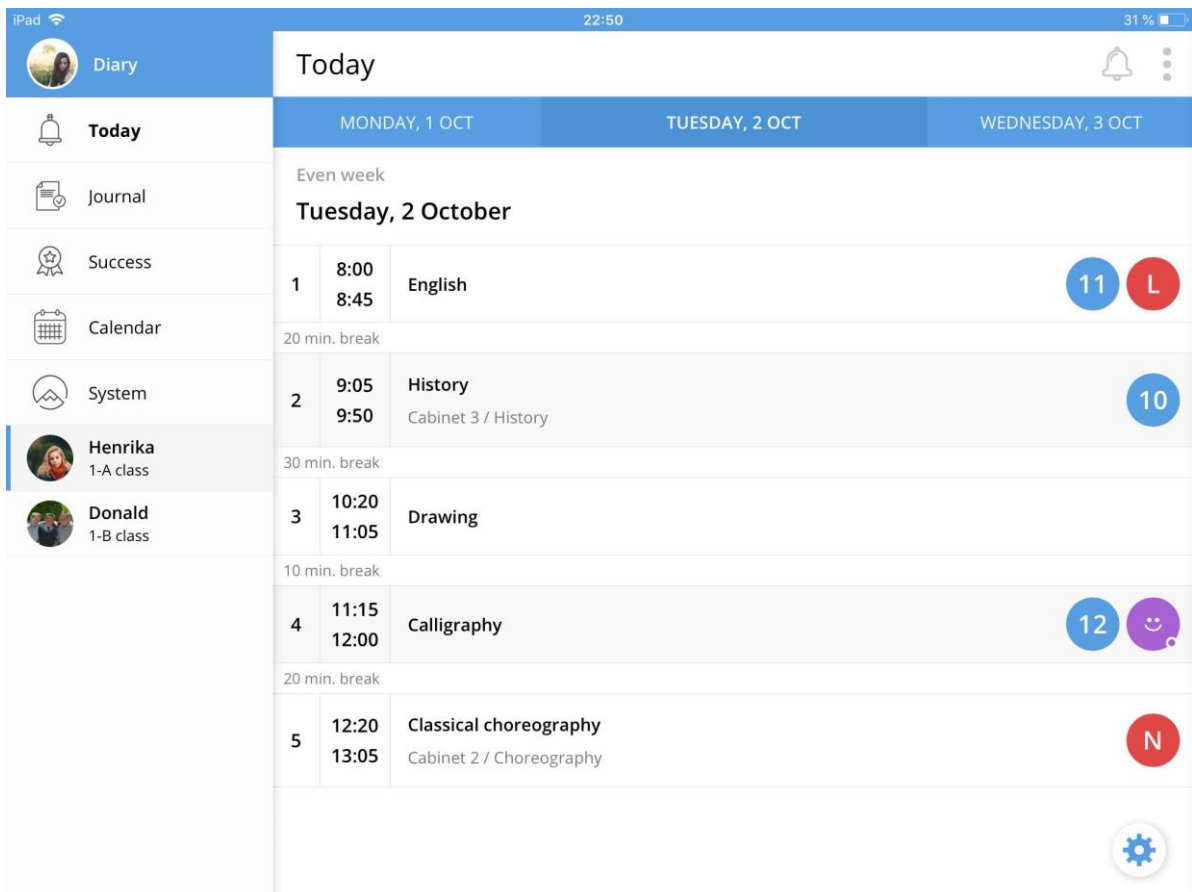


Рисунок 1.2 – Екран додатку «Oshkola»

Головний недолік – постійні зависання і відмови у роботі, які унеможливають комфортне користування. Також сюди можна віднести безпеку і надійність. Реєстрація закладу відбувається через уповноважених представників міністерства, відповіді від яких доводиться чекати доволі довго.

«Всеукраїнська школа онлайн» - мобільний додаток, що значною мірою відрізняється від двох попередніх. Його мета полягає не у допомозі проведення навчального процесу за рахунок спрощення комунікації між

вчителями, учнями та батьками. Цей додаток створений для допомоги учням у самостійному навчанні. Містить навчальні матеріали по всіх предметах за 5-11 клас. Серед матеріалів можна знайти короткі зрозумілі відеоролики, та письмові конспекти. Після ознайомлення з ними, учень має змогу перевірити засвоєний матеріал та перейти до наступної теми. [6]



Рисунок 1.3 – Екрани додатку «ВШО»: екран вибору предмета (а), екран з навчальним відео (б), екран з тестами (в)

До недоліків відноситься відсутність основного необхідного функціоналу. Це пояснюється тим, що додаток ставить перед собою зовсім іншу мету, з якою він чудово справляється.

Після детального аналізу аналогів мобільних додатків супроводження навчального процесу, було визначено їх переваги та недоліки. Його результати представлені в таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця характеристик аналогів мобільних додатків

Характеристика/ Додаток	«Єдина школа»	«Oshkola»	«Всеукраїнська школа онлайн»
Сучасний дизайн	-	+	+
Зручний інтерфейс	+	-	+
Функціональність	+	+	-
Можливість комунікації	+	+	-
Наявність навчальних матеріалів	+	+	+
Наявність інформації про заклад	-	-	-
Безпека даних	-	-	+
Стабільність роботи	+	-	+

На основі даних, представлених в таблиці 1.1, можна сформулювати вимоги до власної розробки, враховуючи переваги і виправляючи недоліки проаналізованих додатків. Розроблюваний мобільний додаток повинен мати сучасний дизайн, зручний та зрозумілий інтуїтивно інтерфейс. Реалізувати основну функціональність (перегляд розкладу та новин), а також реалізувати інструменти комунікації між учасниками навчального процесу, поступового заповнення навчально-методичними та організаційними матеріалами. Не менш важливим є забезпечення безпеки особистих даних користувачів, та стабільна робота додатку.

1.3 Постановка задачі

Основна мета проекту полягає у спрощенні та оптимізації ведення навчального процесу Сумської початкової школи №11 за допомогою мобільного додатку. Для досягнення поставленої мети треба вирішити наступні завдання:

- розробити простий та зручний інтерфейс для додатку;
- реалізувати бізнес-логіку додатку;
- реалізувати зв'язок з базою даних;
- наповнити додаток контентом;
- провести тестування додатку;
- ввести в експлуатацію.

Додаток повинен бути зручним у використанні на більшості телефонів з ОС Android та відповідати усім вимогам замовника. Користувачами додатку будуть учасники навчального процесу НВК №11 – адміністрація школи, вчителі, учні та їх батьки.

Розробка буде виконуватися в середовищі розробки Android Studio [7]. Додаток буде написаний мовою програмування Java [8], інформація буде зберігатися у базі даних реального часу Firebase [9].

Повний список вимог наведено в технічному завданні в додатку А. Планування робіт представлено у додатку Б.

2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

2.1 Структурно-функціональне моделювання

Моделювання складається з етапів, які пов'язані між собою. Процес моделювання починається з абстрактної концептуальної схеми, після неї створюються логічна та фізична моделі.

Нотація IDEF0 використовується для наглядної демонстрації бізнес-логіки проекту [10]. При створенні діаграми в центрі створюється прямокутник, який представляє собою проект. В прямокутник з усіх сторін направлено стрілки, що представляють собою типи вхідних та вихідних даних.

Діаграма будується з точки зору користувача – батьків або учня. Для контекстної діаграми були визначено такі дані:

- Вхідні дані: логін і пароль користувача, потреба користувача у отриманні певної інформації.
- Вихідні дані: інформація, які відповідає потребі користувача та задовольняє її.
- Управління: наявна в додатку інформація, інструкції з користування.
- Механізми: мобільний додаток, користувач, апаратне та технічне забезпечення.

Функціональне моделювання мобільного додатку супроводження навчального процесу НВК №11 в IDEF0 представлено на рисунку 2.1.

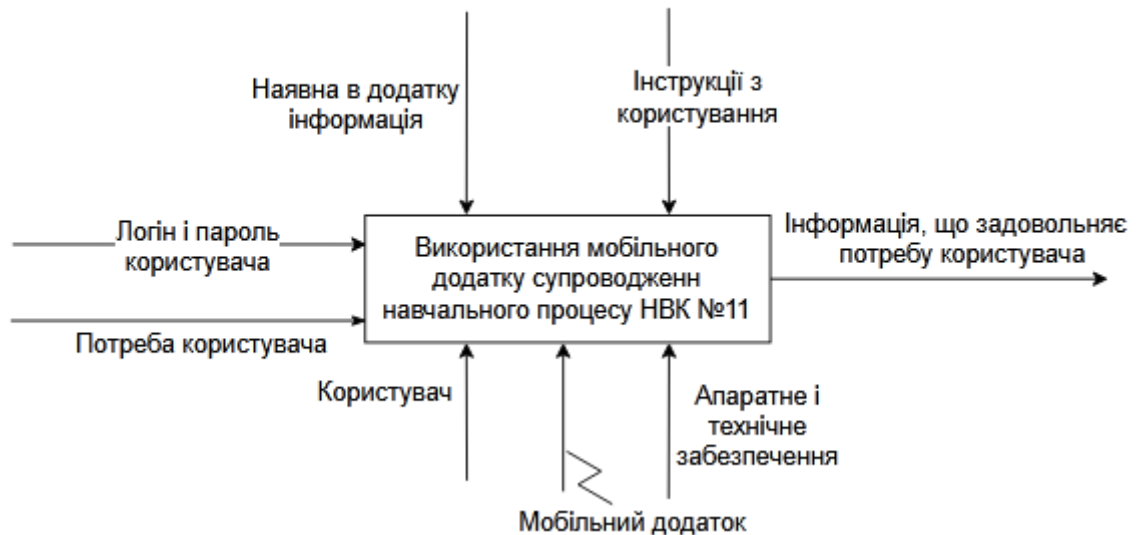


Рисунок 2.1 – Функціональна діаграма

Для користування додатком користувачеві необхідно спершу заповнити форму авторизації, увівши свій логін і пароль. Після авторизації він отримає доступ до функціоналу додатку – перегляду і редагування доступної для нього інформації.

Для деталізації внутрішніх процесів діаграми IDEF0 було виконано декомпозицію, що зображена на рисунку 2.2.

Декомпозиція першого рівня відображає деталізацію процесу використання мобільного додатку супроводження навчального процесу. Структура блоків показує, як одні функції впливають на інші. [11]

Після виконання декомпозиції було виділено 3 основні функції користування додатком:

- авторизація;
- вибір потрібної дії;
- перегляд інформації у відповідності до обраної дії.

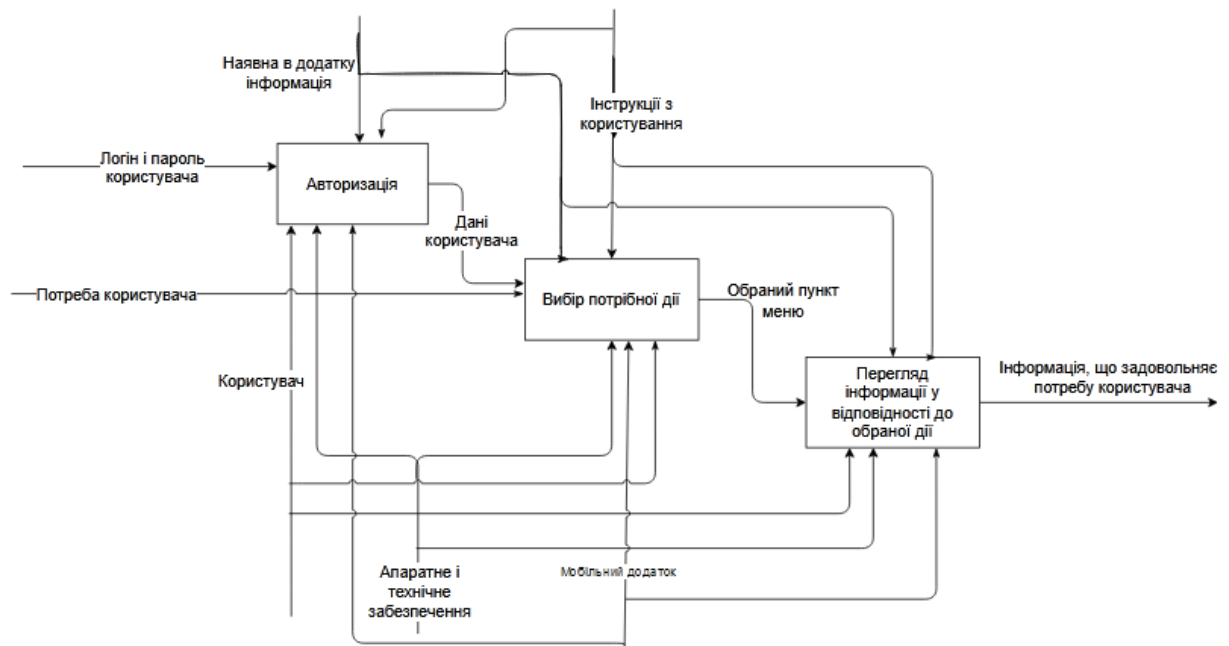


Рисунок 2.2 – Діаграма декомпозиції процесу функціонування

Під час процесу авторизації відбувається введення логіну та паролю користувача в форму авторизації. В результаті користувач отримує доступ до того функціоналу, який передбачено його обліковим записом. Після вибору пункту меню надсилається запит до бази даних за відповідною інформацією, і користувач отримує можливість її переглянути.

2.2 Діаграма варіантів використання

Після проведення моделювання структурно-функціонального моделювання мобільного додатку необхідно розробити діаграму варіантів використання.

Діаграма варіантів використання описує функціональне призначення системи. Суть такої діаграми полягає у графічному відображенні взаємодії між користувачами і компонентами системи [12]. Під взаємодією мається на увазі

те, що актор має можливість взаємодіяти з функціональним модулем. Кожен можливий варіант взаємодії визначається як варіант використання.

Акторами було визначено незареєстрованого користувача, батьків та учня, вчителя та адміністратора.

Діаграма варіантів використання мобільного додатку супроводження навчального процесу представлена на рисунку 2.3.

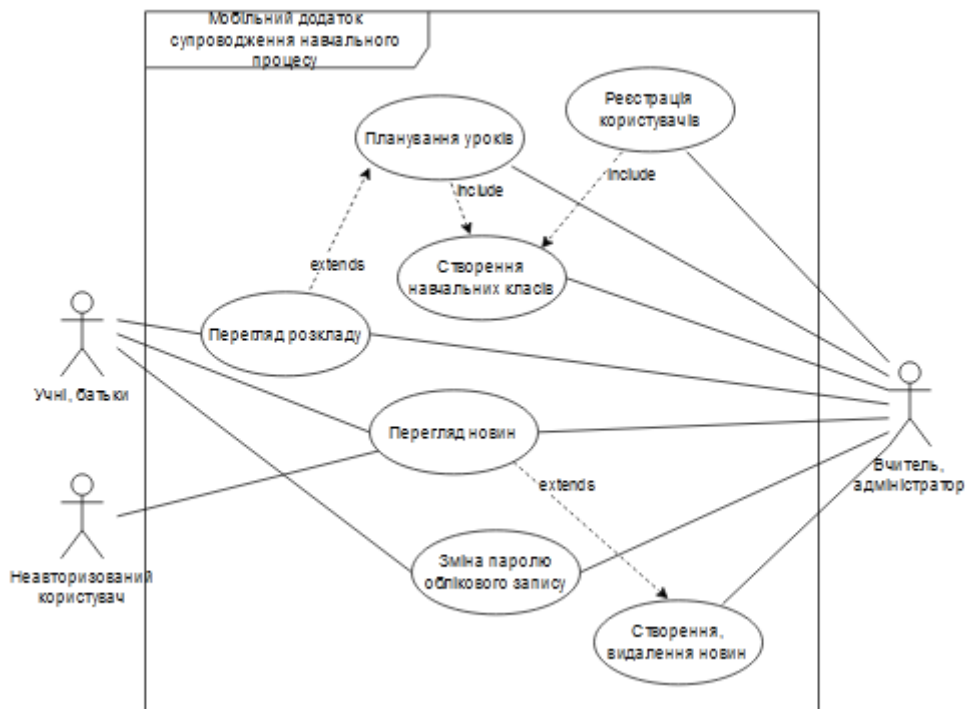


Рисунок 2.3 – Діаграма варіантів використання

Для мобільного додатку супроводження навчального процесу НВК №11 було визначено наступні варіанти використання:

- перегляд новин;
- перегляд розкладу;
- реєстрація користувачів;
- створення навчальних класів;
- планування уроків;
- зміна паролю облікового запису;
- створення, видалення новин.

2.3 Схема даних

Для розробки мобільного додатку супроводження навчального процесу НВК №11 використовувалась база даних в реальному часі Firebase. Ця база не має реляційних відношень у прямому вигляді, але з метою забезпечення зв'язків між сутностями даних в якості зовнішніх ключів використовувались унікальні ідентифікатори сутностей, які створюються самою базою даних.

Для бази даних мобільного додатку супроводження навчального процесу НВК №11 було виділено наступні сутності:

- Користувач
- Стаття новин
- Навчальний клас
- Урок

Кожна з виділених сутностей має свій набір атрибутів, необхідний для функціонування додатку. Користувач містить свою персональну інформацію, його роль та приналежність до певного навчального класу. Для користувачів з роллю вчитель та адміністратор це параметр встановлено як null. Навчальний клас містить унікальний ідентифікатор користувача, що є класним керівником. Урок містить інформацію про урок, а також ідентифікатор класу, якому він призначений, і ідентифікатор вчителя, який буде цей урок проводити. Сутність статті з новинами не має ніяких зв'язків з попередніми, і зберігає в атрибутах саму статтю, і зображення до неї.

Взаємозв'язки між сутностями представлено в Entity Relationship діаграмі [13] на рис. 2.4.

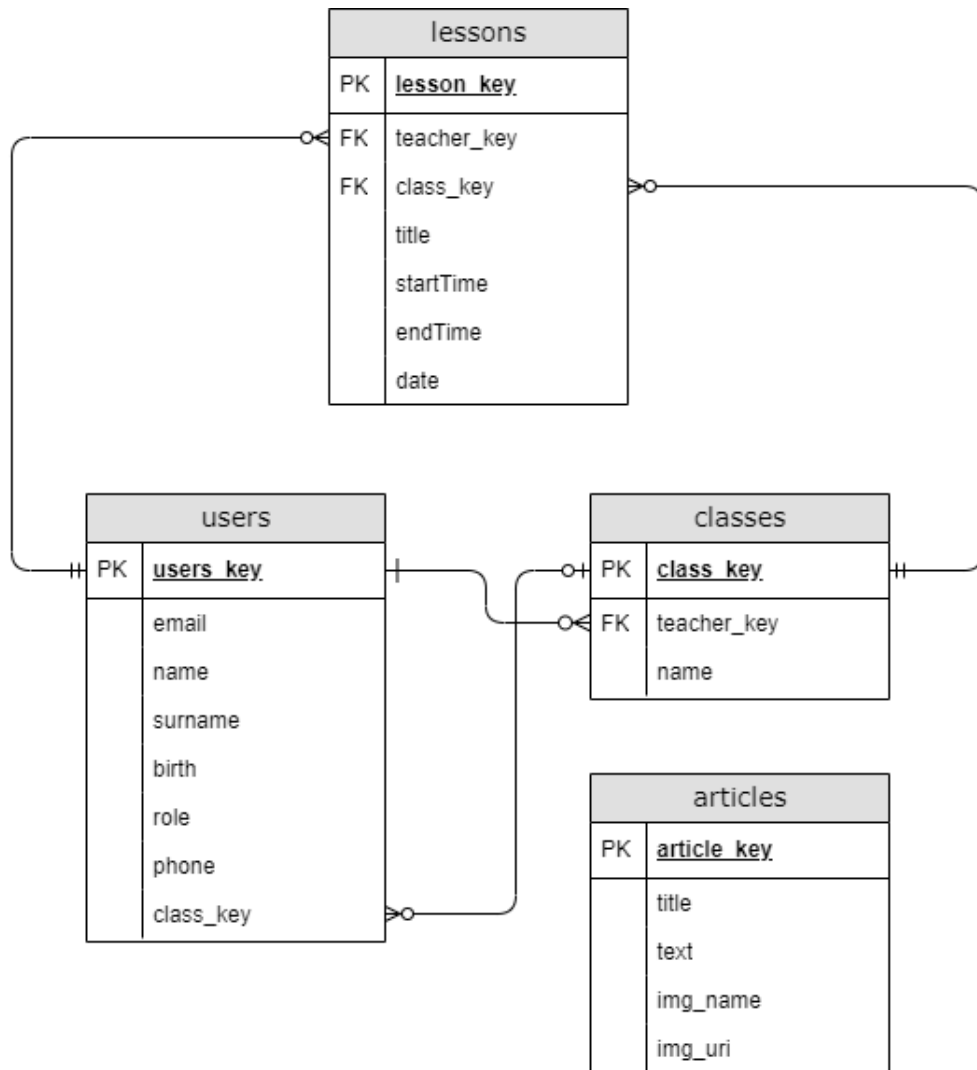


Рисунок 2.4 – Діаграма взаємозв'язків між сутностями БД

3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

3.1 Архітектура мобільного додатку

Під час розробки будь-яких додатків, у тому числі і мобільних, важливо правильно структурувати програмні модулі. Такий підхід дозволяє в майбутньому, при доповненні додатку новим функціоналом значною мірою спростити розуміння вже існуючого коду розробником [14]. Одним із найпопулярніших варіантів вирішення цього питання є використання шаблонів проектування. Найпопулярнішим із таких є архітектурний шаблон MVC. Його суть полягає у розбитті функціоналу додатку на візуальну частину, модель представлення інформації та модулі-контролери які зв'язують дві попередні частини між собою [15]. Такий підхід дозволяє вносити зміни в певну частину проекту, не впливаючи суттєво на інші модулі.

При розробці в Android Studio файли візуального інтерфейсу за замовчуванням відділяються від своїх контролерів, тому для реалізації шаблону MVC залишається лише відокремити модель даних, і реалізувати в контролерах зв'язок між інтерфейсом і шаром представлення інформації.

Для мобільного додатку супроводження навчального процесу основними компонентами будуть фрагменти. За їх візуальне відображення будуть відповідати файли з XML розміткою, а за взаємодію з інформаційною моделлю – контролер.

Структуру програмних компонентів мобільного додатку супроводження навчального процесу наведено на рис. 3.1.

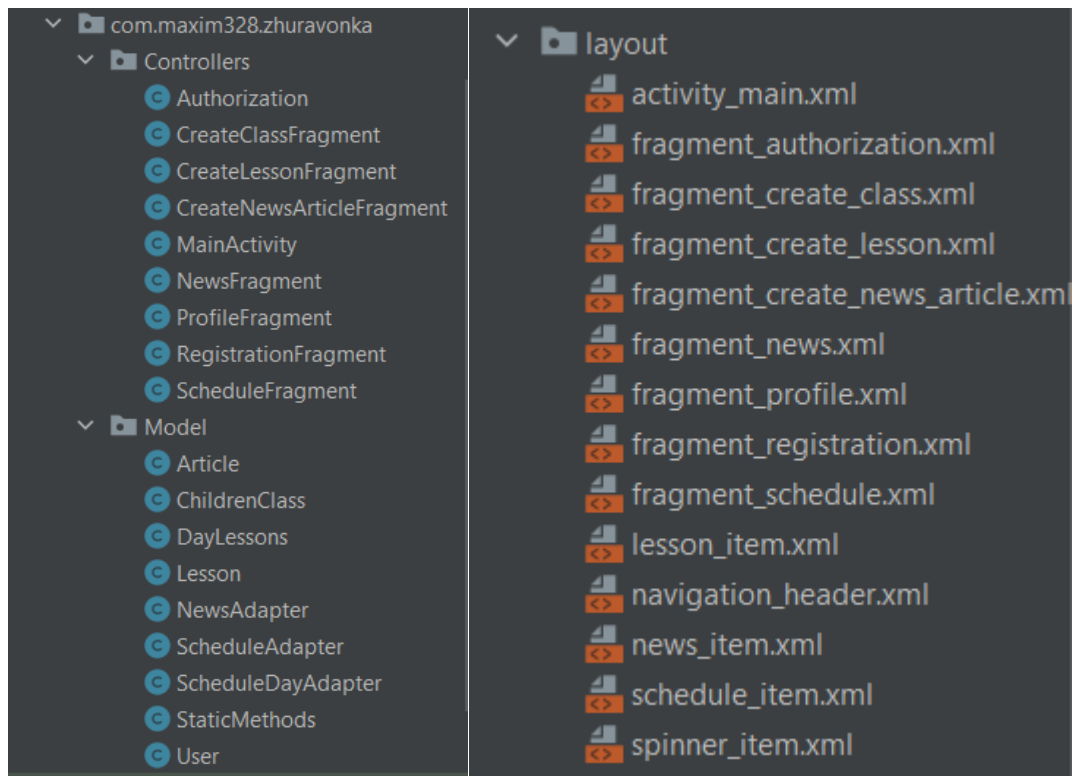


Рисунок 3.1 – Структура програмних компонентів

3.2 Програмна реалізація

Розробка мобільного додатку велась у середовищі Android Studio з використанням мови програмування Java. Мова розмітки графічного інтерфейсу – XML. Проект розроблений під мінімальний API 24(Android 7).

Програма містить лише одну активність, елементом якої є контейнер, в який поміщаються необхідні фрагменти. Навігація між фрагментами здійснюється за допомогою бокового меню в лівій частині. Додатково навігація здійснюється за рахунок динамічних елементів, таких як кнопки та посилання. Компоненти, що відповідають за навігацію наведено на рис. 3.2 – 3.4.

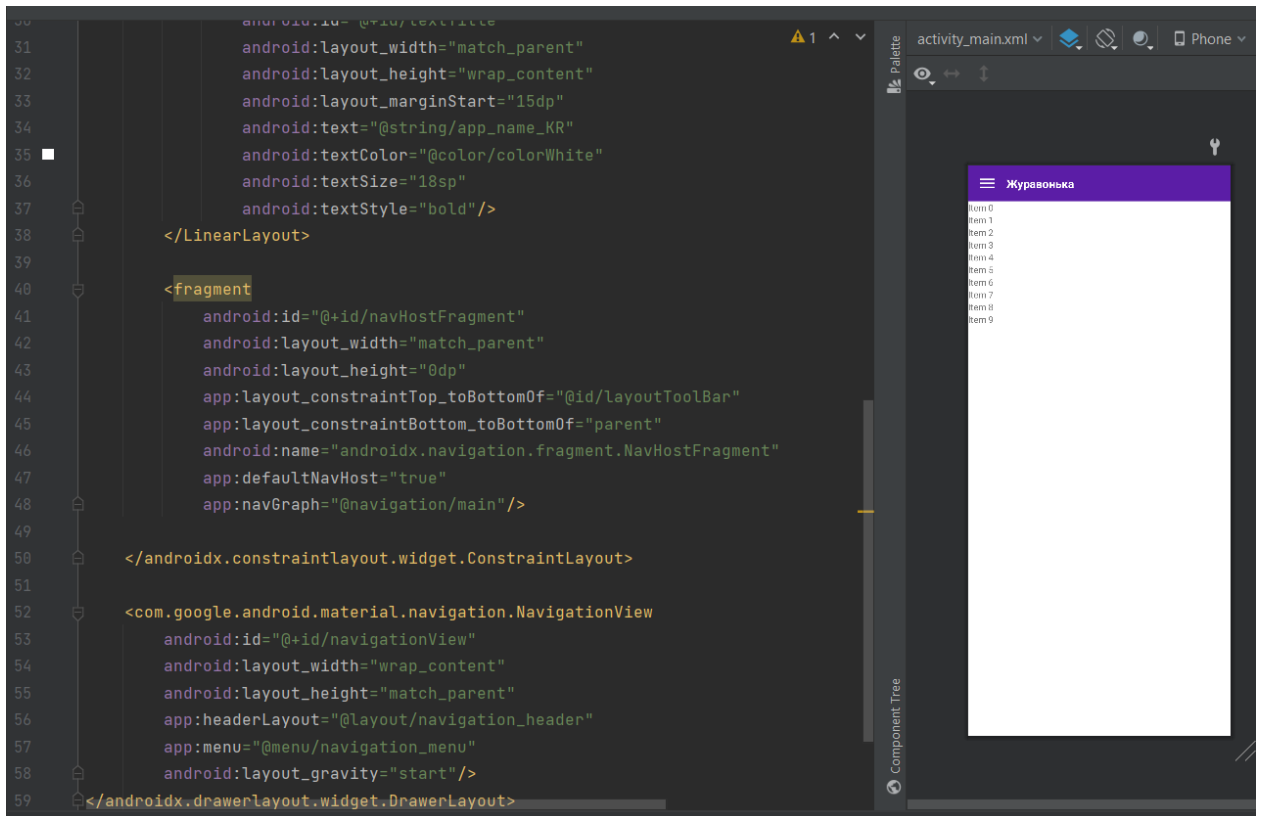


Рисунок 3.2 – Головний екран програми activity_main.xml

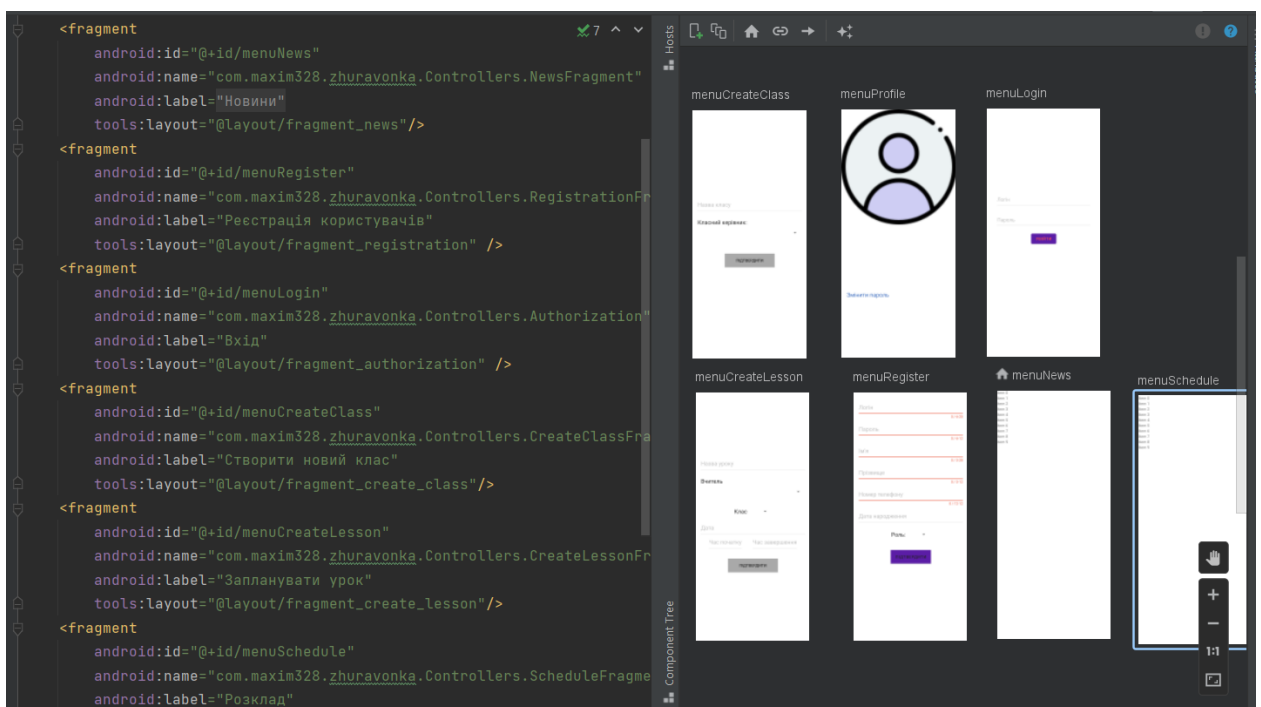


Рисунок 3.3 – Навігаційний компонент main.xml

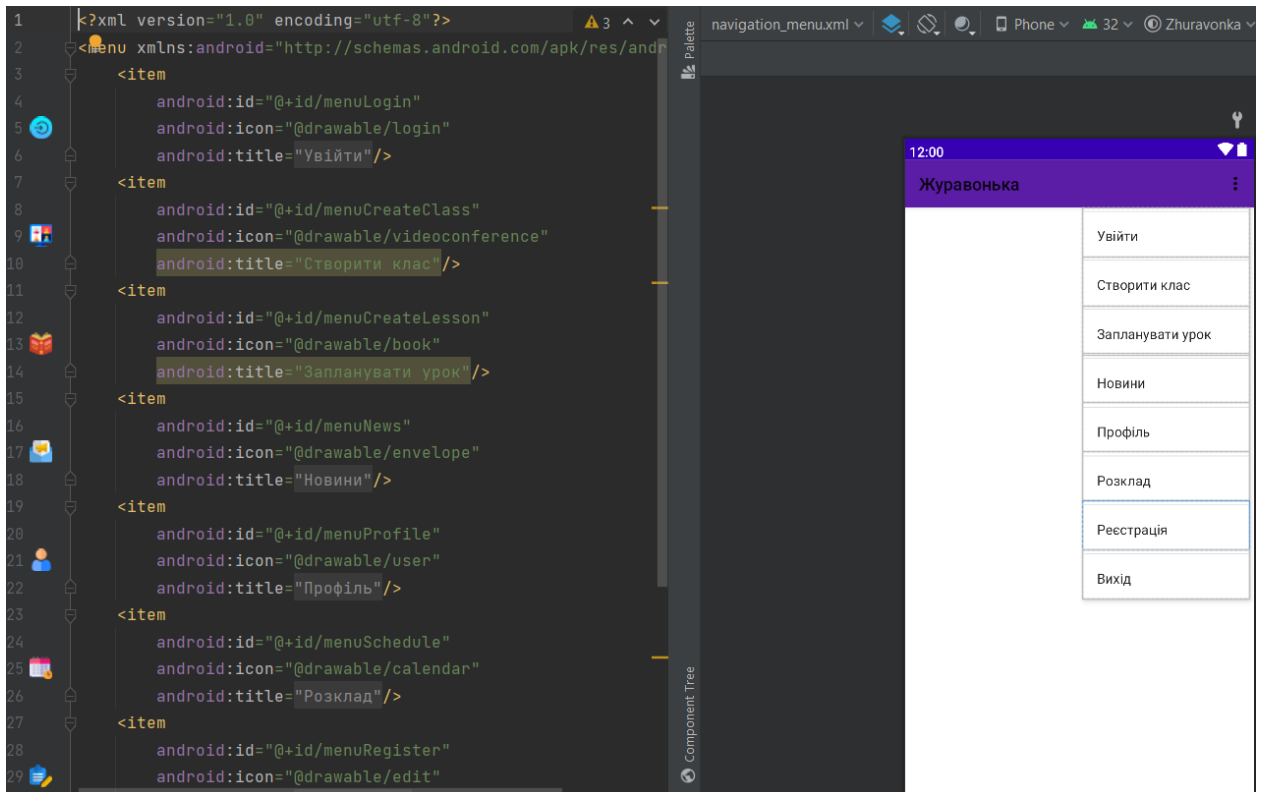


Рисунок 3.4 – Меню навігації `navigation_menu.xml`

Класи, що відповідають за збереження даних, та представляють собою інформаційну модель даних наведено на рис. 3.5 – 3.9

```

public class User {
    private String email, password, name, surname, phone, birth, className, dbKey, role;

    public User(String email, String name, String surname, String phone, String birth, String className, String role) {
        this.email = email;
        this.name = name;
        this.surname = surname;
        this.phone = phone;
        this.role = role;
        this.birth = birth;
        this.className = className;
    }
}

```

Рисунок 3.5 – Клас, що описує користувача


```
public class Lesson {
    private String title, teacher, className, startTime, endTime;

    public Lesson(String title, String teacher, String className, String startTime, String endTime) {
        this.title = title;
        this.teacher = teacher;
        this.startTime = startTime;
        this.className = className;
        this.endTime = endTime;
    }
}
```

Рисунок 3.6 – Клас, що описує урок

```
public class DayLessons {
    private Calendar date;
    private List<Lesson> lessons;

    public DayLessons() {
    }

    public DayLessons(Calendar date, List<Lesson> lessons) {
        this.date = date;
        this.lessons = lessons;
    }
}
```

Рисунок 3.7 – Клас, що описує навчальний день (список уроків)

```
public class ChildrenClass {
    private String name, teacher, dbKey;

    public ChildrenClass(String name, String teacher) {
        this.name = name;
        this.teacher = teacher;
    }
}
```

Рисунок 3.8 – Клас, що описує навчальний клас

```

public class Article {
    private String title, text, imgName, uri;

    public Article(String title, String text, String imgName, String uri) {
        if (imgName.trim().equals("")) imgName = "No name";
        else this.imgName = imgName;
        this.title = title;
        this.text = text;
        this.uri = uri;
    }
}

```

Рисунок 3.9 – Клас, що описує статтю новин

Опис основних контролерів та їх призначення наведено в таблиці 3.1

Таблиця 3.1 – Контролери мобільного додатку

Клас	Призначення
MainActivity.java	Створення головної активності, призначення контролера навігації, маніпулювання елементами меню навігації в залежності від ролі користувача
Authorization.java	Вхід користувача в систему з вказаними ним даними
CreateClassFragment.java	Створення нового навчального класу, та його внесення в базу даних
CreateLessonFragment.java	Створення уроку і його внесення в базу даних, видалення уроків
CreateNewsArticleFragment.java	Створення статті новин, завантаження зображення до Firestore, внесення запису в базу даних
NewsFragment.java	Отримання новин з бази даних, і їх виведення на екран
ProfileFragment.java	Отримання інформації про користувача з бази даних, і її відображення на екрані
RegistrationFragment.java	Внесення в базу даних інформації про нового користувача
ScheduleFragment.java	Отримання списку уроків з бази даних, їх фільтрація та відображення на екрані.

3.3 Використання програмного додатку

Робота з будь-яким додатком розпочинається із входу. Якщо користувач раніше вже заходив у додаток, то його дані запам'ятовуються, і авторизація проводиться автоматично, якщо ж ні – то він вважається гостем, і має можливість лише переглядати новини. Приклади входу з запам'ятованими даними і без, а також сторінку авторизації зображено на рис. 3.10

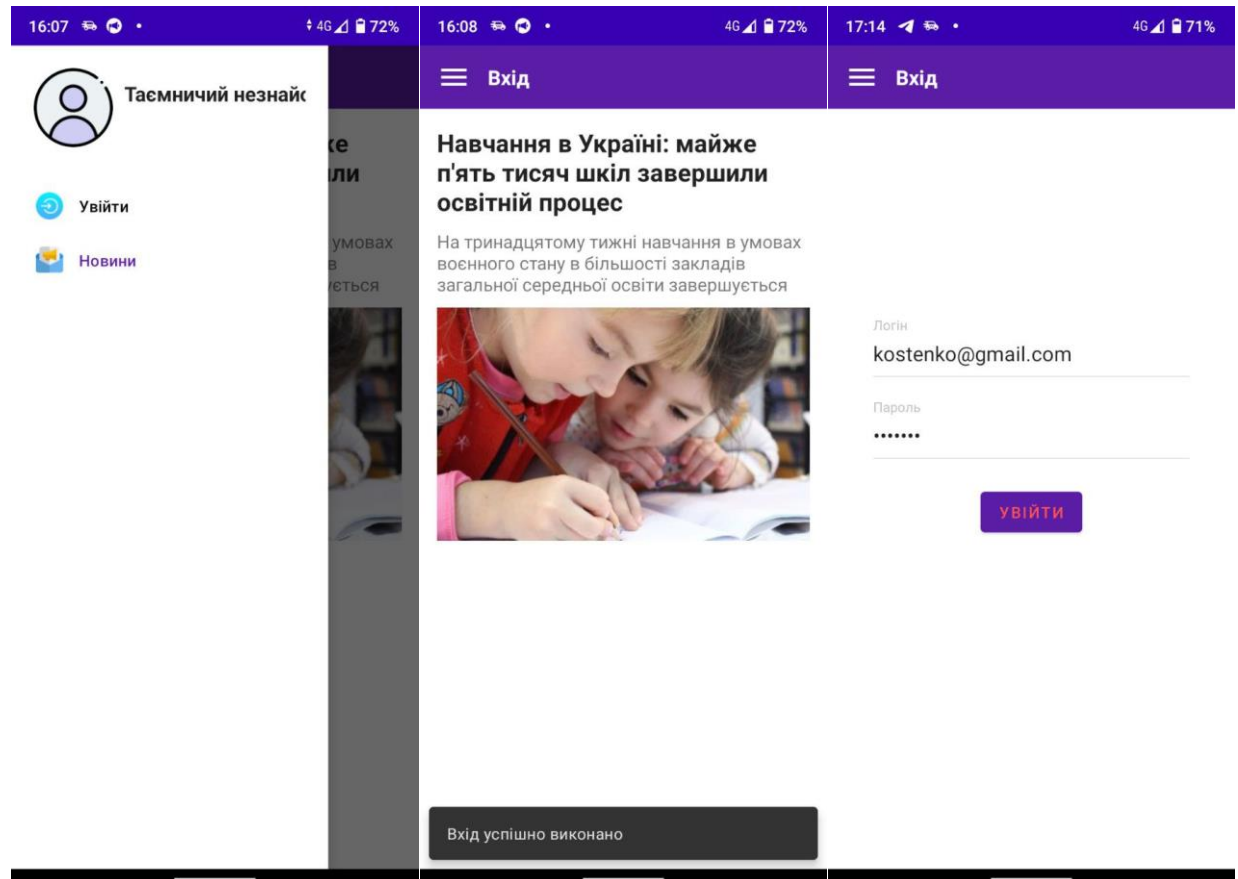


Рисунок 3.10 – Вхід до додатку та сторінка авторизації

Після входу, користувач отримує доступ до нового додатку. Меню навігації для учня і вчителя зображено на рис. 3.11

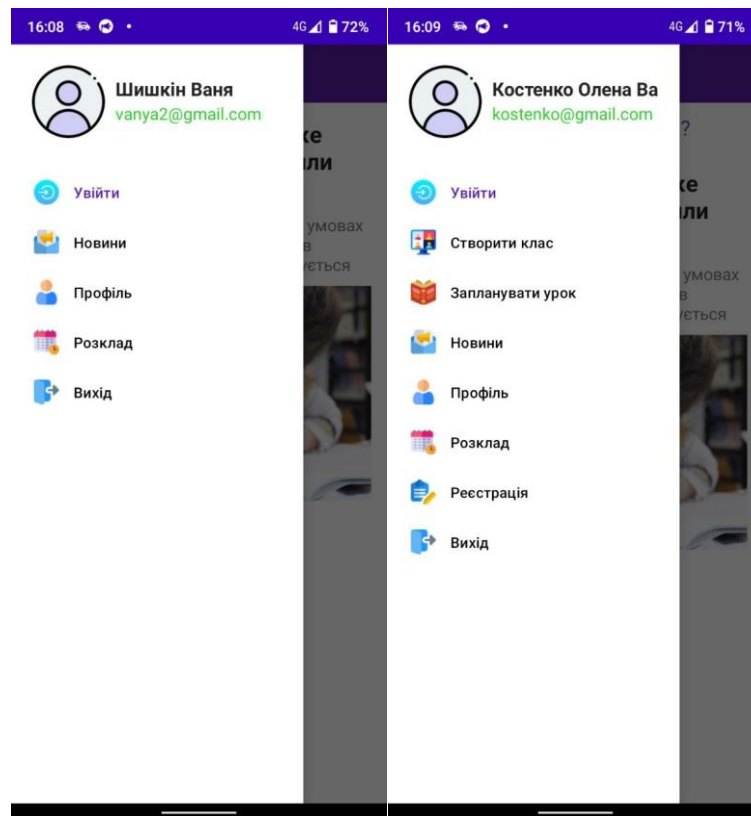


Рисунок 3.11 – Вигляд меню навігації для учня та вчителя

За бажання, вчитель або адміністратор можуть створювати нові новини, або видаляти старі. Для переходу на фрагмент створення і видалення новин у таких користувачів є спеціальне посилання вгорі блоку новин (рис. 3.12). Для того, щоб створити нову статтю необхідно вписати у відповідні поля фрагменту, що відкриється, назву статті, основний текст, назву зображення, та обрати з пристрою саме зображення. Якщо текст буде занадто великим для відображення, то його частина буде прихована, а відкрити її можна буде натиснувши на видиму частину тексту. Процес створення новини зображено на Рис. 3.13. В тому ж фрагменті доступне і видалення статей. Для цього необхідно в списку обрати заголовок необхідної статті, і натиснути кнопку «Видалити». Процес видалення статті зображено на рис. 3.14.



Рисунок 3.12 – Посилання на фрагмент створення і видалення
 НОВИН

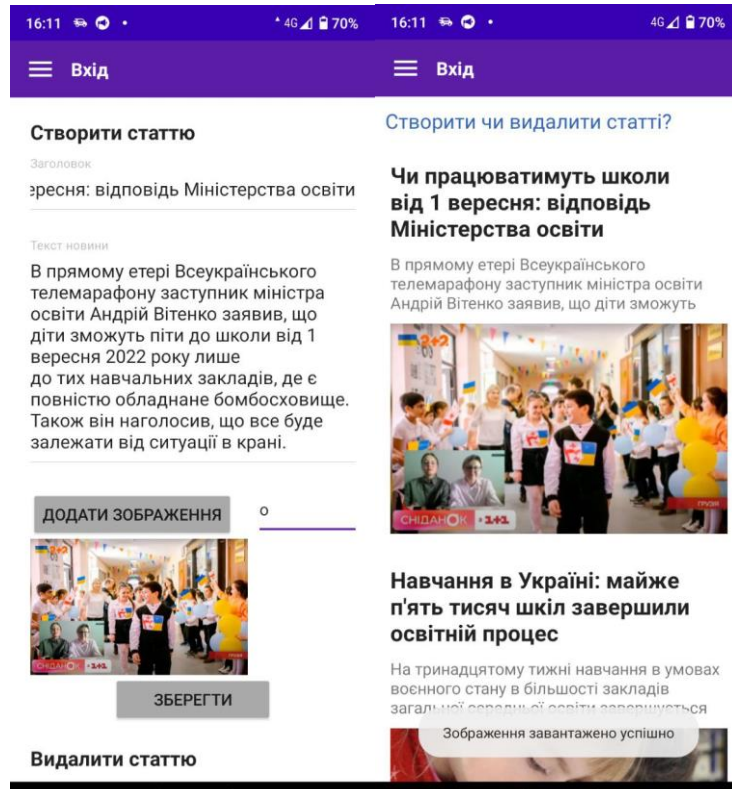


Рисунок 3.13 – Приклад створення новини

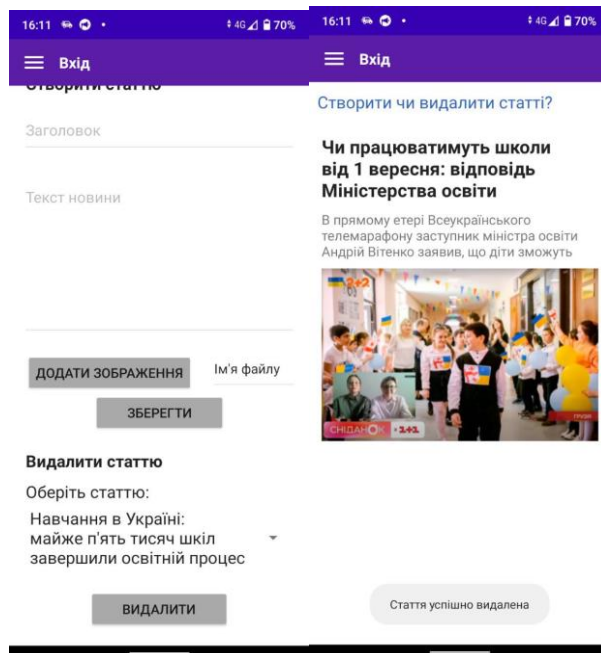


Рисунок 3.14 – Приклад видалення новини

Вчитель також може створювати нові навчальні класи, і додавати туди учнів. Після створення класу, можна запланувати для нього уроки, або видалити вже існуючі. Після цього, всі користувачі що є учасниками класу, зможуть побачити створені уроки в своєму розкладі. (рис. 3.15-3.18).

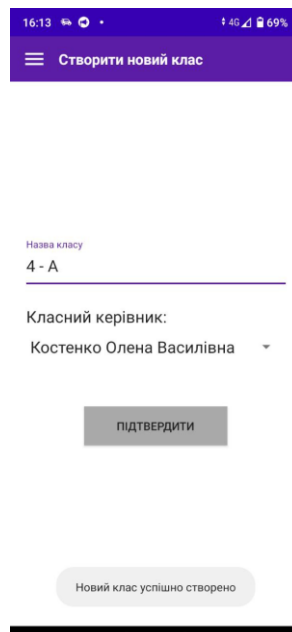


Рисунок 3.15 – Приклад створення навчального класу

16:19 4G 69%

Реєстрація користувачів

Логін
oleg1@gmail.com

Пароль
.....

Ім'я
Олег

Прізвище
Савченко

Номер телефону
380999999995

Дата народження
12.05.2010

Роль: учень

Клас 4 - A

Користувач створений

Рисунок 3.16 – Приклад створення нового учня, і призначення його в щойно створений клас

15:10 22%

Запланувати урок

Створення уроку

Назва уроку
Українська мова

Вчитель
Костенко Олена Василівна

Клас 4 - A

12.06.2022

10:35 11:20

ПІДТВЕРДИТИ

15:10 22%

Запланувати урок

Створення уроку

Назва уроку

Вчитель
Костенко Олена Василівна

Клас 4 - A

Дата

Час початку Час завершення

ПІДТВЕРДИТИ

Видалення уроку

Оберіть урок для видалення:
Я у світі 4 - A 12-06-2022

Видалення уроку

Оберіть урок для видалення:
Я у світі 12-06-2022
Урок успішно створено

Рисунок 3.17 – Приклад призначення уроку

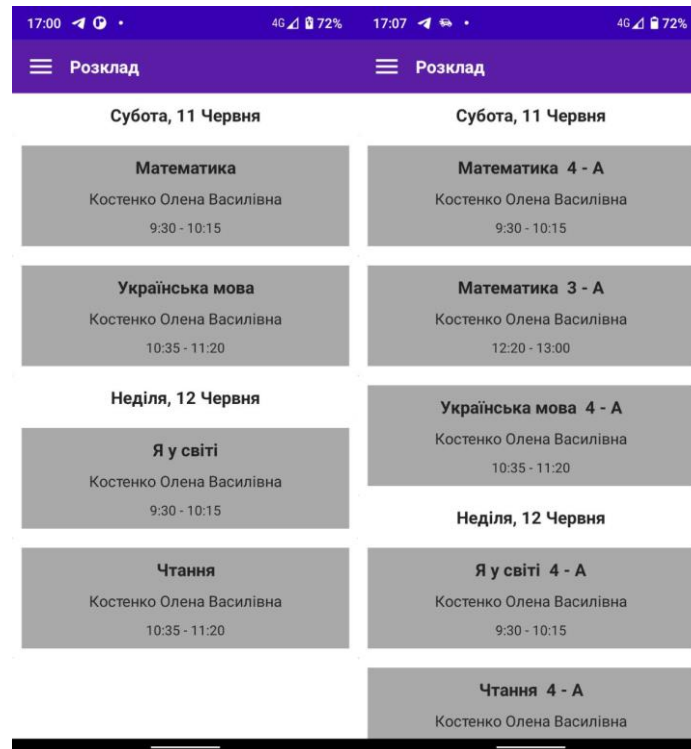


Рисунок 3.17 – Вигляд розкладу від обличчя учня та вчителя

ВИСНОВОК

У ході виконання дипломної роботи було проаналізовано предметну область, розглянуто аналоги того продукту, що має бути розроблений. За результатами аналізу предметної області та ознайомлення зі схожими програмними продуктами було підготовлено специфікацію для розробки мобільного додатку супроводження навчального процесу навчально-виховного комплексу №11.

Наступним кроком до виконання поставленої задачі стало формування мети та вибір засобів реалізації. Мета роботи – розробка мобільного додатку для супроводження навчального процесу на базі ОС Android. У якості середовища було обрано IDE Android Studio, а мовою розробки було обрано Java. Допоміжними інструментами стали Gradle та Firebase.

Під час проектування розроблюваного додатку було створено контекстну діаграму в нотації IDEF0 для демонстрації бізнес-процесів додатку, а також її декомпозицію першого рівня. Додатково була розроблена діаграма варіантів використання, яка допомогла визначити сценарії взаємодії користувачів з мобільним додатком.

Завдяки схемі бази даних, вдалось реалізувати структуру даних в базі даних реального часу Firebase. Додатково використовувались модулі Firestore та Firebase Authentication. [16]

Результатом роботи є зручний та простий мобільний додаток, який забезпечує функціонал перегляду розкладу та новин. Додаток скомпільовано у вигляді apk файлу, посилання на який можна розповсюджувати у вигляді qr-коду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. НУШ коротко | Нова українська школа. Нова українська школа [Електронний ресурс]. Режим доступу: <https://nus.org.ua/about/> (дата звернення: 10 червня 2022).
2. Digital 2022: Ukraine – DataReportal – Global Digital Insights [Електронний ресурс] // DataReportal – Global Digital Insights. – Режим доступу: <https://datareportal.com/reports/digital-2022-ukraine> (дата звернення: 24.05.2022).
3. Діджиталізація в Україні [Електронний ресурс] // Тиждень. – Режим доступу: <http://week.dp.gov.ua/osvitnia-prohrama/pislya91/digitalizaciya-v-ukraini> (дата звернення: 24.05.2022).
4. Розробка мобільних додатків від А до Я: повний гайд [Електронний ресурс] // DAN.IT education. – Режим доступу: <https://dan-it.com.ua/uk/blog/rozrobka-mobilnih-dodatkiv-vid-a-do-ja-povnij-gajd/> (дата звернення: 24.05.2022).
5. FAQ [Електронний ресурс] // Єдина Школа. – Режим доступу: <https://eschool-ua.com/#/faq/teachers> (дата звернення: 24.05.2022).
6. Guide [Електронний ресурс] // Всеукраїнська школа онлайн. – Режим доступу: <https://lms.e-school.net.ua/guide> (дата звернення: 24.05.2022).
7. Documentation [Електронний ресурс] // Android Developers. – Режим доступу: <https://developer.android.com/docs> (дата звернення: 24.05.2022).
8. Java Documentation [Електронний ресурс] // Oracle. – Режим доступу: <https://docs.oracle.com/en/java/> (дата звернення: 24.05.2022).
9. Firebase realtime database | firebase documentation. Firebase [Електронний ресурс]. Режим доступу: <https://firebase.google.com/docs/database?authuser=0&hl=en> (дата звернення 10.06.2022).

10. The complete guide to understand IDEF diagram | edrawmax online. Edrawsoft [Електронний ресурс]. – Режим доступу: <https://www.edrawmax.com/article/the-complete-guide-to-understand-idef-diagram.html> (дата звернення: 10 червня 2022).
11. Пономаренко В.С. Теорія та практика моделювання бізнес-процесів : монографія / В.С. Пономаренко, С.В. Мінухін, С.В. Знахур / М-во освіти і науки України, Харк. нац. екон. ун-т. – Харків : ХНЕУ, 2013. – 244 с
12. What is a use case diagram? [Електронний ресурс]. WhatIs.com. 24.07.2020. Режим доступу: <https://www.techtarget.com/whatis/definition/use-case-diagram> (дата звернення 10.06.2022).
13. ER diagram: entity relationship diagram model | DBMS example. Guru99 [Електронний ресурс]. Режим доступу: <https://www.guru99.com/er-diagram-tutorial-dbms.html> (дата звернення 10.06.2022)
14. MVC Design Pattern - GeeksforGeeks. GeeksforGeeks [Електронний ресурс]. Режим доступу: <https://www.geeksforgeeks.org/mvc-design-pattern/> (дата звернення 10.06.2022).
15. Архітектурний шаблон MODEL-VIEW-CONTROLLER (MVC) для побудови класів. SchoolboyProgrammer [Електронний ресурс]. Режим доступу: <https://schoolboyprog10.blogspot.com/p/mvc-mvc-model-view-controller.html> (дата звернення 10.06.2022).
16. Cloud storage for firebase | firebase documentation. Firebase [Електронний ресурс]. Режим доступу: <https://firebase.google.com/docs/storage?hl=en> (дата звернення 10.06.2022).

ДОДАТОК А
ТЕХНІЧНЕ ЗАВДАННЯ

Технічне завдання
на розробку інформаційної системи
«Мобільного додатку супроводження навчального
процесу НВК №11»

Суми 2022

1. Призначення й мета створення мобільного додатку

Призначення мобільного додатку

Інформаційна система має допомагати в організації навчального процесу за рахунок забезпечення обміну інформацією в усіх необхідних форматах між учнями, батьками і вчителями.

1.1. Мета створення мобільний додатку

Мета мобільного додатку – зробити навчальний процес більш зручним для всіх його учасників.

1.2. Цільова аудиторія

До цільової аудиторії додатку в першу чергу відносяться учні та вчителі, які будуть працювати з додатком найбільше. Додатково до цільової аудиторії можна віднести адміністрацію школи та батьків учнів.

2. Вимоги до мобільного додатку

2.1. Вимоги до мобільного додатку в цілому

2.1.1. Вимоги до структури й функціонування мобільного додатку

Мобільний додаток супроводження навчального процесу НВК №11 має бути реалізований у середовищі Android Studio, і повинен реалізовувати визначений набір функціональних можливостей.

2.1.2. Вимоги до персоналу

Від персоналу не має вимагатися особливих технічних навичок для підтримки й експлуатації додатку. Весь функціонал, необхідний для наповнення додатку необхідним контентом буде реалізовано для користувачів, що мають відповідні ролі. У разі необхідності доповнення

мобільного додатку новим функціоналом необхідно залучення спеціаліста зі знанням мови програмування Java та Android SDK.

2.1.3. Вимоги до збереження інформації

Уся інформація буде зберігати в базі даних в реальному часі Firebase.

2.1.4. Вимоги до розмежування доступу

Розроблюваний мобільний додаток буде розповсюджуватися лише всередині навчального закладу.

Додаток матиме систему обліку користувачів та розподіл прав між ними. При цьому, всього буде виділено чотири типи користувачів: батьки, учні, вчителі, адміністратор.

Користувач, що не увійшов до облікового запису не має ролі, і може переглядати лише новини закладу.

Учні та батьки також зможуть переглядати новини закладу. Крім цього, їм буде доступний перегляд розкладу, але лише свого класу. Аккаунти учнів і батьків створюються, і вони зможуть лише змінити пароль до нього.

Вчителі та адміністратори зможуть створювати статті з новинами, чи видаляти застарілі, створювати нові облікові записи, створювати нові навчальні класи, переглядати розклад будь-якого класу.

2.2. Структура мобільного додатку

2.2.1. Загальна інформація про структуру мобільного додатку

Структура мобільного додатку являє собою набір екранів з функціональними модулями.

Такими розділами є:

Новини – екран з новинами та оголошеннями.

Розклад – інформація про розклад уроків класу користувача.

Профіль – інформація про користувача.

Увійти – вхід до облікового запису.

Реєстрація – створення нового користувача.

Створення класу – створення навчального класу.

Запланувати урок – запланувати урок з тим, чи іншим класом.

2.2.2. Навігація

Навігація між основними модулями додатку буде відбуватися за допомогою бокового меню в лівій частині екрану. Додатково буде створена навігація між другорядними екранами за допомогою динамічних елементів, таких як кнопки та посилання.

2.2.3. Дизайн та структура додатку

Стиль мобільного додатку має бути приємним для ока, без різких кольорів. У якості основних кольорів використовувати білий, фіолетовий та сірий, так як саме в таких кольорах виконаний вже існуючий web-додаток школи.

Інтерфейс має бути зручним та зрозумілим звичайному користувачу з поверхневими навичками володіння мобільними пристроями, сучасним, приємним для сприйняття.

2.3. Вимоги до функціонування системи

2.3.1. Потреби користувача

Потреби користувача представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ІД	Потреби користувача	Джерело
UN-01	Перегляд новин закладу	Будь-який користувач
UN-02	Перегляд розкладу свого класу.	Батьки, учень
UN-03	Перегляд розкладу будь-якого класу	Вчитель, адміністратор
UN-04	Реєстрація нових користувачів	Вчитель, адміністратор
UN-05	Створення нових статей новин	Вчитель, адміністратор
UN-06	Створення нових навчальних класів	Вчитель, адміністратор
UN-07	Планування уроків	Вчитель, адміністратор
UN-08	Зміна паролю облікового запису	Будь-який авторизований користувач
UN-09	Доступ до бази даних	Адміністратор

2.3.2. Функціональні вимоги

На основі потреб користувача були визначені такі функціональні вимоги:

- реєстрація та авторизація користувачів;
- блоки з новинами;
- перегляд розкладу;
- модуль створення та видалення новин;
- модуль створення нових навчальних класів;
- модуль планування уроків;
- можливість зміни паролю облікового запису;

2.3.3. Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги,

визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Модуль перегляду новин	М	Надає можливість перегляду новин.
SR-02	Модуль з розкладом уроків	М	Перегляд розкладу уроків для конкретного класу, чи за вибором.
SR-03	Реєстрація користувачів	М	Надає можливість створення нових облікових записів
SR-04	Модуль планування уроків	М	Дозволяє вчителям додавати уроки в розклад учнів.
SR-05	Модуль створення навчальних класів	S	Дозволяє вчителям створювати нові навчальні класи.
SR-06	Додавання нових статей з новинами та видалення старих	S	Дозволяє вчителям та адміністраторам керувати наповненням контентом розділу з новинами закладу.
SR-07	Зміна паролю облікового запису	C	Дозволяє користувачам встановлювати власний пароль облікового запису, замість призначеного вчителем

Умовні позначення в таблиці А.2:

- Must have (M) – вимоги, які повинні бути реалізовані в системі;
- Should have (S) – вимоги, які мають бути виконані, але вони

можуть почекати своєї черги;

- Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4.Вимоги до видів забезпечення

2.4.1. Вимоги до інформаційного забезпечення

Реалізація мобільного додатку відбувається з використанням:

- Java 8 – для реалізації логіки бізнес-процесів додатку;
- Android 10 SDK – набір інструментів для розробки додатків для смартфонів на базі ОС Android;
- Firebase – для збереження інформації в базі даних, реєстрації та авторизації користувачів;
- Gradle 7.3.3 – система збірки проекту.

2.4.2. Вимоги до лінгвістичного забезпечення

Мобільний додаток має бути виконаний українською мовою.

2.4.3. Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Операційна система: мінімальна – Android 7 і вище, рекомендована – Android 10 і вище.

3. Склад і зміст робіт зі створення мобільного додатку

Докладний опис етапів роботи зі створення мобільного додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення мобільного додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Постановка цілей необхідних для досягнення певного результату	6 днів
2	Складання технічного завдання	6 дні
3	Підготовка прототипу	4 дні
4	Програмна реалізація інтерфейсу	10 днів
5	Реалізація основних модулів бізнес-логіки	37 днів
6	Реалізація з'єднання з базою даних	6 днів
7	Наповнення контентом	8 днів
8	Тестування додатку	12 днів
9	Впровадження в дію	13 днів
	Загальна тривалість робіт	107 днів

3.1.Вимоги до складу й змісту робіт із введення мобільного додатку в експлуатацію

Для того, щоб додатком могли користуватись учасники навчального процесу НВК №11 необхідно створити інсталяційний арк файл, та забезпечити його подальше поширення у межах навчального закладу. База даних Firebase не потребує розміщення на сервері, та є доступною в будь-який момент за наявності інтернет з'єднання.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

Інформаційні технології стрімко розвиваються і проникають в усі сфери людської діяльності. Галузь освіти не стала винятком. Сучасне навчання значною мірою змінилося у порівнянні з тим, що було десять чи двадцять років тому. Однією з найбільш важливих змін є активне впровадження інформаційних технологій в навчальний процес. Колись було важко уявити, що навчатися повноцінно можна без відвідування закладів освіти, просто сидячи вдома, а зараз це – буденна справа. Міністерство освіти наразі активно працює над реформою нової української школи. Одним із головних принципів НУШ є запровадження в освітній галузі використання інформаційно-комунікаційних технологій на постійній основі для розширення можливостей педагога.

Мобільний додаток супроводження навчального процесу якраз і є одним із вищеописаних спеціалізованих інструментів. На відміну від більшості технологій, що зараз використовуються у навчальному процесі, він буде направлений лише на використання в навчанні. Такий підхід означає що він не матиме зайвих та відволікаючих функцій, що допоможе більш якісно опрацьовувати інформацію.

Деталізація мети проекту методом SMART. Визначення мети проекту за допомогою SMART-методу допомагає чітко визначити призначення кінцевого продукту та обмеження під час його розробки. Крім того, мета, сформована за цим методом дає можливість визначити напрямок, у якому необхідно рухатись для її досягнення. Результати деталізації методом SMART розміщені у таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific (конкретна)	Створити мобільний додаток супроводження навчального процесу НВК №11
Measurable (вимірювана)	Розробити мобільний додаток за 3 місяці, використовуючи мінімум ресурсів.
Achievable (досяжна, узгоджена)	Для виконання проекту наявні необхідні знання мови програмування Java, середовища Android Studio, баз даних MySQL та навичок написання документації. Враховуючи доступні ресурсні можливості та обмеження мета є такою, яку можливо досягти.
Relevant (реалістична)	Створений мобільний додаток спростить та пришвидшить взаємодію між вчителем та учнями, дозволить використовувати у навчальному процесі нові форми активності.
Time-framed (обмежена в часі)	Ціль має часове обмеження. Термін досягнення мети проекту визначено за домовленістю між замовником та виконавцем і складає три місяці.

Планування змісту робіт. WBS (Work Breakdown Structure – Ієрархічна структура робіт) – це графічне розбиття проекту на елементарні задачі, які пов’язані за ієрархією з фінальним продуктом проекту. Структура декомпозиції робіт дозволяє наглядно зобразити весь об’єм проекту, спростити організацію командної роботи, дозволяє коректно оцінити затрати та час, необхідний для виконання проекту. Крім того, WBS дозволяє структурувати та розділити проект на основні компоненти, якими легко керувати.

На найвищому (першому) рівні розміщений продукт проекту. Основні компоненти, що забезпечують досягнення мети проекту, розташовуються на другому рівні декомпозиції. Декомпозиція робіт виконується до тих пір, поки вони не стануть елементарними.

Елементарні роботи – це дії, які мають однозначний чіткий результат, відповідальність за які лежить на одній конкретній особі, для яких можна

обчислити необхідні витрати ресурсів. На рисунку Б.1 представлено WBS з розробки мобільного додатку супроводження навчального процесу.

В результаті проведення декомпозиції проекту було отримано п'ять основних напрямів діяльності – підготовка документації, розробка мобільного додатку, наповнення його контентом, тестування та введення в реалізацію. Кожен з цих напрямів має свої підзадачі.

На етапі підготовки документації відбувається збір інформації про предметну область продукту, аналіз вимог до нього та розробка чіткого й зрозумілого ТЗ, після чого починається розробка. Під час етапу розробки визначається інтерфейс майбутнього додатку, програмно реалізується його функціонал, та налаштовується взаємодія з базою даних. Коли продукт вже отримав весь необхідний функціонал, відбувається його наповнення даним, після чого відбувається стадія тестування, під час якої важливо від лица потенційного користувача перевірити працездатність всіх його модулів. Якщо ж стадія тестування пройшла успішно, і значних недоліків знайдено не було, то продукт переходить на етап впровадження в дію і в результаті потрапляє до замовника.

Наступним етапом після декомпозиції процесів є розробка організаційної структури виконавців або OBS. Для успішного управління проектом необхідно знати яка організація чи виконавець відповідні за кожен пакет чи рівень робіт. Саме для цього і використовується OBS. В цій структурі на верхньому рівні знаходиться керівник проекту, а на нижчих організаціях розташовані організації, відділи, чи працівники що відповідають за конкретні роботи чи пакети робіт з WBS-структури.

На рисунку Б.2 представлено організаційну структуру планування проекту.

Список виконавців, що функціонують в проекті описано в таблиці Б.2.

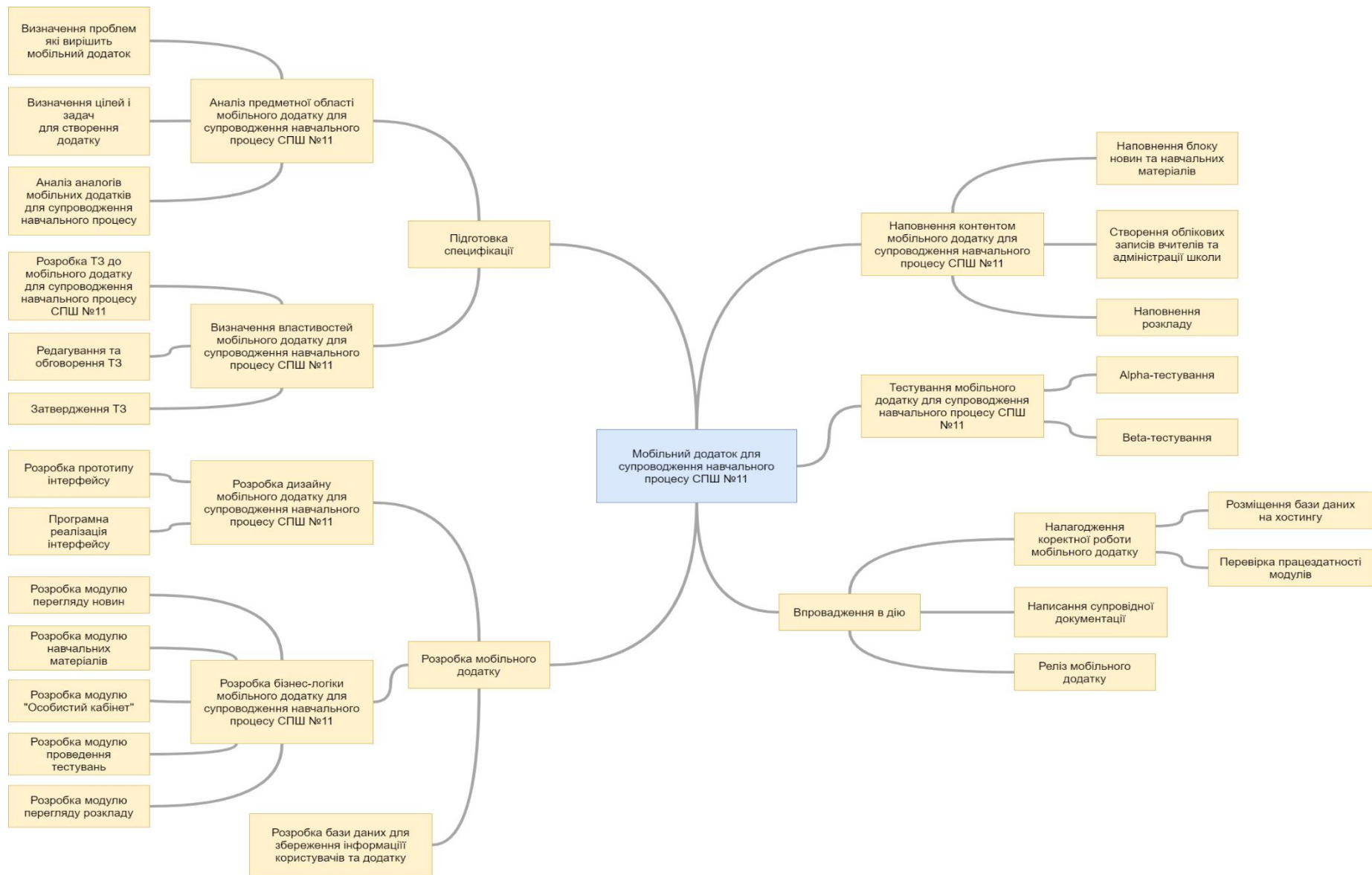


Рисунок Б.1 – WBS-структура робіт проекту

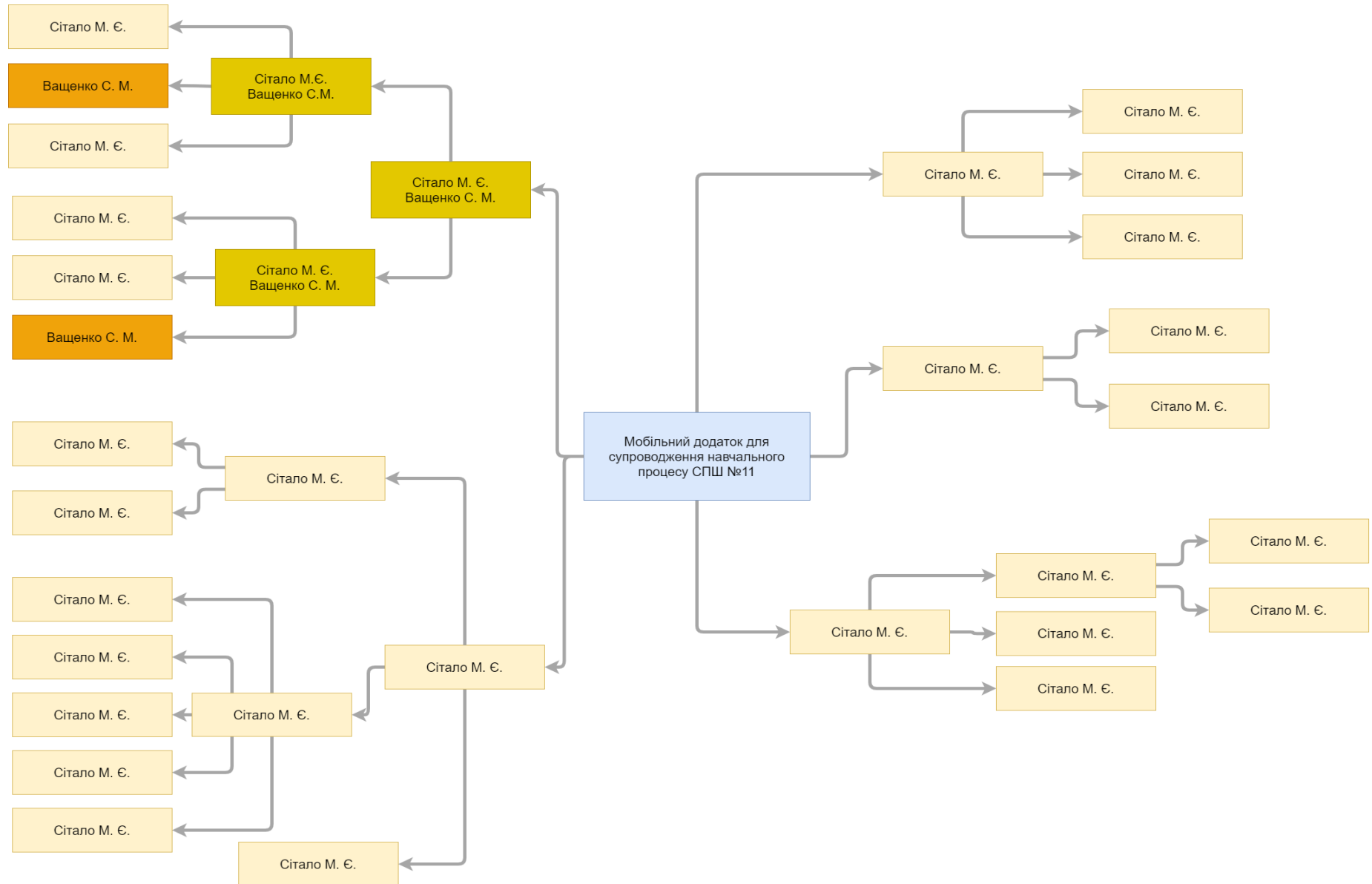


Рисунок Б.2 – OBS-структура робіт проекту

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Сітало М.Є.	Виконує розробку програмних модулів та дизайну додатку.
Проектувальник	Сітало М.Є.	Виконує проектування бази даних та розробляє структуру мобільного додатку.
Тестувальник	Сітало М.Є.	Відповідає за тестування функціоналу та дизайну додатку.
Керівник проекту	Ващенко С. М.	Формує завдання на розробку проекту.
Менеджер проекту	Сітало М.Є.	Відповідає за дотримання термінів, розподіл ресурсів та завдань між учасниками, написання документації. Виконує збір та аналіз даних.

Для відстеження ходу виконання проекту дуже зручно використовувати його календарний план (діаграму Ганта). Вона дозволяє в простій і зрозумілій формі зобразити структуру проекту, відповідальних за кожне завдання осіб, часові обмеження стосовно кожної задачі.

Календарний графік проекту представлено на рисунках Б.3-Б.5.

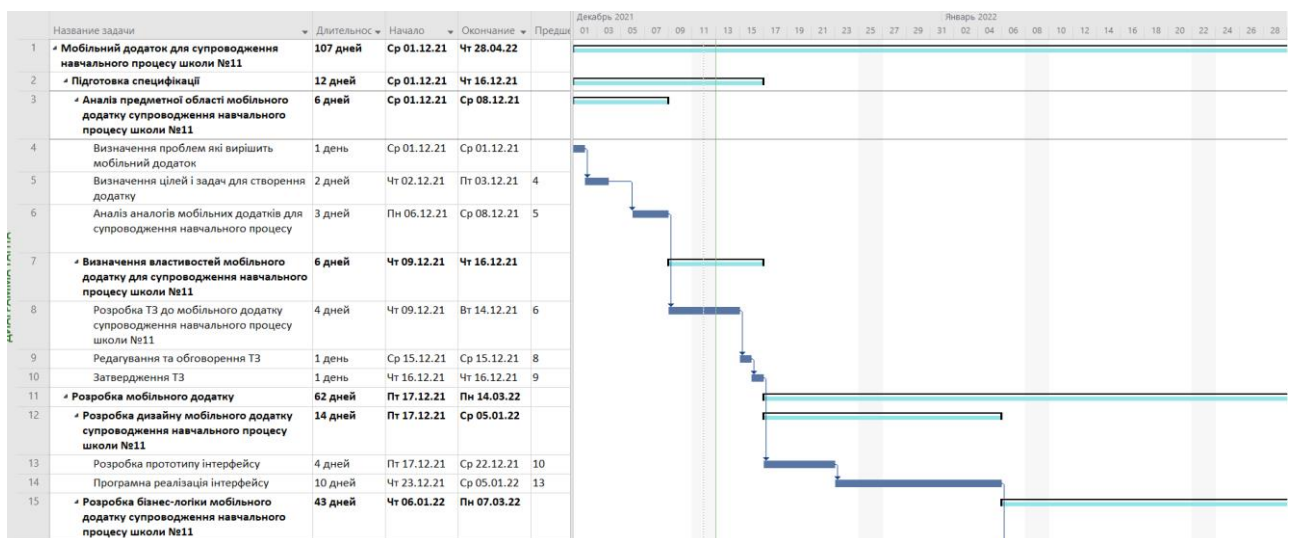


Рисунок Б.3 – Календарний графік проекту, частина 1

Таблиця Б.3 – Ідентифікація ризиків

№	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Неправильно сформоване ТЗ проекту	0,5	0,5	0,25
2	Непорозуміння між розробником та замовником	0,4	0,5	0,2
3	Нечіткі та неточні вимоги від замовника	0,6	0,6	0,36
4	Недостатній рівень знань розробника	0,2	0,6	0,12
5	Нераціональний розподіл часу	0,4	0,6	0,24
6	Часте редагування ТЗ	0,3	0,3	0,09
7	Створення незрозумілого користувачеві інтерфейсу	0,3	0,5	0,15
8	Некоректна робота одного із програмних модулів	0,2	0,4	0,08
9	Несправності технічного обладнання	0,1	0,7	0,07
10	Невірна оцінка масштабу проекту	0,3	0,4	0,12

Для того, щоб оцінити небезпеку потенційних буде використано матрицю ймовірності та ризику. Таке оцінювання дозволить розділити існуючі ризики на прийнятні, виправдані та недопустимі, що в свою чергу

дозволить розробити план дій на випадок виникнення того чи іншого ризику. Прийнятні ризики не завдають значної шкоди в масштабах проекту, тому їх можна просто ігнорувати. Виправдані ризики завдають помітну шкоду, але при додаткових стратегіях можна уникнути значних збитків. Недопустимі ризики притягнуть за собою надзвичайні втрати ресурсів, тому необхідно будь-якими шляхами знизити їх ймовірність, або зменшити вплив на проект. Розподілення ризиків за їх впливом та імовірністю показано у таблиці Б.4. Шкала оцінювання за ризиками представлена у таблиці Б.5. Заходи реагування на виявлені ризики в проекті наявні у таблиці Б.6.

Таблиця Б.4 – Розподілення ризиків за їх впливом і ймовірністю

Ймовірність	Вплив загрози(ризикау)				
	Дуже малий 0,05	Малий 0,1	Середній 0,2	Великий 0,4	Дуже великий 0,8
0,9					
0,7				R3(0,36)	
0,5			R2(0,2) R1(0,25)		
0,3	R6(0,09)	R7(0,15) R12(0,12)	R5(0,24)		
0,1	R8(0,08) R9(0,07)	R4(0,12)			

Таблиця Б.5 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	$0,005 \leq R \leq 0,05$	1, 4, 6, 7, 8, 9, 10
2	Виправдані	$0,05 < R \leq 0,14$	2, 3, 5
3	Недопустимі	$0,14 < R \leq 0,72$	

Таблиця Б.6 - Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
1	Відкритий	Неправильно сформоване ТЗ проекту	Висока	Середній	2	Визначити технічні аспекти та умови реалізації продукту, його цільове призначення	Ухилення	Погодити всі проблеми та шляхи їх вирішення з керівником
2	Відкритий	Непорозуміння між розробником та замовником	Середня	Середній	2	Налагодити гарні відносини, дотримуватися ділового етикету.	Попередження	Необхідно знайти та усунути причину непорозуміння
3	Відкритий	Нечіткі та неточні вимоги від замовника	Висока	Великий	3	Визначити всі необхідні параметри проекту, чіткі вимоги	Перенесення	Уточнити всі деталі у замовника, за необхідності внести правки
4	Відкритий	Недостатній рівень знань розробника	Низька	Середній	1	Обирати знайомі технології та засоби	Пом'якшення	Виділити час на підвищення кваліфікації робітників

Продовження табл. Б.6

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
5	Відкритий	Нераціональний розподіл часу	Середня	Середній	2	Чітке дотримання термінів, визначених на етапі планування	Зменшення	Виділення додаткового резерву часу, більш ретельне дотримання термінів
6	Відкритий	Часте редагування ТЗ	Низька	Низький	1	Дізнатися всі деталі та чіткі вимоги по кожній задачі	Зменшення	Дізнатися причину змін, та усунути її, замість наслідків
7	Відкритий	Створення незрозумілого користувачеві інтерфейсу	Низька	Середній	1	Створити звичний для типового користувача інтерфейс, додати пояснення	Ухилення	Зробити підказки в інтерфейсі або прибрати чи переробити функції, з якими виникають проблеми
8	Відкритий	Некоректна робота одного із програмних модулів	Низька	Низький	1	Провести детальне тестування	Ухилення	Пошук проблеми та внесення змін для її видалення

Продовження табл. Б.6

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
9	Відкритий	Несправності технічного обладнання	Низька	Середній	1	Підготовка додаткової техніки, яка у разі поломки зможе замінити зламану	Перенесення	Здати несправне обладнання на ремонт
10	Відкритий	Невірна оцінка масштабу проекту	Низька	Низький	1	Провести детальний аналіз проекту, розумно розподілити час на їх виконання, залишивши невеликий резерв.	Пом'якшення	Повторна оцінка проекту, зміна плану

ДОДАТОК В

ЛІСТИНГ ПРОГРАМНОГО КОДУ ОСНОВНИХ МОДУЛІВ МОБІЛЬНОГО ДОДАТКУ

MainActivity.java

```
package com.maxim328.zhuravonka.Controllers;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.NavigationUI;

import android.os.Bundle;
import android.view.MenuItem;
import android.widget.TextView;

import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.maxim328.zhuravonka.Model.StaticMethods;
import com.maxim328.zhuravonka.Model.User;
import com.maxim328.zhuravonka.R;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        DrawerLayout drawerLayout = findViewById(R.id.drawerLayout);

        findViewById(R.id.imageMenu).setOnClickListener(v ->
drawerLayout.openDrawer(GravityCompat.START));

        NavigationView navigationView = findViewById(R.id.navigationView);
        navigationView.setItemIconTintList(null);
```



```

        NavController navController = Navigation.findNavController(this,
R.id.navHostFragment);

        FirebaseAuth auth = FirebaseAuth.getInstance();
        if (auth.getCurrentUser() != null &&
!auth.getCurrentUser().isAnonymous()) {
            DatabaseReference usersDB = FirebaseDatabase
                .getInstance("https://zhuravonka-12a93-default-rtdb.europe-
west1.firebaseio.com")
                .getReference("users");
            usersDB.child(auth.getCurrentUser().getUid()).get()
                .addOnCompleteListener(task -> {
                    if (task.isSuccessful()) {
                        User currentUser =
task.getResult().getValue(User.class);
                        if (currentUser == null) {

navigationView.getMenu().findItem(R.id.menuCreateLesson).setVisible(false);

navigationView.getMenu().findItem(R.id.menuCreateClass).setVisible(false);

navigationView.getMenu().findItem(R.id.menuProfile).setVisible(false);

navigationView.getMenu().findItem(R.id.menuRegister).setVisible(false);

navigationView.getMenu().findItem(R.id.menuSchedule).setVisible(false);

navigationView.getMenu().findItem(R.id.menuSignOut).setVisible(false);
                            return;
                        }
                        else if (currentUser.getRole().equals("0") ||
currentUser.getRole().equals("1")) {

navigationView.getMenu().findItem(R.id.menuCreateLesson).setVisible(false);

navigationView.getMenu().findItem(R.id.menuCreateClass).setVisible(false);

navigationView.getMenu().findItem(R.id.menuRegister).setVisible(false);
                            }

((TextView)navigationView.findViewById(R.id.userNameHeader)).setText(StaticMethods.get
TeacherName(currentUser));

```

```

        ((TextView) navigationView.findViewById(R.id.userEmailHeader)).setText(currentUser.getEmail());
    }

    });
}
else {

navigationView.getMenu().findItem(R.id.menuCreateLesson).setVisible(false);

navigationView.getMenu().findItem(R.id.menuCreateClass).setVisible(false);

navigationView.getMenu().findItem(R.id.menuProfile).setVisible(false);

navigationView.getMenu().findItem(R.id.menuRegister).setVisible(false);

navigationView.getMenu().findItem(R.id.menuSchedule).setVisible(false);

navigationView.getMenu().findItem(R.id.menuSignOut).setVisible(false);
}

navigationView.getMenu().findItem(R.id.menuSignOut).setOnMenuItemClickListener(item ->
{
    FirebaseAuth.getInstance().signOut();

    ((TextView) navigationView.findViewById(R.id.userNameHeader)).setText("Таємничий незнайомець");

    ((TextView) navigationView.findViewById(R.id.userEmailHeader)).setText("");

navigationView.getMenu().findItem(R.id.menuSignOut).setVisible(false);
    navigationView.getMenu().findItem(R.id.menuLogin).setVisible(true);

navigationView.getMenu().findItem(R.id.menuCreateLesson).setVisible(false);

navigationView.getMenu().findItem(R.id.menuCreateClass).setVisible(false);

navigationView.getMenu().findItem(R.id.menuProfile).setVisible(false);

navigationView.getMenu().findItem(R.id.menuRegister).setVisible(false);

navigationView.getMenu().findItem(R.id.menuSchedule).setVisible(false);
    return true;
}

```

```

    });

    NavigationUI.setupWithNavController(navigationView, navController);

    TextView textTitle = findViewById(R.id.textTitle);
    navController.addOnDestinationChangeListener((navController1,
navDestination, bundle) -> textTitle.setText(navDestination.getLabel()));
    }

}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Controllers.MainActivity">
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:id="@+id/layoutToolBar"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:background="@color/colorPrimary"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:paddingStart="15dp"
        android:paddingEnd="15dp"
        app:layout_constraintTop_toTopOf="parent">
    <ImageView
        android:id="@+id/imageMenu"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:contentDescription="@string/app_name"
        android:src="@drawable/ic_menu"
        app:tint="@color/colorWhite"/>
    <TextView
        android:id="@+id/textTitle"
        android:layout_width="match_parent"

```



```

import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.maxim328.zhuravonka.Model.ChildrenClass;
import com.maxim328.zhuravonka.Model.DayLessons;
import com.maxim328.zhuravonka.Model.Lesson;
import com.maxim328.zhuravonka.Model.ScheduleAdapter;
import com.maxim328.zhuravonka.Model.User;
import com.maxim328.zhuravonka.R;

import java.time.LocalDate;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collection;
import java.util.Comparator;
import java.util.Date;
import java.util.HashMap;
import java.util.List;

public class ScheduleFragment extends Fragment {
    private String currentUserClass;
    private String currentUserRole;
    private int pos = -2;

    public ScheduleFragment() {
        // Required empty public constructor
    }

    public static ScheduleFragment newInstance(String param1, String param2) {
        ScheduleFragment fragment = new ScheduleFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

```



```

0);

ArrayList<>();

task1.getResult().getValue();

ArrayList<>();

map = ((HashMap) v).values();

lesson:map) {

    || currentUserRole.equals("3") {

lessonItem.getTitle();

classesKeys.get(lessonItem.getClassName());

(currentUserClass.equals(lessonItem.getClassName())) {

((String)k).split("-");

c.set(Integer.parseInt(dateStrings[2]),

Integer.parseInt(dateStrings[1]) - 1,

Integer.parseInt(dateStrings[0]));

currentDate.set(Calendar.HOUR_OF_DAY,

currentDate.set(Calendar.MINUTE, 0);
currentDate.set(Calendar.SECOND, 0);
List<DayLessons> lessons = new

HashMap result1 = (HashMap)

result1.forEach((k, v) -> {
    List<Lesson> lessonsDay = new

Collection<HashMap<String, String>>

for (HashMap<String, String>

    Lesson lessonItem = new Lesson(
        lesson.get("title"),
        lesson.get("teacher"),
        lesson.get("className"),
        lesson.get("startTime"),
        lesson.get("endTime"));
    if(currentUserRole.equals("2")

        String title =

        title += " ";
        title +=

        lessonItem.setTitle(title);
        lessonsDay.add(lessonItem);
    } else if

        lessonsDay.add(lessonItem);
    }

}

Calendar c = Calendar.getInstance();
String[] dateStrings =

```

```

        if(lessonsDay.size() > 0 &&
!c.before(currentDate)) {
            lessonsDay.sort((o1, o2) -> {
                String[] o1Time =
o1.getStartTime().split(":");
                String[] o2Time =
o2.getStartTime().split(":");
                int hours_compare =
Integer.compare(Integer.parseInt(o1Time[0]), Integer.parseInt(o2Time[0]));
                if (hours_compare != 0)
                    return
Integer.compare(Integer.parseInt(o1Time[1]), Integer.parseInt(o2Time[1]));
                });
            lessons.add(new DayLessons(c,
lessonsDay));
        }
    });

    if (lessons.size() > 0) {
lessons.sort(Comparator.comparing(DayLessons::getDate));
        RecyclerView recyclerView =
mainView.findViewById(R.id.mainLessonRecyclerView);
        LinearLayoutManager layoutManager =
new LinearLayoutManager(getContext());
        ScheduleAdapter adapter = new
ScheduleAdapter(getContext(), lessons);
        recyclerView.setAdapter(adapter);

recyclerView.setLayoutManager(layoutManager);
    } else {
        Toast.makeText(getContext(), "Для
вашого класу ще не заплановано уроків", Toast.LENGTH_LONG).show();
    }
}
});
}
});
}
return mainView;
}

```



```

private void setCurrentUserClass(String currentUserRole) {
    this.currentUserClass = currentUserRole;
}

private void setCurrentUserRole(String currentUserRole) {
    this.currentUserRole = currentUserRole;
}
}

```

fragment_schedule.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Controllers.ScheduleFragment">

    <LinearLayout
        android:id="@+id/scheduleLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <LinearLayout
            android:id="@+id/classSelectionLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center"
            android:visibility="gone">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/classTitle"
                android:textSize="20sp"
                android:textColor="@color/colorPrimaryText"
                android:textColorHint="@color/colorPrimaryDark"
                android:layout_marginTop="15dp"
                android:layout_marginBottom="20dp"
                android:layout_marginEnd="40dp"/>

            <Spinner

```

```

        android:id="@+id/classSelectionSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="20dp"
        android:contentDescription="@string/roleTitle"
        android:minHeight="20dp"/>

</LinearLayout>

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/mainLessonRecyclerView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

</LinearLayout>
</ScrollView>

```

NewsFragmet.java

```

package com.maxim328.zhuravonka.Controllers;

import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.maxim328.zhuravonka.Model.Article;
import com.maxim328.zhuravonka.Model.NewsAdapter;

```

```

import com.maxim328.zhuravonka.Model.User;
import com.maxim328.zhuravonka.R;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class NewsFragment extends Fragment {
    private RecyclerView recyclerView;
    private NewsAdapter adapter;

    private DatabaseReference db;
    FirebaseAuth auth;
    private List<Article> articles;

    public NewsFragment() {
        // Required empty public constructor
    }

    public static NewsFragment newInstance(String param1, String param2) {
        NewsFragment fragment = new NewsFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_news, container, false);
        recyclerView = view.findViewById(R.id.recyclerViewNews);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));

        articles = new ArrayList<>();

        if (FirebaseAuth.getInstance().getCurrentUser() != null &&
!FirebaseAuth.getInstance().getCurrentUser().isAnonymous()) {
            DatabaseReference usersDB = FirebaseDatabase

```

```

        .getInstance("https://zhuravonka-12a93-default-rtdb.europe-
west1.firebaseio.com")
        .getReference("users");

usersDB.child(FirebaseAuth.getInstance().getCurrentUser().getUid()).get()
        .addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                User currentUser =
task.getResult().getValue(User.class);
                if (currentUser == null) return;
                if (currentUser.getRole().equals("2") ||
currentUser.getRole().equals("3")) {
                    ((TextView)
view.findViewById(R.id.editArticlesLink)).setVisibility(View.VISIBLE);
                } else
                    ((TextView)
view.findViewById(R.id.editArticlesLink)).setVisibility(View.GONE);
            }
        });
    });
}

((TextView)view.findViewById(R.id.editArticlesLink)).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FragmentManager manager = getParentFragmentManager();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.replace(R.id.navHostFragment,
CreateNewsArticleFragment.class, null);
        transaction.commit();
    }
});

auth = FirebaseAuth.getInstance();
if (auth.getCurrentUser()==null) auth.signInAnonymously();

db = FirebaseDatabase
        .getInstance("https://zhuravonka-12a93-default-rtdb.europe-
west1.firebaseio.com")
        .getReference("articles");

db.addValueEventListener(new ValueEventListener() {
    @Override

```

```

        public void onDataChange(@NonNull DataSnapshot snapshot) {
            for (DataSnapshot dataSnapshot: snapshot.getChildren()) {
                Article article = dataSnapshot.getValue(Article.class);
                articles.add(article);
            }
            Collections.reverse(articles);
            adapter = new NewsAdapter(getContext(), articles);

            recyclerView.setAdapter(adapter);
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            Toast.makeText(getContext(), error.getMessage(),
                Toast.LENGTH_LONG).show();
        }
    });
    return view;
}
}

```

fragment_news.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Controllers.NewsFragment"
    android:orientation="vertical">

    <TextView
        android:id="@+id/editArticlesLink"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Створити чи видалити статті?"
        android:textSize="20sp"
        android:padding="5dp"
        android:layout_margin="5dp"
        android:textColor="@color/colorTextLinks"
        android:visibility="gone"/>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerViewNews"
        android:layout_width="match_parent"

```

```

        android:layout_height="match_parent"
        android:layout_below="@id/editArticlesLink"/>

</RelativeLayout>

```

CreateNewsArticle.java

```

package com.maxim328.zhuravonka.Controllers;

import static android.app.Activity.RESULT_OK;

import android.content.ContentResolver;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.widget.AppCompatButton;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.MimeTypeMap;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.maxim328.zhuravonka.Model.Article;
import com.maxim328.zhuravonka.R;
import com.rengwuxian.materialedittext.MaterialEditText;

```

```

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;

public class CreateNewsArticleFragment extends Fragment {
    private static final int PICK_IMAGE_REQUEST = 1;
    private Button uploadButton, confirmButton;
    private MaterialEditText titleEditText, textEditText, imageTextField;
    RelativeLayout imageUploadLayout;
    private ImageView imagePreview;

    private StorageReference storageReference;
    private DatabaseReference db;
    private FirebaseAuth dbAuth;
    private Spinner spinner;

    private ContentResolver cr;

    private Uri imageUri;

    List<Article> articles;
    List<String> dbKeys;

    public CreateNewsArticleFragment() {
        // Required empty public constructor
    }

    public static CreateNewsArticleFragment newInstance() {
        CreateNewsArticleFragment fragment = new CreateNewsArticleFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        db = FirebaseDatabase
            .getInstance("https://zhuravonka-12a93-default-rtdb.europe-
west1.firebaseio.com")
            .getReference("articles");

        articles = new ArrayList<>();

```

```

db.get().addOnCompleteListener(task -> {
    if (task.isSuccessful()) {
        DataSnapshot dataSnapshot = task.getResult();
        Object o = dataSnapshot.getValue();
        HashMap<String, Object> result = (HashMap<String, Object>) o;
        articles = new ArrayList<>();
        dbKeys = new ArrayList<>();
        if (result == null) return;
        result.forEach((k, v) -> {
            HashMap<String, String> map = (HashMap<String, String>) v;
            Article currentArticle = new Article(
                map.get("title"),
                map.get("text"),
                map.get("name"),
                map.get("uri"));
            articles.add(currentArticle);
            dbKeys.add(k);
        });
        Collections.reverse(articles);
        Collections.reverse(dbKeys);
        List<String> articlesString = new ArrayList<>();
        for (int i = 0; i < articles.size(); i++) {
            articlesString.add(articles.get(i).getTitle());
        }
        ArrayAdapter<String> adapter = new ArrayAdapter<>(getContext(),
R.layout.spinner_item, articlesString);
        adapter.setDropDownViewResource(R.layout.spinner_item);
        spinner.setAdapter(adapter);
    }
    else Toast.makeText(getContext(), "Помилка зеднання з базою даних.",
Toast.LENGTH_LONG).show();
});

storageReference =
FirebaseStorage.getInstance().getReference("uploads");
dbAuth = FirebaseAuth.getInstance();
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_create_news_article,
container, false);

```



```

uploadButton = view.findViewById(R.id.uploadImageButton);
confirmButton = view.findViewById(R.id.confirmImageUploadButton);
titleEditText = view.findViewById(R.id.newsArticleTitle);
textEditText = view.findViewById(R.id.newsArticlePrimaryText);
imageTextField = view.findViewById(R.id.imageUploadedFilename);
imagePreview = view.findViewById(R.id.imagePreview);
imageUploadLayout = view.findViewById(R.id.imageUploadLayout);
cr = view.getContext().getContentResolver();

uploadButton.setOnClickListener(v -> openFileChooser());

confirmButton.setOnClickListener(v -> uploadImage(getContext()));

List<String> articlesString = new ArrayList<>();
for (int i = 0; i < articles.size(); i++) {
    articlesString.add(articles.get(i).getTitle());
}

spinner = (Spinner) view.findViewById(R.id.articleSelectionSpinner);

AppCompatActivity deleteArticleButton =
view.findViewById(R.id.confirmDeletingNewsButton);
deleteArticleButton.setOnClickListener(v -> {
    if (spinner.getSelectedItemPosition() == -1) {
        Toast.makeText(getContext(), "Неможливо видалити статтю: статті
відсутні", Toast.LENGTH_LONG).show();
        return;
    }
    db.child(dbKeys.get(spinner.getSelectedItemPosition()))
        .removeValue().addOnSuccessListener(unused -> {
            Toast.makeText(getContext(), "Стаття успішно видалена",
Toast.LENGTH_LONG).show();

            FragmentManager manager = getParentFragmentManager();
            FragmentTransaction transaction =
manager.beginTransaction();
            transaction.replace(R.id.navHostFragment,
NewsFragment.class, null);
            transaction.commit();
        })
        .addOnFailureListener(e -> Toast.makeText(getContext(),
"Помилка видалення", Toast.LENGTH_LONG).show());
    });

return view;
}

```

```

private String getFileExtension(Uri uri) {
    MimeTypeMap mime = MimeTypeMap.getSingleton();
    return mime.getExtensionFromMimeType(cr.getType(uri));
}

private void uploadImage(Context context) {
    if (imageUri == null) {
        Toast.makeText(getContext(), "Помилка створення статті. Зображення
не завантажено.", Toast.LENGTH_LONG).show();
    }
    try {
        StorageReference fileReference =
storageReference.child(System.currentTimeMillis() + "." + getFileExtension(imageUri));
        fileReference.putFile(imageUri)
            .addOnSuccessListener(taskSnapshot -> {
                fileReference.getDownloadUrl().addOnSuccessListener(uri
-> {
                    if
(titleEditText.getText().toString().trim().equals("") ||
textEditText.getText().toString().trim().equals("")) {
                        Toast.makeText(context, "Помилка створення
статті. Неможливо створити статтю з порожнім заголовком чи текстом.",
Toast.LENGTH_LONG).show();
                    }
                    return;
                } else {
                    Article article = new
Article(titleEditText.getText().toString().trim(),
textEditText.getText().toString().trim(),
imageTextField.getText().toString().trim(),
uri.toString());
                    String imageId = db.push().getKey();
                    db.child(imageId).setValue(article).addOnSuccessListener(UNUSED ->
                        Toast.makeText(context, "Стаття
успішно створена", Toast.LENGTH_LONG).show())
                        .addOnFailureListener(e ->
                            Toast.makeText(context, "Помилка
створення статті", Toast.LENGTH_LONG).show());
                    FragmentManager manager =
getParentFragmentManager();

```

```

        FragmentTransaction transaction =
manager.beginTransaction();

        transaction.replace(R.id.navHostFragment,
NewsFragment.class, null);

        transaction.commit();
    }
});
}).addOnFailureListener(e ->
    Toast.makeText(context, "Помилка завантаження
зображення", Toast.LENGTH_LONG).show());
    } catch (Exception e) {
        Toast.makeText(context, "Помилка створення статті. Зображення не
завантажено.", Toast.LENGTH_LONG).show();
    }
}

@Override
public void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK
        && data != null && data.getData() != null) {
        imageUri = data.getData();
        imagePreview.setImageURI(imageUri);
    }
}

private void openFileChooser() {
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(intent, PICK_IMAGE_REQUEST);
}
}

```

fragment_create_news_article.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <LinearLayout

```

```

android:layout_width="match_parent"
android:layout_height="wrap_content"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:padding="20dp"
android:orientation="vertical">

```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:text="Створити статтю"
    android:textSize="20sp"
    android:layout_marginBottom="10dp"
    android:textColor="@color/colorPrimaryText"/>

```

```

<com.rengwuxian.materialedittext.MaterialEditText
    android:id="@+id/newsArticleTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/newsTitleRes"
    android:layout_marginBottom="20dp"
    android:textSize="18sp"
    android:inputType="text"
    android:maxLength="150"
    app:met_floatingLabel="highlight"
/>

```

```

<com.rengwuxian.materialedittext.MaterialEditText
    android:id="@+id/newsArticlePrimaryText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/newsTextRes"
    android:layout_marginBottom="20dp"
    android:textSize="18sp"
    android:gravity="top"
    android:inputType="textMultiLine"
    app:met_floatingLabel="highlight"
    android:minHeight="200dp"
/>

```

```

<RelativeLayout
    android:id="@+id/imageUploadLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```

```

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/uploadImageButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="@color/colorGray"
    android:text="@string/addImageTitle"
    android:textSize="16sp"/>

<com.rengwuxian.materialedittext.MaterialEditText
    android:id="@+id/imageUploadedFilename"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="25dp"
    android:layout_toEndOf="@id/uploadImageButton"
    android:text="@string/fileNameTitle"
    android:textSize="16sp"
    android:gravity="start|center"/>
</RelativeLayout>

<ImageView
    android:id="@+id/imagePreview"
    android:background="@color/colorGray"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:adjustViewBounds="true"
    android:maxHeight="500dp"/>

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/confirmImageUploadButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="@color/colorGray"
    android:text="@string/saveTitle"
    android:layout_gravity="center"
    android:textSize="16sp"
    android:paddingStart="40dp"
    android:paddingEnd="40dp"/>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:text="Видалити статтю"
    android:textSize="20sp"

```

```

        android:layout_marginTop="20dp"
        android:layout_marginBottom="10dp"
        android:textColor="@color/colorPrimaryText"/>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Оберіть статтю:"
    android:textSize="20sp"
    android:textColor="@color/colorPrimaryText"
    android:textColorHint="@color/colorPrimaryDark"/>

<Spinner
    android:id="@+id/articleSelectionSpinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:contentDescription="@string/classTeacherTitle"
    android:minHeight="48dp"/>

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/confirmDeletingNewsButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="@color/colorGray"
    android:text="Видалити"
    android:layout_gravity="center"
    android:textSize="16sp"
    android:paddingStart="40dp"
    android:paddingEnd="40dp"/>

</LinearLayout>
</ScrollView>

```

RegistrationFragment.java

```

package com.maxim328.zhuravonka.Controllers;

import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;

```

```

import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.maxim328.zhuravonka.Model.ChildrenClass;
import com.maxim328.zhuravonka.Model.User;
import com.maxim328.zhuravonka.R;
import com.rengwuxian.materialedittext.MaterialEditText;

import java.util.ArrayList;
import java.util.List;
import java.util.Objects;

public class RegistrationFragment extends Fragment {
    private FirebaseAuth auth;
    private DatabaseReference classesDB;
    private DatabaseReference users;

    private final List<String> roles;
    private final List<ChildrenClass> classes = new ArrayList<>();
    private final List<String> classesString = new ArrayList<>();

    private Spinner classSelectionSpinner;

    public RegistrationFragment() {
        roles = new ArrayList<>();
        roles.add("учень");
        roles.add("батько");
        roles.add("вчитель");
        roles.add("адміністратор");
    }

    public static RegistrationFragment newInstance() {

```

```

        RegistrationFragment fragment = new RegistrationFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        auth = FirebaseAuth.getInstance();
        if (auth.getCurrentUser() == null) auth.signInAnonymously();

        classesDB = FirebaseDatabase
            .getInstance("https://zhuravonka-12a93-default-rtdb.europe-
west1.firebaseio.com")
            .getReference("classes");

        classesDB.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                for (DataSnapshot dataSnapshot: snapshot.getChildren()) {
                    ChildrenClass currentClass =
dataSnapshot.getValue(ChildrenClass.class);
                    currentClass.setDbKey(dataSnapshot.getKey());
                    classes.add(currentClass);
                }

                for (int i = 0; i < classes.size(); i++) {
                    classesString.add(classes.get(i).getName());
                }

                ArrayAdapter<String> adapter = new ArrayAdapter<>(getContext(),
R.layout.spinner_item, classesString);
                adapter.setDropDownViewResource(R.layout.spinner_item);
                classSelectionSpinner = (Spinner)
getView().findViewById(R.id.classSelectionSpinner);
                classSelectionSpinner.setAdapter(adapter);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(getContext(), error.getMessage(),
Toast.LENGTH_LONG).show();
            }
        });
    }
}

```



```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_registration, container,
false);

    ArrayAdapter<String> adapter = new ArrayAdapter<>(getContext(),
R.layout.spinner_item, roles);
    adapter.setDropDownViewResource(R.layout.spinner_item);
    Spinner spinner = (Spinner)
view.findViewById(R.id.roleSelectionSpinner);
    spinner.setAdapter(adapter);

    LinearLayout selectClassLayout =
view.findViewById(R.id.classSelectionLayout);

    spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
            if (spinner.getSelectedItemPosition() == 0)
selectClassLayout.setVisibility(View.VISIBLE);
            else selectClassLayout.setVisibility(View.GONE);
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {
        }
    });

    Button registerBtn = view.findViewById(R.id.confirmRegistrationButton);
    auth = FirebaseAuth.getInstance();
    users = FirebaseDatabase
        .getInstance("https://zhuravonka-12a93-default-rtdb.europe-
west1.firebaseio.com")
        .getReference("users");

    registerBtn.setOnClickListener(v -> {
        MaterialEditText email = view.findViewById(R.id.loginField);
        MaterialEditText pass = view.findViewById(R.id.passField);
        MaterialEditText name = view.findViewById(R.id.nameField);
        MaterialEditText surname = view.findViewById(R.id.surnameField);
        MaterialEditText phone = view.findViewById(R.id.phoneField);
        MaterialEditText birth = view.findViewById(R.id.birthField);
    });
}

```

```

String role = "admin";
if (!(email.validate() && pass.validate() && name.validate() &&
    surname.validate() && phone.validate() && birth.validate()))
{
    Snackbar.make(view.findViewById(R.id.registrationFragment),
"Перевірте правильність заповнення полів", Snackbar.LENGTH_LONG).show();
    return;
}
if (email.getText().toString().trim().length() < 6 ||
email.getText().toString().trim().length() > 20 ||
    pass.getText().toString().trim().length() < 6 ||
pass.getText().toString().trim().length() > 12 ||
    name.getText().toString().trim().length() < 3 ||
name.getText().toString().trim().length() > 28 ||
    surname.getText().toString().trim().length() < 3 ||
surname.getText().toString().trim().length() > 12 ||
    phone.getText().toString().trim().length() != 12) {
    Toast.makeText(getContext(), "Невірна довжина одного із полів.",
Toast.LENGTH_LONG).show();
    return;
}

auth.createUserWithEmailAndPassword(Objects.requireNonNull(email.getText()).toString()
,
    Objects.requireNonNull(pass.getText()).toString()
    .addOnSuccessListener(authResult -> {
        User user = new User(email.getText().toString(),

Objects.requireNonNull(name.getText()).toString(),

Objects.requireNonNull(surname.getText()).toString(),

Objects.requireNonNull(phone.getText()).toString(),

Objects.requireNonNull(birth.getText()).toString(),
        spinner.getSelectedItemPosition() == 0 ?
classes.get(classSelectionSpinner.getSelectedItemPosition()).getDbKey() : null,

String.valueOf(spinner.getSelectedItemPosition()));

users.child(Objects.requireNonNull(FirebaseAuth.getInstance().getCurrentUser()).getUid
()).setValue(user)

        .addOnSuccessListener(unused ->

Snackbar.make(view.findViewById(R.id.registrationFragment),

```

```

        "Користувач створений",
        Snackbar.LENGTH_LONG).show()
        .addOnFailureListener(e ->

Snackbar.make(view.findViewById(R.id.registrationFragment),
        "Помилка створення користувача",
        Snackbar.LENGTH_LONG).show());
    }).addOnFailureListener(e -> {
        System.out.println("email error + ");
        e.printStackTrace();
    });
});

return view;
}
}

```

fragment_registration.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/registrationFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:orientation="vertical"
        android:gravity="center">

        <com.rengwuxian.materialEditText.MaterialEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/loginField"
            android:hint="@string/loginTitle"
            android:inputType="text"
            android:text=""
            android:textColor="@color/colorPrimaryText"
            android:textColorHint="@color/colorPrimaryDark"
            android:textSize="20sp"
            app:met_floatingLabel="highlight"
            app:met_minCharacters="6"

```

```
        app:met_maxCharacters="20"
    />

<com.rengwuxian.materialedittext.MaterialEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/passField"
    android:hint="@string/passTitle"
    android:inputType="textPassword"
    android:text=""
    android:textColor="@color/colorPrimaryText"
    android:textColorHint="@color/colorPrimaryDark"
    android:textSize="20sp"
    app:met_floatingLabel="highlight"
    app:met_minCharacters="6"
    app:met_maxCharacters="12"
/>

<com.rengwuxian.materialedittext.MaterialEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/nameField"
    android:hint="@string/nameTitle"
    android:inputType="text"
    android:text=""
    android:textColor="@color/colorPrimaryText"
    android:textColorHint="@color/colorPrimaryDark"
    android:textSize="20sp"
    app:met_floatingLabel="highlight"
    app:met_minCharacters="3"
    app:met_maxCharacters="28"
/>

<com.rengwuxian.materialedittext.MaterialEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/surnameField"
    android:hint="@string/surnameTitle"
    android:inputType="text"
    android:text=""
    android:textColor="@color/colorPrimaryText"
    android:textColorHint="@color/colorPrimaryDark"
    android:textSize="20sp"
    app:met_floatingLabel="highlight"
    app:met_minCharacters="3"
/>
```

```
        app:met_maxCharacters="12"
    />

<com.rengwuxian.materialedittext.MaterialEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/phoneField"
    android:hint="@string/phoneTitle"
    android:inputType="phone"
    android:text=""
    android:textColor="@color/colorPrimaryText"
    android:textColorHint="@color/colorPrimaryDark"
    android:textSize="20sp"
    app:met_floatingLabel="highlight"
    app:met_minCharacters="12"
    app:met_maxCharacters="12"
/>

<com.rengwuxian.materialedittext.MaterialEditText
    android:id="@+id/birthField"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/birthTitle"
    android:inputType="date"
    android:text=""
    android:textColor="@color/colorPrimaryText"
    android:textColorHint="#A1887F"
    android:textSize="20sp"
    app:met_floatingLabel="highlight" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/roleTitle"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryText"
        android:textColorHint="@color/colorPrimaryDark"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="15dp"
```

```
        android:layout_marginEnd="40dp"/>

    <Spinner
        android:id="@+id/roleSelectionSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="15dp"
        android:contentDescription="@string/roleTitle"
        android:minHeight="20dp"/>

</LinearLayout>

<LinearLayout
    android:id="@+id/classSelectionLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center"
    android:visibility="gone">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/classTitle"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryText"
        android:textColorHint="@color/colorPrimaryDark"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="20dp"
        android:layout_marginEnd="40dp"/>

    <Spinner
        android:id="@+id/classSelectionSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="20dp"
        android:contentDescription="@string/roleTitle"
        android:minHeight="20dp"/>

</LinearLayout>

<Button
    android:id="@+id/confirmRegistrationButton"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        android:background="#A9A9A9"
        android:backgroundTint="#BD3D3D"
        android:text="@string/confirmTitle"
        android:textColor="@color/colorPrimaryText" />

    </LinearLayout>

</ScrollView>

```

Authorization.java

```

package com.maxim328.zhuravonka.Controllers;

import android.os.Bundle;

import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.material.navigation.NavigationView;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.maxim328.zhuravonka.Model.StaticMethods;
import com.maxim328.zhuravonka.Model.User;
import com.maxim328.zhuravonka.R;
import com.rengwuxian.materialedittext.MaterialEditText;

public class Authorization extends Fragment {

    FirebaseAuth auth;
    Button authorizationBtn;

```

```

public Authorization() {
    // Required empty public constructor
}

public static Authorization newInstance(String param1, String param2) {
    Authorization fragment = new Authorization();
    return fragment;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_authorization, container,
false);

    authorizationBtn = view.findViewById(R.id.confirmAuthorizationButton);
    auth = FirebaseAuth.getInstance();

    authorizationBtn.setOnClickListener(v -> {
        MaterialEditText login = view.findViewById(R.id.loginAuthField);
        MaterialEditText pass = view.findViewById(R.id.passAuthField);
        auth.signInWithEmailAndPassword(login.getText().toString(),
pass.getText().toString())
            .addOnSuccessListener(authResult -> {
                Toast.makeText(getContext(), "Вхід успішно виконано",
Toast.LENGTH_LONG).show();

                DatabaseReference usersDB = FirebaseDatabase
                    .getInstance("https://zhuravonka-12a93-default-
rtdb.europe-west1.firebaseio.com")
                    .getReference("users");
                NavigationView navigationView =
getActivity().findViewById(R.id.navigationView);
                usersDB.child(auth.getCurrentUser().getUid()).get()
                    .addOnCompleteListener(task -> {
                        if (task.isSuccessful()) {
                            User currentUser =
task.getResult().getValue(User.class);

```



```

((TextView) getActivity().findViewById(R.id.userNameHeader)).setText(StaticMethods.getT
eacherName(currentUser));

((TextView) getActivity().findViewById(R.id.userEmailHeader)).setText(currentUser.getEm
ail());

                                if (currentUser.getRole().equals("2") ||
currentUser.getRole().equals("3")) {

navigationView.getMenu().findItem(R.id.menuCreateLesson).setVisible(true);

navigationView.getMenu().findItem(R.id.menuCreateClass).setVisible(true);

navigationView.getMenu().findItem(R.id.menuProfile).setVisible(true);

navigationView.getMenu().findItem(R.id.menuRegister).setVisible(true);

navigationView.getMenu().findItem(R.id.menuSchedule).setVisible(true);

navigationView.getMenu().findItem(R.id.menuSignOut).setVisible(true);
                                }
                                else {

navigationView.getMenu().findItem(R.id.menuProfile).setVisible(true);

navigationView.getMenu().findItem(R.id.menuSchedule).setVisible(true);

navigationView.getMenu().findItem(R.id.menuSignOut).setVisible(true);

navigationView.getMenu().findItem(R.id.menuRegister).setVisible(false);

navigationView.getMenu().findItem(R.id.menuCreateClass).setVisible(false);

navigationView.getMenu().findItem(R.id.menuCreateLesson).setVisible(false);
                                }

                                fragmentManager =
getParentFragmentManager();

                                FragmentTransaction transaction =
manager.beginTransaction();

transaction.replace(R.id.navHostFragment, NewsFragment.class, null);
                                transaction.commit();
                                }

});

```

```

navigationView.getMenu().findItem(R.id.menuSignOut).setVisible(true);
        }).addOnFailureListener(e -> Toast.makeText(getContext(),
"Помилка авторизації. Невірний email чи пароль.", Toast.LENGTH_LONG).show());

    });

    return view;
}
}

```

fragment_authorization.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/authorizationFragment"
    android:layout_width="match_parent"
    android:layout_marginStart="40dp"
    android:layout_marginEnd="40dp"
    android:layout_marginBottom="70dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center">

    <com.rengwuxian.materialedittext.MaterialEditText
        android:id="@+id/loginAuthField"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="96dp"
        android:hint="@string/loginTitle"
        android:inputType="text"
        android:selectAllOnFocus="true"
        app:met_floatingLabel="highlight"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <com.rengwuxian.materialedittext.MaterialEditText
        android:id="@+id/passAuthField"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"

```

```

android:hint="@string/passTitle"
android:imeActionLabel="@string/action_sign_in_short"
android:imeOptions="actionDone"
android:inputType="textPassword"
app:met_floatingLabel="highlight"
android:selectAllOnFocus="true"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/loginAuthField" />

```

```
<Button
```

```

    android:id="@+id/confirmAuthorizationButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="start"
    android:layout_marginStart="48dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="48dp"
    android:layout_marginBottom="64dp"
    android:enabled="true"
    android:text="@string/authorizationTitle"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/passAuthField"
    app:layout_constraintVertical_bias="0.2" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```