КОМП'ЮТЕРНІ НАУКИ

УДК 004.42
DOI https://doi.org/10.37734/2518-7171-2024-3-8

ПРОЄКТУВАННЯ ТА РОЗРОБКА ANDROID-ЗАСТОСУНКУ ДЛЯ РОБОТИ ІЗ РОЗКЛАДОМ УНІВЕРСИТЕТУ

Ю. Ф. ОЛЕКСІЙЧУК, кандидат фізико-математичних наук;
І. А. ГОРОШКО, магістр
(Полтавський університет економіки і торгівлі)

***Анотація. Мета дослідження**. Метою дослідження є розробка та програмна реалізація Android-застосунку для роботи з розкладом занять у Полтавському університеті економіки і торгівлі. Даний застосунок призначений для використання студентами, викладачами та іншими зацікавленими особами і має на меті зробити користування розкладом більш зручним, підвищити доступність розкладу, а також забезпечити низку додаткових переваг порівняно з існуючим web-застосунком. **Методика дослідження**. Методика дослідження включає аналіз існуючих рішень та тенденцій у використанні мобільних застосунків, зокрема в освітній сфері. Операційна система Android популярна в Україні та світі. Розробка Android-застосунку здійснюється з урахуванням функціоналу існуючого web-застосунку ПУЕТ. Особливості проєктування включають:*

1. Введення та збереження користувацьких даних, що дозволяє уникнути повторного введення та забезпечує зручність користування.

2. Локальне зберігання даних, що дає можливість перегляду розкладу без доступу до інтернету.

3. Налаштування сповіщень про зміни в розкладі, що підвищує оперативність отримання актуальної інформації.

Дослідження передбачає проєктування інтерфейсу, розробку архітектури застосунку, програмну реалізацію функціоналу та тестування розробленого застосунку з метою перевірки його зручності та надійності в реальних умовах експлуатації.

***Результати.** Мобільний застосунок реалізовано за допомогою мови програмування Java. Для вирішення різних задач використані бібліотеки GSON, Retrofit.*

***Висновки.** Розроблений мобільний застосунок забезпечує користувачам доступ до розкладу занять. Основною перевагою є можливість автономної роботи застосунку: дані зберігаються локально, а у разі наявного з'єднання з сервером відбувається їх оновлення. У майбутньому планується його модифікація та розширення шляхом додавання нового функціоналу.*

***Ключові слова:** мобільний застосунок, Android, Java, розклад занять.*

Постановка проблеми в загальному вигляді. В сучасному світі використання мобільних застосунків є звичною та зручною справою для різноманітних задач. Для отримання розкладу занять в Полтавському університеті економіки і торгівлі студентами та викладачами можна скористатися кількома способами, включаючи web-сайт. Розробка Android-застосунку зробить користування розкладом більш зручним та дозволить підвищити доступність розкладу.

Аналіз останніх досліджень і публікацій. На початок 2024 року в Україні було активно 55,64 млн мобільних пристрої, що становить 148,7% від загальної кількості населення [1]. Трохи більше 70% смартфонів в Україні працюють на операційній системі Android [2]. У світі Android є найпопулярнішою операційною системою, якщо враховувати всі апаратні системи [3].

Використання мобільних застосунків набуває все більшої популярності і в освіті [4], зокрема при роботі з розкладом в університетах [5-12]. Це може бути спеціальний застосунок, який орієнтований на роботу з розкладом [5-6,10,12], або одна із функцій в більш загальній системі управління [7-9].

В Полтавському університеті економіки і торгівлі (ПУЕТ) реалізований зручний web-застосунок, а якому можна подивитися розклад студентської групи чи груп, розклад викладача та завантаженість аудиторії [13]. Але розробка Android-застосунку є актуальною, оскільки дозволяє отримувати інформацію більш зручно та надійно.

Формування цілей статті. В статті розглядаються особливості проєктування та програмної реалізації Android-застосунку для роботи з розкладом ПУЕТ, який можуть використовувати студенти, викладачі університету та інші зацікавлені

особи. Даний застосунок повторює функціонал web-застосунку [13] та має ряд переваги. Основні із них:

1. Після вибору режиму користувач може ввести свої дані: викладач – прізвище та ім'я, студент – назву академічної групи або груп. Після цього дані запам'ятовуються і не потребують повторного введення. При необхідності ці дані можна змінити в налаштуваннях.

2. Дані зберігаються локально і розклад можна подивитися у випадку відсутності інтернету або проблем із сервером.

3. Сповіщення про зміни в розкладі.

Виклад основного матеріалу дослідження. Для програмної реалізації мобільного застосунку вибрана мова програмування Java [14], оскільки вона є однією із найпопулярніших мов програмування [15] і використовується для вирішення різноманітних задач.

Код мобільного застосунку складається з пакета, призначеного для роботи з даними, пакета, призначеного для забезпечення відображення інформації на екрані мобільного пристрою, та головного класу MainActivity.

Пакет призначений для роботи з даними має назву data та складається з інтерфейсів DataCallback, DataCallbackAllLessons та класів DataManager, DataRepository, DataUtils та наступних пакетів:

– database, який призначений для роботи з локальною базою даних Room. Він містить класи AppDatabase, ClassroomEntity, GroupEntity, LessonsConverter, LessonsEntity, TeacherEntity та чотири інтерфейси: @Dao interface ClassroomDao, interface GroupDao, interface LessonsDao, interface TeacherDao;

– model, який призначений для опису списків об'єктів, що одержуються з сервера, і містить чотири класи: Classroom, Group, Lesson, Teacher;

– network, який призначений для роботи з сервером. Цей пакет містить інтерфейс APIInterface з хедером та чотирма конструкціями @GET запитів до сервера для отримання даних, класи APIClient та ServerDataSource.

Пакет призначений для забезпечення відображення інформації на екрані має назву presentation та складається з пакетів:

– app_settings, який призначений для налаштування роботи додатку та містить клас class SettingsActivity;

– ui, який включає в себе класи користувацького інтерфейсу: AdapterClassroomName, AdapterGroupName, AdapterTeacherName, CalendarManager, LessonAdapter, LessonFragment, PresentationUtils, ViewPagerAdapter;

– widget, який забезпечує роботу віджету та містить три класи: ScheduleRemoteViewsFactory, ScheduleWidgetProvider, ScheduleWidgetService.

Сервер університету зберігає та надає по запиту списки викладачів, номери аудиторій та назви навчальних груп у форматі JSON [16]. Для отримання інформації з сервера використовується бібліотека Retrofit [17] – типобезпечний HTTP-клієнт для Android та Java. Ця бібліотека полегшує роботу з API у клієнт-серверних програмах, тому її використання дозволяє спростити взаємодію нашої програми із зовнішніми службами.

Для того, щоб застосунок мав можливість використовувати отримані від сервера дані, потрібно перетворити ці дані до зрозумілого для застосунку та зручного для обробки вигляду. Застосунок може отримувати три списки даних від серверу: це список викладачів, аудиторій та навчальних груп. Кожний з цих списків у програмі описано в окремих класах. Для серіалізації та десеріалізації JSON-даних використовується бібліотека GSON [18].

Для отримання даних від сервера використовується пакет network. У ньому є два інтерфейси. Інтерфейс DataCallback використовується для обробки результатів асинхронних запитів на завантаження даних.

```
public interface DataCallback<T> {
    void onDataLoaded(T data);
    void onError(Throwable throwable);
}
```

Він має два методи:

– onDataLoaded(T data), який викликається при успішному завантаженні даних;

– onError(Throwable throwable), який викликається при виникненні помилки під час завантаження даних.

Інтерфейс DataCallbackAllLessons схожий на DataCallback, але спеціально для завантаження списку всіх уроків.

```
public interface DataCallbackAllLessons {
        void onDataLoaded(List<List<Lesson>>
allLessons);
    void onError(Throwable throwable);
}
```

Клас DataManager відповідає за надання інтерфейсу для взаємодії з даними в додатку. Він виступає як посередник між зовнішнім середовищем наприклад UI-компонентами та внутрішніми джерелами даних - серверними та локальними базами даних. Клас забезпечує спрощений доступ до даних через декілька методів, які викликають відповідні методи класу DataRepository. Основні елементи:

– поле dataRepository відповідає за зберігання екземпляру DataRepository, який використовується для виконання всіх операцій з даними;

– поле context використовується для отримання доступу до ресурсів, файлової системи та інших системних сервісів Android;

– конструктор DataManager(Context context) ініціалізує DataManager та викликає метод initRepository() для ініціалізації DataRepository;

– метод initRepository() ініціалізує DataRepository з використанням ServerDataSource та DAO об'єктів (TeacherDao, ClassroomDao, GroupDao, LessonsDao), отриманих з AppDatabase;

– метод getTeachersList(DataCallback<List<Teacher>> callback) для обробки результатів завантаження списку викладачів викликає метод getTeachersListFromDataRepository з DataRepository для отримання списку викладачів;

– метод getRoomsList(DataCallback<List<Classroom>> callback) для обробки результатів завантаження списку аудиторій, викликає метод getClassroomsListFromDataRepository з DataRepository для отримання списку аудиторій;

– метод getGroupsList(DataCallback<List<Group>> callback) для обробки результатів завантаження списку груп, викликає метод getGroupsListFromDataRepository з DataRepository для отримання списку груп;

– метод getLessonsList(List<String> groupBands, DataCallbackAllLessons callback) з аргументами: groupBands – список групових ідентифікаторів, callback – колбек для обробки результатів завантаження списку уроків; викликає метод getLessonsListFromDataRepository з DataRepository для отримання списку уроків для заданих груп.

DataManager надає спрощений інтерфейс для взаємодії з даними. Наприклад, для отримання списку викладачів, можна викликати метод getTeachersList, передавши колбек для обробки результатів:

```
DataManager dataManager = new DataManager(context);
dataManager.getTeachersList(new DataCallback<List<Teacher>>() {
    @Override
    public void onDataLoaded(List<Teacher> data) {
        // Обробка успішно завантажених даних
    }

    @Override
    public void onError(Throwable throwable) {
        // Обробка помилки
    }
});
```

Подібний підхід використовується для отримання списків аудиторій, груп та уроків. Цей клас абстрагує складність взаємодії з сервером та локальною базою даних, роблячи його зручним для використання.

DataRepository є класом, що відповідає за управління даними між сервером, локальною базою даних та UI-компонентами. Клас обробляє запити до серверу, отримання даних з локальної бази даних та збереження даних в базу даних. Основні елементи класу – поля, конструктори та методи.

Поля:

– context: Context – контекст додатка для доступу до ресурсів;

– timestamp: String – використовується для зберігання часового штампа;

– serverDataSource: ServerDataSource – джерело даних з сервера;

– teacherDao: TeacherDao – DAO для роботи з таблицею викладачів;

– classroomDao: ClassroomDao – DAO для роботи з таблицею аудиторій;

– groupDao: GroupDao – DAO для роботи з таблицею груп;

– lessonsDao: LessonsDao – DAO для роботи з таблицею уроків;

– executor: Executor – виконавець для асинхронних задач;

– mainHandler: Handler – обробник для виконання завдань в основному потоці.

Конструктор DataRepository(Context context, ServerDataSource serverDataSource, TeacherDao teacherDao, ClassroomDao classroomDao, GroupDao groupDao, LessonsDao lessonsDao) ініціалізує DataRepository з відповідними DAO та серверним джерелом даних і має такі аргументи:

– context – контекст додатка;

– serverDataSource – джерело даних з сервера;

– teacherDao – DAO для роботи з таблицею викладачів;

– classroomDao – DAO для роботи з таблицею аудиторій;

– groupDao – DAO для роботи з таблицею груп;

– lessonsDao – DAO для роботи з таблицею уроків.

Методи:

– showToast(String message) з аргументом message – повідомлення для відображення, відображає тост-повідомлення з заданим текстом в основному потоці;

– getTeachersListFromDataRepository(final DataCallback<List<Teacher>> callback) з аргументом callback – колбек для обробки результатів завантаження списку викладачів, отримує список викладачів з сервера. Якщо сервер не надає даних, то завантажує дані з локальної бази даних;

– getClassroomsListFromDataRepository(final DataCallback<List<Classroom>> callback) з аргументом, callback – колбек для обробки результатів завантаження списку аудиторій, отримує список аудиторій з сервера. Якщо сервер не надає даних, то завантажує дані з локальної бази даних;

– getGroupsListFromDataRepository(final DataCallback<List<Group>> callback) з аргументом (callback – колбек для обробки результатів завантаження списку груп) отримує список груп

з сервера. Якщо сервер не надає даних, то завантажує дані з локальної бази даних;

– getLessonsListFromDataRepository(List<String> groupBands, DataCallbackAllLessons callback) з аргументами: groupBands – список ідентифікаторів груп, callback – колбек для обробки результатів завантаження списку уроків; отримує список уроків для заданих груп з сервера. Якщо сервер не надає даних, то завантажує дані з локальної бази даних;

– saveTeachersList(List<Teacher> teacherList) з аргументом teacherList – список викладачів для збереження, зберігає список викладачів у локальну базу даних, попередньо видаляючи існуючі записи;

– saveRoomsList(List<Classroom> classroomList) з аргументом classroomList – список аудиторій для збереження, зберігає список аудиторій у локальну базу даних, попередньо видаляючи існуючі записи;

– saveGroupsList(List<Group> groupList) з аргументом groupList – список груп для збереження, зберігає список груп у локальну базу даних, попередньо видаляючи існуючі записи;

– convertTeacherEntitiesToList(List<TeacherEntity> teacherEntities) з аргументом teacherEntities – список сутностей викладачів, конвертує список сутностей викладачів у список моделей викладачів;

– convertClassroomEntitiesToList(List<ClassroomEntity> classroomEntities) з аргументом classroomEntities – список сутностей аудиторій, конвертує список сутностей аудиторій у список моделей аудиторій;

– convertGroupEntitiesToList(List<GroupEntity> groupEntities) з аргументом groupEntities – список сутностей груп, конвертує список сутностей груп у список моделей груп;

– saveLessonsList(List<List<Lesson>> allLessonsList) з аргументом allLessonsList – список списків уроків для збереження, зберігає список уроків у локальну базу даних, попередньо видаляючи існуючі записи;

– convertLessonsEntitiesToList() конвертує список сутностей уроків у список списків уроків з локальної бази даних.

DataRepository є внутрішнім компонентом DataManager, який забезпечує доступ до даних, управління ними та їх зберігання. Наприклад, для отримання списку викладачів можна викликати метод getTeachersListFromDataRepository з відповідним колбеком для обробки результатів:

```
DataRepository dataRepository = new DataRepository(context, serverDataSource, teacherDao, classroomDao, groupDao, lessonsDao);
dataRepository.getTeachersListFromDataRepository(new DataCallback<List<Teacher>>() {
    @Override
    public void onDataLoaded(List<Teacher> data) {
        // Обробка успішно завантажених даних
    }

    @Override
    public void onError(Throwable throwable) {
        // Обробка помилки
    }
});
```

Подібний підхід використовується для отримання списків аудиторій, груп та уроків.

Після запуску програми виконується перевірка наявності інтернет-зв'язку, у разі наявності інтернету виконується запит до серверу і, якщо сервер надає очікувану відповідь, користувачу пропонується обрати параметри пошуку (викладач, аудиторія, навчальна група). Якщо відсутній інтернет або сервер не повертає очікуваної відповіді, користувачу буде запропоновано перевірити налаштування смартфону для забезпечення доступу до мережі інтернет або надати згоду на отримання інформації з локальної бази даних, куди було збережено раніше отриманий розклад. На головному екрані, окрім вибору параметрів пошуку для формування URL-запиту, також можна здійснити перехід на екран налаштування роботи програми, ці налаштування можуть бути збережені для подальшого застосування, після чого буде виконано повернення на головний екран програми.

У налаштуваннях програми доступні варіанти вибору світлого чи темного дизайну застосунку, можливість використання віджету на одному з екранів телефону у світлому або темному варіанті. Також пропонується можливість додавання розкладу до календарю Google і вибір періоду доби, на протязі якого програма не буде подавати звукових та вібраційних сигналів. Програма може відображати розклад на обрану кількість днів, тижнів або місяців. Важливим є вибір періодичності оновлення даних розкладу та налаштування сповіщення про можливі зміни у розкладі у порівнянні зі збереженою раніше інформацією. Користувач має також можливість налаштування отримання нагадувань про початок занять і варіант розкладу, який буде контролюватися та відображатися за замовчуванням.

Висновки із зазначених проблем і перспективи подальших досліджень у поданому напрямі. Розроблений мобільний застосунок дозволяє користувачам отримувати розклад занять у розрізі академічних груп, викладачів та аудиторій. Він може працювати автономно, у випадку наявного з'єднання з сервером відбувається оновлення даних. Додаток перебуває на етапі фінального тестування. В майбутньому він може бути модифікований та розширений за рахунок додавання нового функціоналу:

– можливість робити нотатки для певних занять;

– можливість додавати в розклад інші події;

– інтеграція з Google Calendar або подібними сервісами тощо.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Digital 2024: Ukraine – DataReportal – Global Digital Insights. URL: https://datareportal.com/reports/digital-2024-ukraine (дата звернення: 16.06.2024).

2. Mobile Operating System Market Share Ukraine | Statcounter Global Stats. URL: https://gs.statcounter.com/os-market-share/mobile/ukraine (дата звернення: 16.06.2024).

3. Operating System Market Share Worldwide | Statcounter Global Stats. URL: https://gs.statcounter.com/os-market-share (дата звернення: 16.06.2024).

4. Горошко І., Горошко В. Горошко А. Мобільні додатки в освіті: розробка інтелектуального android-застосунку для поліпшення інформаційної доступності розкладу занять. *Measuring and computing devices in technological processes*. 1 (Бер 2024), 13–20. DOI: https://doi.org/10.31891/2219-9365-2024-77-2.

5. Muhammad S. H., Galadanci B. S., Mustapha A., Yahaya A. S. Design and implementation of an android and web-based university timetable customization system. *Bayero Journal of Pure and Applied Sciences*. 2017. 10(1). 320–325. DOI: https://doi.org/10.4314/bajopas.v10i1.50

6. Yi L. Y., Mahrom N., Calvin L. Android-based timetable manager for University students using rule-based algorithm. *In AIP Conference Proceedings.* 2023. Vol. 2579, No. 1. *AIP Publishing.* DOI: https://doi.org/10.1063/5.0114159

7. Ejiyi C. J., Deng J., Ejiyi T. U., Salako A. A., Ejiyi M. B., Anomihe C. G. Design and Development of Android Application for Educational Institutes. *In Journal of Physics: Conference Series.* 2021. Vol. 1769, №. 1, p. 012066. IOP Publishing.

8. Hossain I., Ullah S. A., Haque A. K. M. M. Managing the Activities of a University Department through Android Application. International J. Eng. Inf. Syst., 2023. 7(1), 57–65.

9. Дорошенко Т. А. Android-додаток київського університету імені Бориса Грінченка. *Інформаційні технології 2015: зб. тез II Української конференції молодих науковців*, 28–29 трав. 2015 р., м. Київ, 30-32.

10. Марченко М. М., Римар П. В. Розробка мобільного додатку «РОЗКЛАД ЗАНЯТЬ» під платформу Android. *Прикладні інформаційні технології*. 2020. 113–116.

11. Махун Д., Демида Б. Дослідження технології створення прикладних аплікацій для ОС ANDROID. *Вісник Національного університету Львівська політехніка. Комп'ютерні науки та інформаційні технології*. 2013. (771). 111–120.

12. Міченко О. О. Розробка Android додатку автоматизованої системи формування розкладу. *Сучасні інформаційні технології та системи в управлінні.* 2018. *Збірник матеріалів І Міжнародної науково-практичної конференції молодих вчених, аспірантів і студентів*.

13. Ольховська О. В., Кошова О. П., Ольховський Д. М., Семикоз Д. С. Розробка web-застосунку для формування розкладу в закладі вищої освіти. *Вісник Херсонського національного технічного університету*. 2023. 1 (84). 155–162. https://doi.org/10.35546/kntu2078-4481.2023.1.21

14. Smyth N. Android Studio 4.2 Development Essentials-Java Edition: Developing Android Apps Using Android Studio 4.2, Java and Android Jetpack. Ebookfrenzy. 2021.

15. Programming-language popularity worldwide 2023 | Statista. URL: https://www.statista.com/statistics/869092/worldwide-software-developer-survey-languages-used/ (дата звернення: 16.06.2024).

16. Bray T. The javascript object notation (json) data interchange format (No. rfc7159). 2014.

17. Lachgar M., Benouda H., Elfirdoussi S. Android rest APIS: volley vs retrofit. *In 2018 international symposium on advanced electrical and communication technologies (ISAECT).* 2018, November. pp. 1-6. IEEE.

18. Patel S. K. Instant Gson. Packt Publishing Ltd. 2013.

REFERENCES

1. Digital 2024: Ukraine – DataReportal – Global Digital Insights. Retrieved from: https://datareportal.com/reports/digital-2024-ukraine (дата звернення: 16.06.2024).

2. Mobile Operating System Market Share Ukraine | Statcounter Global Stats. Retrieved from: https://gs.statcounter.com/os-market-share/mobile/ukraine (дата звернення: 16.06.2024).

3. Operating System Market Share Worldwide | Statcounter Global Stats. Retrieved from: https://gs.statcounter.com/os-market-share (дата звернення: 16.06.2024).

4. Horoshko, I., Horoshko, V., & Horoshko, A. (2024). Mobilni dodatky v osviti: rozrobka intelektualnoho android-zastosunku dlia polipshennia informatsiinoi dostupnosti rozkladu zaniat [Development of an intelligent android application to improve the information accessibility of the lesson schedule]. *Measuring and computing devices in technological processes.* 1 (May 2024), 13–20. DOI: https://doi.org/10.31891/2219-9365-2024-77-2 [in Ukrainian].

5. Muhammad, S. H., Galadanci, B. S., Mustapha, A., & Yahaya, A. S. (2017). Design and implementation of an android and web-based university timetable customization system. *Bayero Journal of Pure and Applied Sciences*, 10(1), 320–325. DOI: https://doi.org/10.4314/bajopas.v10i1.50

6. Yi, L. Y., Mahrom, N., & Calvin, L. (2023, October). Android-based timetable manager for University students using rule-based algorithm. *In AIP Conference Proceedings (Vol. 2579, No. 1). AIP Publishing.* DOI: https://doi.org/10.1063/5.0114159

7. Ejiyi, C. J., Deng, J., Ejiyi, T. U., Salako, A. A., Ejiyi, M. B., & Anomihe, C. G. (2021). Design and Development of Android Application for Educational Institutes. *In Journal of Physics: Conference Series* (Vol. 1769, No. 1, p. 012066). IOP Publishing.

8. Hossain, I., Ullah, S. A., & Haque, A. K. (2023). Managing the Activities of a University Department through Android Application. *International J. Eng. Inf. Syst.,* 7(1), 57–65.

9. Doroshenko, T. A. (2015). Android-dodatok kyivskoho universytetu imeni Borysa Hrinchenka [Android application of Borys Grinchenko Kyiv University]. *Informatsiini tekhnolohii 2015: zb. tez II Ukrainskoi konferentsii molodykh naukovtsiv [Information technologies 2015: coll. theses of the 2nd Ukrainian Conference of Young Scientists],* May 28–29. 2015, Kyiv, 30–32 [in Ukrainian].

10. Marchenko, M. M., & Rymar, P. V. (2020). Rozrobka mobilnoho dodatku «ROZKLAD ZANIaT» pid platformu Android [Development of the mobile application "CLASS SCHEDULE" for the Android platform]. *Prykladni informatsiini tekhnolohii [Applied information technologies]*, 113–116 [in Ukrainian].

11. Makhun, D., & Demyda, B. (2013). Doslidzhennia tekhnolohii stvorennia prykladnykh aplikatsii dlia OS ANDROID [Research on the technology of creating applied applications for the ANDROID OS]. *Visnyk Natsionalnoho universytetu Lvivska politekhnika. Kompiuterni nauky ta informatsiini tekhnolohii –Bulletin of the Lviv Polytechnic National University. Computer Science and Information Technology,* (771), 111–120 [in Ukrainian].

12. Michenko, O. O. (2018). Rozrobka Android dodatku avtomatyzovanoi systemy formuvannia rozkladu [Development of an Android application of an automated system for creating a schedule]. *Suchasni informatsiini tekhnolohii ta systemy v upravlinni. Zbirnyk materialiv I Mizhnarodnoi naukovo-praktychnoi konferentsii molodykh vchenykh, aspirantiv i studentiv – Modern information technologies and management systems. Collection of materials of the 1st International scientific and practical conference of young scientists, graduate students and students].* [in Ukrainian].

13. Olkhovska, O. V., Koshova, O. P., Olkhovskyi, D. M., & Semykoz, D. S. (2023). Rozrobka web-zastosunku dlia formuvannia rozkladu v zakladi vyshchoi osvity. [Development of a web application for creating a schedule in a higher education institution]. *Visnyk Khersonskoho natsionalnoho tekhnichnoho universytetu – Bulletin of the Kherson National Technical University*, 1 (84), 155–162. https://doi.org/10.35546/kntu2078-4481.2023.1.21 [in Ukrainian].

14. Smyth, N. (2021). Android Studio 4.2 Development Essentials-Java Edition: Developing Android Apps Using Android Studio 4.2, Java and Android Jetpack. Ebookfrenzy.

15. Programming-language popularity worldwide 2023 | Statista. Retrieved from: https://www.statista.com/statistics/869092/worldwide-software-developer-survey-languages-used/ (дата звернення: 16.06.2024).

16. Bray, T. (2014). The javascript object notation (json) data interchange format (No. rfc7159).

17. Lachgar, M., Benouda, H., & Elfirdoussi, S. (2018, November). Android rest APIS: volley vs retrofit. In 2018 international symposium on advanced electrical and communication technologies (ISAECT) (pp. 1-6). IEEE.

18. Patel, S. K. (2013). Instant Gson. Packt Publishing Ltd.

Yu. Oleksiichuk**, PhD; **I. Horoshko**, Master (Poltava University of Economics and Trade).* ***Design and development of an android application for working with the university schedule

***Abstract. Purpose of the Study**. The purpose of the study is the development and software implementation of an Android application for managing the class schedule at Poltava University of Economics and Trade. This application is intended for use by students, faculty, and other interested parties. It aims to make schedule management more convenient, enhance schedule accessibility, and provide several additional advantages compared to the existing web application. **Methodology.** The research methodology includes the analysis of existing solutions and trends in the use of mobile applications, particularly in the educational sector. The Android operating system is popular in Ukraine and globally. The development of the Android application takes into account the functionality of the existing web application. The design features include:*

1. Input and storage of user data which avoids repeated input and ensures ease of use.

2. Local data storage, which allows viewing the schedule without internet access.

3. Notifications for schedule changes, which enhance the timeliness of receiving up-to-date information.

The study involves interface design, application architecture development, software implementation of functionality, and testing of the developed application to verify its convenience and reliability in real-world conditions.

***Results.** The mobile application was implemented using the Java programming language. Libraries such as GSON and Retrofit were used to address various tasks.*

***Conclusions.** The developed mobile application provides users with access to the class schedule. The primary advantage is its ability to operate autonomously: data is stored locally, and updates occur when connected to the server. Plans include modification and expansion by adding new functionalities.*

***Key words:** mobile application, Android, Java, class schedule.*