



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної підготовки

випускна кваліфікаційна робота бакалавра

«ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ОБМІНУ МЕДІА КОНТЕНТОМ»

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проекту

№	П.І.Б.	Група
1	Бондаренко М. О.	КН-П-192
2	Андрухів В. Р.	КН-А-191
3	Панкратов. М. А	КН-Д-192
4	Тарасюк А. С.	КН-П-191
5	Кохановский І. О.	КН-П-191
6	Сапригін А. Д.	КН-П-191

Автор звіту

_____ Сапригін Андрій Дмитрович

АНОТАЦІЯ

УДК 004.9

С-191

Сапригін А. Д. Соціальна мережа для обміну фотографіями та відео: Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса: ОТУШ, 2023 - 13 с.

Робота присвячена дослідженням, розвиненню та впровадженню інформаційної системи, головним призначенням якої є взаємодія між користувачами. Додаткові функції системи включають реєстрацію, авторизацію та автентифікацію, можливість створити новий пароль.

Актуальність роботи визначається великою кількістю людей, які хочуть вести активний спосіб життя в інтернет-просторі без витрат на покупку преміум привілеїв. Програма націлена на людей, які хочуть: ділитися новими враженнями, фото, відео, стежити за активністю інших користувачів, коментувати чийсь думку, знайомитися з новими людьми, брати участь у різних спільнотах. Особливу увагу привертає простота використання програми без зайвого. Все що потрібно користувачеві, це: аутентифікувати свою особистість, змінити свій пароль якщо забув, створити пост, видалити пост, завантажити на пристрій медіа контент, знайти користувачів (групи), лайкати, зберігати, коментувати та пересилати пости.

Об'єктом дослідження у рамках даної роботи виступає засіб комунікації, який використовує додаток із багатьма потрібними йому можливостями. У якості задач дослідження було встановлене наступне:

- провести огляд інформаційних систем, що враховують відомості про те, які з інструментів найчастіше використовують в соціальних мережах, встановити характерні риси та напрями удосконалення.
- визначити програмні засоби (інтерфейси API), які дозволяють досягти максимальної швидкості, комфорту та безпечності користувачу.
- скласти програмне забезпечення, реалізувавши у ньому результати, досягнуті у попередніх задачах, провести випробування за різних умов та з різних пристроїв.
- впровадити у систему програмні засоби захисту даних для покращення показників інформаційної безпеки створеної системи.

Методи дослідження, що використані для досягнення поставлених задач, утворені суперпозицією методів системного аналізу, моделювання та когнітивної методології.

При складанні звіту використано 42 посилання на друковані та електронні джерела інформації.

ЗМІСТ

ВСТУП _____	3
Технічне завдання до проєкту _____	4
РОЗДІЛ 1. Загальна характеристика системи _____	6
РОЗДІЛ 2. Реалізація клієнтської частини _____	7
РОЗДІЛ 3. Результати випробувань _____	10
ВИСНОВКИ _____	14
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ _____	13

ВСТУП

Розвиток і поширення інформаційних технологій супроводжується перенесенням великої кількості можливостей у цифровий простір. Стратегія розвитку інформаційного суспільства в Україні визначає, що «комунікаційні технології є необхідним інструментом соціально-економічного прогресу, одним з основних чинників інноваційного розвитку економіки» [1].

На сьогоднішній день кожна друга людина використовує свій телефон не тільки для мобільного зв'язку, але і для розваг, а саме соціальних мереж. Завдяки соціальним мережам людина може дізнатися багато про іншу людину навіть не телефонуючи їй і тим більше не виходячи з дому. Також, соціальна мережа може бути в ролі рекламного майданчика, за яку не обов'язково потрібно платити, як за послугу. Завдяки рекламним можливостям людина може просунути своє ремесло, бізнес і так далі, про яке може дізнатися більша кількість людей.

Однак, якщо давати людям можливість ділитися своєю особистою інформацією це потребує покращення інформаційної безпеки захисту персональних даних користувача. Конфіденційність може зумовлюватися перевіркою особистості, а саме використанням можливості реєстрації, авторизації та аутентифікації як мінімум. Це актуалізує додаткові дослідження і розробки в зазначеній області.

Як предмет даної роботи була обрана досить поширена система обміну графічним контентом у функціональність якої входить розміщення постів різного роду, які приходять від одних користувачів та обробляються іншими, головною метою яких є донести до читачів ідею самого посту завдяки опису та графічному контенту.

Також функції переважної більшості подібних систем повинні передбачати можливість додавати історії - короткі відео формати, додавати місце створення публікації, обмеження, що накладаються на групу користувачів.

Детальне технічне завдання до проекту наводиться далі. У рамках даного звіту розглядається клієнтська частина, що являє собою веб версію додатку ClickShot та мобільний додаток, створений для користувачів пристроїв з операційною системою «Android». Звіт складається з трьох розділів у яких описується загальна характеристика системи, деталі створення клієнтської частини та результати її розгортання та тестування. Висновки відображають результати, що досягнуті у рамках даної частини проекту, загальний висновок є сукупністю звітів усіх учасників проекту.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЕКТУ

Загальний опис.

Програмне забезпечення (ПЗ) необхідно для надання можливості ділитися інформацією з іншими людьми та реагувати на контент, який був доданий.

Призначення:

Програмне забезпечення орієнтовно на людей, які ведуть активне соціальне життя в інтернет просторі.

Вимоги до функціоналу:

Реєстрація користувачів:

Користувач ПЗ має змогу зареєструватись як за допомогою авторизаційних даних: емейлу, логіну, паролю, дати народження.

При реєстрації обов'язково зазначається емейл адреса користувача. Далі вказується: логін (унікальний ідентифікатор користувача), пароль, день народження завдяки якому усі підписники будуть повідомлені. Останній етап реєстрації - створення аватарки. Користувач може пропустити цей крок. На сторінці реєстрації користувач може обрати: чи буде відображено його емейл, чи будуть приходити листи новин від додатку на пошту клієнта. Також клієнт матиме можливість на етапі створення емейл адреси вказати, чи буде запам'ятовувати його авторизаційні дані, додаток у майбутньому.

Основне робоче місце користувача ПЗ клієнтської частини повинне забезпечити усі необхідні засоби для:

- Авторизації, зокрема реєстрації та аутентифікації
- Зміни паролю у разі, якщо користувач забув
- Розміщення постів
- Перегляду посів
- Лайкнути, додати в збереженні, відправити клієнту, залишити коментар
- Перегляду коментарів на пости
- Відповідей на коментарі інших користувачів
- Завантаження медіа-контент поста
- Копіювання посилання на пост
- Знаходження користувачів додатку
- Фільтрування постів
- Переходу на сторінку користувача
- Підписки, відписки від користувача
- Перегляду інформації користувача
- Зміни особистої інформації (нік, email, дата народження, тощо.)

Основне робоче місце користувача ПЗ адміністраторської частини повинне забезпечити усі необхідні засоби для:

- Авторизації, зокрема реєстрації та аутентифікації
- Зміни паролю у разі, якщо користувач забув
- Перегляду всіх клієнтів та їх редагування
- Перегляду всіх постів та їх редагування
- Перегляду всіх коментарів щодо посту та їх редагування

Архітектура:

ПЗ має бути спроектоване за клієнт-серверною архітектурою з розподіленими вузлами: Backend - серверне програмне забезпечення, що забезпечує збереження та оброблення поточних даних, взаємодію з базою даних. Надає програмний інтерфейс взаємодії (API) з іншими вузлами.

Web App - додаток, доступний з веб-простору (сайт), що забезпечує візуальний інтерфейс користувачів усіх категорій.

Mobile App - додаток, що встановлюється на мобільних пристроях (смартфон, планшет).

Усі модулі ПЗ повинні мати засоби

- розгортання на новому пристрої (інсталяції)

Технології

Зберігання вихідних кодів ПЗ повинно здійснюватись за допомогою системи контролю версій (вибір конкретної системи узгоджується групою проекту).

Структура файлів та каталогів повинна відбивати архітектуру ПЗ, на зразок:

- main
 - Backend
 - Web
 - Mobile

Для кожного з модулів рекомендується (з можливістю часткової зміни) наступні технології створення:

Backend - Java, MySQL

Web - HTML, CSS, Javascript, Typescript, JS фреймворки, зокрема React JS, Next JS

Mobile - Java або Kotlin

Вихідний код повинен оформлятися у відповідності до загальноприйнятих стандартів, на зразок стандартів GNU <http://www.gnu.org/prep/standards/>, у тому числі з дотриманням вимог до коментування та документування коду.

РОЗДІЛ 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

У відповідності до технічного завдання (ТЗ) система була спроектована за клієнт-серверною розподіленою архітектурою. У складі системи було умовно відокремлено наступні архітектурні частини:

- «Mockup»
- «Backend»
- «Frontend»

Частина «Mockup» являє собою набір моделей та описів поведінки користувача. Модель подає інтерфейси взаємодії користувачів різного типу з інформаційною системою, траєкторії рухів та переходів між різними інтерфейсами. При створенні інтерфейсів застосовано принципи розумної навігації, включно із підходами UI/UX, висновки колористики та ергономіки мобільних пристроїв. Частина розміщена за адресою <https://www.figma.com/file/iaMsWLpWsp6rpYfFM8WW4Z/ClickShot-graduate-work?type=design&node-id=0-1&mode=design>

Частина «Backend» реалізована як централізований вузол збереження та оброблення даних, що створюються та модифікуються протягом життєвого циклу інформаційної системи. Дана частина включає у себе базу даних проекту та програмний інтерфейс (API) доступу до основних її функцій, зокрема, керування контентом, авторизації, узгодження профілів користувачів різного типу та/або різних пристроїв.

Частина «Backend» виконана на базі програмного забезпечення Java, СУБД «MySQL».

Клієнтська частина системи «Front End», призначена для надання користувачеві доступу до функцій системи через засоби візуального інтерфейсу. Для обміну даними між «Backend», «Frontend», використовується бібліотека «Retrofit».

Подальший звіт присвячений детальному опису цієї частини.

З метою покращення безпеки, звертаючи увагу на той факт, що мобільні пристрої досить часто підключаються до незахищених мереж передавання даних, інформаційний обмін між частинами «Backend», «Frontend» захищений стандартом шифрування PBCS, SHA1. Це убезпечує роботу системи у відкритих мережах, а також спрощує задачі автентифікації та доступу до контенту з обмеженим доступом без наявності ключів розшифрування даних, навіть за умови дискредитації інших модулів системи.

РОЗДІЛ 2

РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ЧАСТИНИ

Клієнтська частина інформаційної системи реалізована з використанням наступних технологій:

- Операційна система: Android, гарантується підтримка систем з версіями API від 23 до 33 (поточна на час початку створення проекту).
- Засоби розроблення: мова програмування Java (OpenJDK 64-Bit Server VM by Oracle Corporation) та середовище Android Studio

У складі клієнтської частини (мобільного додатку для операційної системи Android та веб версії) передбачено та реалізовано наступні функції:

- Авторизація користувача, реєстрація нових користувачів
- Перегляд та додавання нових постів
- Коментування постів
- Знаходження постів
- Керування особистими та поточними даними (кабінетом)

Початок роботи користувача із додатком супроводжується його реєстрацією як учасника інформаційної системи. При реєстрації користувач обов'язково надає відомості про електронну пошту, через яку буде здійснюватись оповіщення користувача щодо актуальних питань, а також про зміну активності у його особистому просторі. Попередньо до реєстрації введені дані проходять попередню валідацію за допомогою засобів регулярних виразів

```
Pattern userName = Pattern.compile("(?=[A-Za-z\\d])[A-Za-z][A-Za-z\\d.-]{0,19}$"),  
email = Pattern.compile("[a-zA-Z\\d_\\-\\.]+@[a-zA-Z\\d_\\-\\.]+\\.([a-zA-Z]{2,5})$"),  
password = Pattern.compile("(?=[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[@$!%*?&])[A-Za-z\\d@$!%*?&]{8,32}$");
```

Авторизація користувача здійснюється за допомогою персонального секрету (логіну та паролю), введенного на етапі реєстрації.

При реєстрації чи авторизації здійснюється комунікація з частиною «Backend» у результаті чого авторизація підтверджується або спростовується. Дані, що передаються між зазначеними частинами, шифруються за допомогою протоколу https, а саме методами шифрування TLS/SSL. Обмін даними з серверною частиною реалізовано за допомогою окремих програмних засобів. У якості прикладу далі наведено приклад передавання даних методом POST та GET, що широко вживається у різних частинах проекту

Java (Android)

```
// POST - multipart  
@retrofit2.http.Multipart  
@POST("/addPost")
```

```

Call<String> STRING_CALL(
    @Part("media") RequestBody file,
    @Nullable @Part("otherInfo") String description
);

// складання Part - media (@Body)
requestBody media = RequestBody.create(MediaType.parse(getString(R.string.mime_image) + "/"
+ extension), mediaBytes);
requestBody media = RequestBody.create(MediaType.parse(getString(R.string.mime_video) + "/"
+ extension), mediaBytes);
requestBody media = RequestBody.create(MediaType.parse(getString(R.string.mime_audio) + "/"
+ extension), mediaBytes);

// складанні Part - otherInfo (@Body)
JSONObject jsonObject = new JSONObject();
jsonObject.put("author", login);
jsonObject.put("description", description);
jsonObject.put("metadata", metadata);
jsonObject.put("postponePublication", postponePublication != null ?
DateFormatting.formatToDateTime(postponePublication) :
DateFormatting.formatToDateTime(new Date()));
jsonObject.put("taggedPeople", taggedPeople);
return jsonObject;

// GET - regular request
@GET("/confirm")
Call<String> STRING_CALL(
    @Query("login") String login
);

// приклад створення інстансу об'єкту Retrofit
public static void addUser(Callback<String> callback, String body) throws IOException,
JSONException {
// create main interface
SendToAddUser mainInterface = retrofit.create(SendToAddUser.class);

// initialize call
Call<String> call = mainInterface.STRING_CALL(body);
call.enqueue(callback);
}

```

3 важливих функцій додатку є локалізація та зберігання стану авторизованого користувача.

Для забезпечення функціональності локалізації вжито наступні дії:

1. Створення файлів які відповідають за надання ресурсів, а саме українських та англійських речень.
2. Створення класу, який має колекцію локалей, та обрану локаль

```
public class Localisation {
```

```

public Map<String, Locale> getLocales() {
    return locales;
}

private final Map<String, Locale> locales;
public static Locale chosenLocale;
private final ArrayAdapter<String> adapter;

public Localisation(Activity activity) {
    String[] languagesArray = {activity.getString(R.string.en),
        activity.getString(R.string.uk)};

    locales = new HashMap<>();
    locales.put("English", new Locale("en", "EN"));
    locales.put("Українська", new Locale("uk", "UK"));

    adapter = new ArrayAdapter<>(activity, android.R.layout.simple_spinner_item,
languagesArray);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
}

public static void setFirstLocale(Spinner languages) {
    if (Localisation.chosenLocale != null) {
        if (Localisation.chosenLocale.toString().equals("en_EN")) {
            languages.setSelection(0);
        } else if (Localisation.chosenLocale.toString().equals("uk_UK")) {
            languages.setSelection(1);
        }
    }
}

public static Configuration setLocalize(AdapterView<?> parent, Localisation localisation, int
position) {
    String language = (String) parent.getItemAtPosition(position);

    Locale locale = Objects.requireNonNull(localisation.getLocales().get(language));
    Locale.setDefault(locale);

    Localisation.chosenLocale = locale;

    Configuration configuration = new Configuration();
    configuration.setLocale(locale);

    return configuration;
}

public ArrayAdapter<String> getAdapter() {
    return adapter;
}
}

```

Для зберігання функціональності зберігання стану користувача (кешу) вжито наступні дії:

1. Створення класу роботи з локальним об'єктом кешу

```

public class Cache {
    public static void saveSP(Context context, String key, Object obj) {
        android.content.SharedPreferences preferences =
context.getSharedPreferences(CacheScopes.MAIN_SCOPE.toString(), Context.MODE_PRIVATE);

        android.content.SharedPreferences.Editor editor = preferences.edit();

        if (obj.getClass().equals(Boolean.class)) {
            editor.putBoolean(key, (Boolean) obj);
        } else if (obj.getClass().equals(Integer.class)) {
            editor.putInt(key, (Integer) obj);
        } else if (obj.getClass().equals(String.class)) {
            editor.putString(key, (String) obj);
        }

        editor.apply();
    }

    public static void deleteSP(Context context, String key) {
        android.content.SharedPreferences sharedPreferences =
context.getSharedPreferences(CacheScopes.MAIN_SCOPE.toString(), Context.MODE_PRIVATE);
        android.content.SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.remove(key).apply();
    }

    public static boolean loadBoolSP(Context context, String key) {
        android.content.SharedPreferences sharedPreferences =
context.getSharedPreferences(CacheScopes.MAIN_SCOPE.toString(), Context.MODE_PRIVATE);
        return sharedPreferences.getBoolean(key, false);
    }

    public static int loadIntSP(Context context, String key) {
        android.content.SharedPreferences sharedPreferences =
context.getSharedPreferences(CacheScopes.MAIN_SCOPE.toString(), Context.MODE_PRIVATE);
        return sharedPreferences.getInt(key, 0);
    }

    public static String loadStringSP(Context context, String key) {
        android.content.SharedPreferences sharedPreferences =
context.getSharedPreferences(CacheScopes.MAIN_SCOPE.toString(), Context.MODE_PRIVATE);
        return sharedPreferences.getString(key, "");
    }
}

```

2. Створення класу видалення всього кешу додатку коли користувач дає згоду

```

public class DeleteApplicationCache {
    public static void deleteCache(Context context) {
        try {
            File dir = context.getCacheDir();
            boolean temp = deleteDir(dir);
            Log.d("deleteCache: ", String.valueOf(temp));
        } catch (Exception e) {
            Log.e("deleteCache: ", e.getMessage());
        }
    }
}

```

```

public static boolean deleteDir(File dir) {
    if (dir != null && dir.isDirectory()) {
        String[] children = dir.list();
        for (int i = 0; i < Objects.requireNonNull(children).length; i++) {
            boolean success = deleteDir(new File(dir, children[i]));
            if (!success) {
                return false;
            }
        }
        return dir.delete();
    } else if (dir != null && dir.isFile()) {
        return dir.delete();
    } else {
        return false;    }}}

```

Важливою частиною додатку є визначення розширення файлів та їх MIME типів. Для реалізації такої можливості було створено клас, який визначає розширення файлу виходячи з контексту файла та локального посилання на ресурс.

```

public class FindExtension {
    @SuppressWarnings("Range")
    public static String getExtension(Uri selectedImageUri, Context context) {
        String fileName = null;
        if (selectedImageUri.getScheme().equals("content")) {
            Cursor cursor = context.getContentResolver().query(selectedImageUri, null, null, null, null);
            try {
                if (cursor != null && cursor.moveToFirst()) {
                    fileName =
cursor.getString(cursor.getColumnIndex(OpenableColumns.DISPLAY_NAME));
                }
            } finally {
                assert cursor != null;
                cursor.close();
            }
        }
        if (fileName == null){
            fileName = selectedImageUri.getPath();
            int mark = fileName.lastIndexOf("/");
            if (mark != -1){
                fileName = fileName.substring(mark + 1);
            }
        }
        return fileName.substring(fileName.lastIndexOf(".") + 1);
    }
}

```

З повним кодом проекту можна ознайомитись у репозиторії <https://github.com/Saprsuu6/Diplom>

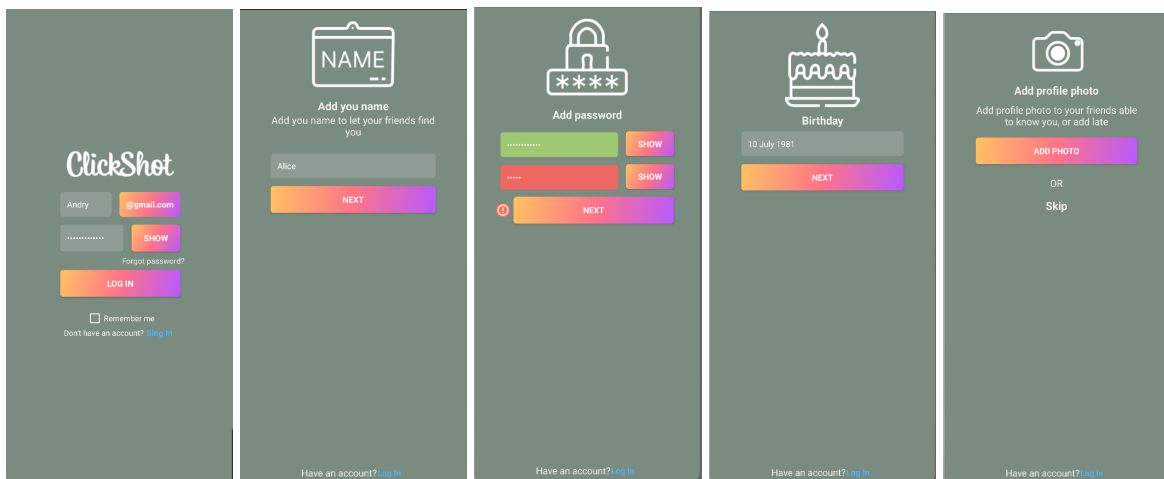
РОЗДІЛ 3

РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Результатом компіляції програмного коду клієнтської частини являється інсталяційний (APK) файл додатку, що може бути встановлений на смартфон або планшет з операційною системою «Android». Головна серія випробувань, звіт з якої наведено далі, проведена на смартфоні «Xiaomi Redmi Note 10» з API 33.

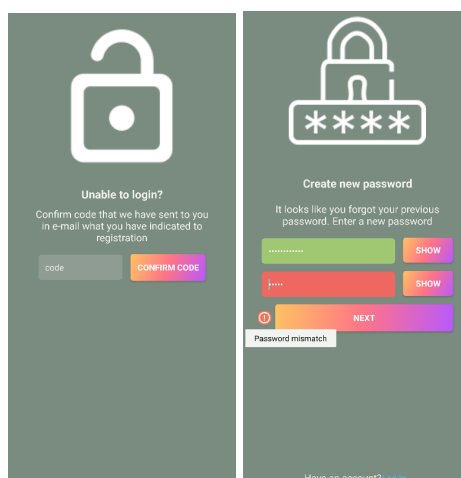
Початок роботи нового користувача з додатком починається з реєстрації користувача у системі.

Якщо користувач раніше проходив реєстрацію, то він може увійти до системи за допомогою персональних даних, зазначених при реєстрації. Процес авторизації показано на мал. 1, 2, 3, 4, 5, 6



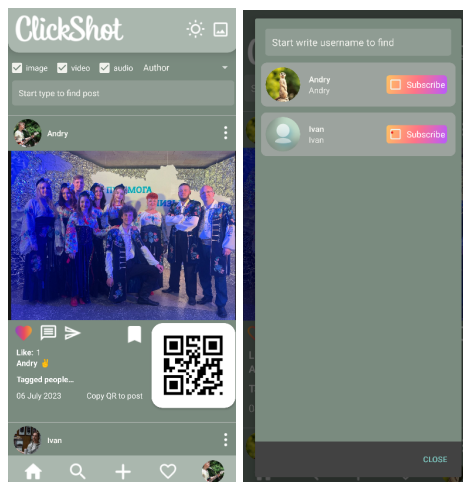
Мал. 1, 2, 3, 4, 5, 6 - процес авторизації

У разі, якщо користувач забув свій пароль він має можливість його відновити. Можливість відновлення паролю відображено на мал. 9, 10



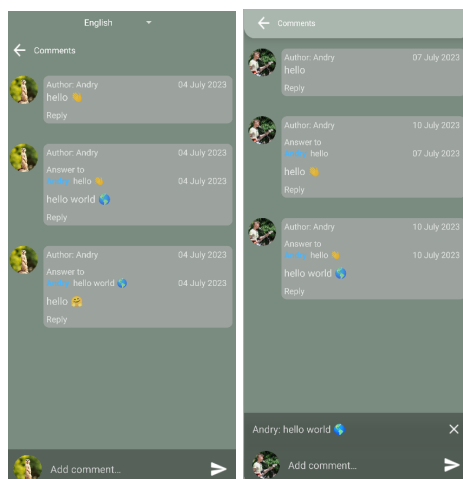
Мал. 9, 10 - можливість відновлення паролю

Після входу до системи користувач отримує доступ до основних функцій системи. Однією з таких функцій є перегляд контенту. Можливість перегляду новостної стрічки та пошук користувачів відображено на мал. 11, 12



Мал. 11, 12 - можливість перегляду новостної стрічки та пошук користувачів

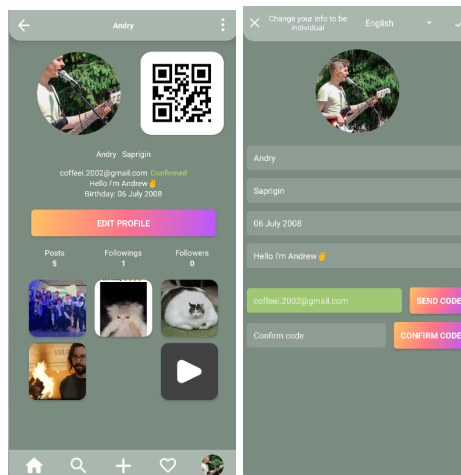
Користувач додатку має можливість переглядати коментарі на пост та відповідати на них. Можливість відправки, відповіді, та перегляду коментарів відображено на мал. 13



Мал. 13 - можливість відправки, відповіді, та перегляду коментарів

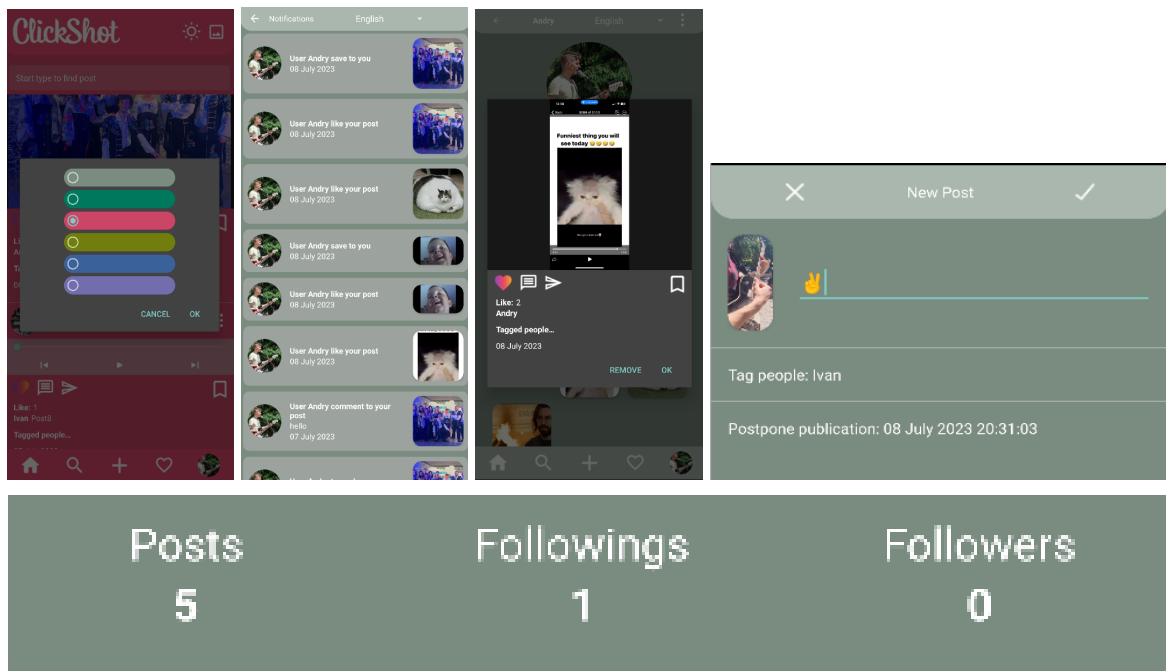
Через кабінет користувач має змогу переглядати персональні дані, переглядати статус підтвердження email, коригувати свої персональні дані.

Перегляд інформації про користувача та зміна особливих даних відображено на мал. 14, 15



Мал. 14, 15 - перегляд інформації про користувача та зміна особливих даних

Також, клієнт має можливість обирати тему додатку, переглядати збережені медіафайли, дивитися повідомлення активності інших користувачів щодо свого контенту, переглядати конкретний пост, додавати нові пости, дивитися підписників та тих на кого поточний користувач підписан. Дивіться мал. 16, 17, 18, 19, 20



У підсумку, всі функції клієнтської частини, що були регламентовані технічним завданням було реалізовано та випробувано. Поточна версія додатку, що реалізує клієнтську частину, доступна у репозиторії <https://github.com/Saprsuu6/Diplom>

ВИСНОВКИ

Проведено огляд інформаційних систем, що враховують відомості про локацію користувачів, виявлено, що суттєве поширення мають системи, які пов'язані із наданням транспортних послуг. Саме для таких систем відомості про локацію мають основне значення. У якості прикладу вибрано для реалізації систему замовлення пасажирських перевезень (таксі).

Визначено програмні засоби (інтерфейси API), які дозволяють отримувати та відправляти дані про стан користувача на даний момент та відправку та отримання медіаконтенту у трьох різних форматах: фото, відео та аудіо, перевагу надано засобом від “Square” Retrofit Library. Також, для відображення стану завантаження даних з сервера, було використано ShimerLayout від “Facebook”. Для завантаження медіа контенту з інтернету було використано механізм Glide. Дані засоби було вжито при створенні системи.

Спроековано, реалізовано та випробувано систему узгодження взаємодії користувачів з урахуванням геопозиціонування її учасників. Система складена за розподіленою клієнт-серверною архітектурою та складається з чотирьох частин, розміщених за адресами:

- «Mockup»
<https://www.figma.com/file/iaMsWLpWsp6rpYfFM8WW4Z/ClickShot-graduate-work?type=design&node-id=0-1&t=k5jgRmsJ01lbFYTV-0>
- «Backend» <https://github.com/drDrake6/ClickshotBackend>
- «Frontend» (WebVersion) <https://github.com/MiroslavaBo/clickshot>
- «Frontend» (Mobile Version) <https://github.com/Saprsuu6/Diplom>

У систему запроваджено можливість реагувати на медіа-контент за допомогою жестів. Також вжито засобів підтвердження персональних даних користувача через електронну пошту. Це дозволяє реалізувати у майбутньому засоби двофакторної автентифікації, а також ускладнює зловмисникам вживання підроблених (невласних) даних для дискредитації системи.

Систему було випробувано шляхом інсталяції на фізичні пристрої та проведення тестових сеансів комунікації. Результати випробування підтвердили загальну працездатність створеної системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стратегія розвитку інформаційного суспільства в Україні. Розпорядження Кабінету Міністрів України від 15 травня 2013 р. № 386-р. // Урядовий портал. URL: <https://www.kmu.gov.ua/npas/246420577>
2. Стратегія кібербезпеки України. Указ Президента України від 14 травня 2021 року // Урядовий кур'єр від 28.08.2021 — № 165. URL: <https://zakon.rada.gov.ua/laws/show/447/2021/print>
3. Решетило В.П., Федотова Ю.В. Невизначеність та ризик: співвідношення понять та специфіка прийняття рішень // Проблеми системного підходу в економіці, випуск № 3(77)-2, 2020. с. 149. URL: http://psae-jrnl.nau.in.ua/journal/3_77_2_2020_ukr/23.pdf
4. Олександр Смичніков. Параноя або сучасні реалії: як зберегти конфіденційність у мережі. - ФБЮЧЕР МЕДІА. URL: <https://mind.ua/openmind/20183329-paranoya-abo-suchasni-realiyi-yak-zberegiti-konfidencijnist-u-merezhi> (дата звернення 30.05.2022)
5. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. – Київ. – 2017. – 110 с.
6. Величко О.М. Інтелектуальні інформаційні системи: структура і застосування. Підручник / Величко О.М., Гордієнко Т.Б. - Херсон. - 2022. - 728 с. ISBN: 978-966-289-552-0
7. Булгакова О.С. Інформатика: візуальне програмування. Навчальний посібник (стереотипне видання) / Булгакова О.С., Зосімов В.В., Бронницька Н.А., Танкова Н.В. - Херсон. - 2020. - 312 с. ISBN: 978-966-289-039-6
8. Булгакова О.С. Методи та системи штучного інтелекту: теорія та практика. Навчальний посібник / Булгакова О.С., Зосімов В.В., Поздеев В.О. - Херсон. - 2020. - 356 с. ISBN: 978-966-289-364-9
9. Чемакіна О.В. Дизайн системи візуальної інформації. Навчальний посібник / Чемакіна О.В., Рубцов А.Л., Свірко В.О. - Херсон. - 2019. - 200 с. ISBN: 978-966-289-216-1
10. Шеховцов А.В. Комп'ютерні технології для дизайнерів. Навчальний посібник (стереотипне видання) / Шеховцов А.В., Полетаєва Г.Н., Крючковський Д.О., Бараненко Р.В. - Херсон. - 2019. - 318 с. ISBN: 978-966-2393-08-8
11. Чайковська М.П. Концептуально-методологічні засади управління маркетинговими ІТ-проектами в умовах цифрових трансформацій. Монографія / Чайковська М.П. - Херсон. - 2021. - 370 с. ISBN: 978-966-289-537-7
12. Реалізація адаптивного інтерфейсу користувача в Android // URL: <https://www.fandroid.info/realizatsiya-adaptivnogo-polzovatelskogo-interfejsa-v-android/>
13. Retrofit документація// URL: <https://square.github.io/retrofit/>
14. Glide документація // URL: <https://bumptech.github.io/glide/>
15. Використання QR посилання у мобільних додатках
 - 15.1. <https://github.com/journeyapps/zxing-android-embedded>
 - 15.2. <http://surl.li/iywou>
16. Android elements документація // URL: <https://developer.android.com/>
 - 16.1. <https://developer.android.com/reference/androidx/swiperefreshlayout/widget/SwipeRefreshLayout>
 - 16.2. <https://android-developers.googleblog.com/2017/02/build-flexible-layouts-with.html>
 - 16.3. <https://facebook.github.io/shimmer-android/>