



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

випускна кваліфікаційна робота бакалавра

«ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ОБМІНУ МЕДІАКОНТЕНТОМ»

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

_____ Кохановський Іван Олегович

АНОТАЦІЯ

УДК 004.9

К-191

Кохановський І. О. Соціальна мережа для обміну медіа-контентом: 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. Також вона буде корисна тим хто вимушено переміщений і підтримує контакти переважно через такі системи. 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 медіаконтент, знайти користувачів (групи), лайкати, зберігати, коментувати та пересилати пости.

Об'єктом дослідження у рамках даної роботи виступає засіб комунікації. 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.

ЗМІСТ

ВСТУП _____	3
Технічне завдання до проекту _____	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 підтримує контакти з іншими за допомогою мобільного зв'язку, але у ситуації де мобільний оператор, послугами якого користувалася людина, не може надавати свої послуги, треба мати альтернативні засоби підтримання зв'язку. Завдяки соціальним мережам людина може дізнатися багато про іншу людину навіть не телефонуючи їй або знаходячись у країні де потрібний мобільний оператор не працює. Також, соціальна мережа може бути в ролі рекламного майданчика, завдяки чому за неї не обов'язково потрібно платити. 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
- Дати оцінку, 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.)

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 додержанням 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».

Вихідні коди та ресурси додатку розміщені на репозиторії <https://github.com/drDrake6/ClickshotBackend.git>. 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 СЕРВЕРНОЇ ЧАСТИНИ

Серверна частина інформаційної системи реалізована з використанням наступних технологій:

- Сервер: Tomcat
- Засоби розроблення: мова програмування Java (OpenJDK SE Runtime Environment) та середовище IntelliJ Idea

У складі серверної частини передбачено та реалізовано наступні функції:

- Аутентифікація 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.

У якості прикладу далі наведено приклад передавання даних методом POST та GET, що широко вживається у різних частинах проекту

```
Scanner s = new Scanner(req.getInputStream(), "UTF-8").useDelimiter("\\A");
String bodyText = s.hasNext() ? s.next() : null;
if(bodyText == null) {
    System.out.println("AddUserServlet::doPost | body was null");
    //res.getWriter().write("body was null");
    return null;
}
JSONObject bodyObj = new JSONObject(bodyText);
return bodyObj;
```

Нижче представлений фрагмент коду реєстрації

```
JSONObject bodyUser = bodyParseService.parseBody(req);

try {
    bodyUser.put("birthday", bodyUser.getString("birthday") + " 00:00:00");
}
catch (Exception ex){
```

```

    res.getWriter().write("2: birthday can not be empty");
    return;
}

User user = null;

try {
    user = new User(bodyUser);
}
catch (Exception ex){
    res.getWriter().write(ex.getMessage());
    return;
}

try {
    if(userDAO.getUser(user.getLogin()) != null){
        res.getWriter().write("1: user with such login exists");
        return;
    }
} catch (JSONException ex) {
    res.getWriter().write(ex.getMessage());
    return;
}

user.setAvatar("user_icon_without_photo.PNG");

try {
    user.setToken(UUID.randomUUID().toString());
    user.setEmail_code("not_confirmed");
    userDAO.add(user);
}
catch (Exception ex){
    res.getWriter().write(ex.getMessage());
    return;
}

res.getWriter().write("0: " + user.getToken());

```

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

Аутентифікація підтверджується якщо реквізити для входу не правильні та якщо користувача з таким логіном не існує інакше аутентифікація спростовується. У разі успішної аутентифікація користувачеві відправляється токен авторизації, який зберігається у базі та за допомогою якого користувач може виконувати дії над власними даними такі як видалення своїх постів або зміна своїх коментарів.

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 Servlet API. У якості прикладу далі наведено приклад отримання даних аутентифікації методом POST та обробка їх

```
JSONObject credentialsObj = bodyParseService.parseBody(req);
String login = "";
String password = "";
try {
    login = credentialsObj.getString("login");
    password = credentialsObj.getString("password");
    System.out.println(login);
} catch (JSONException ex) {
    res.getWriter().write("3: " + ex.getClass().getName() + "\n" + ex.getMessage());
    return;
}

User user = userDAO.getUser(login);
if(user != null && userDAO.CheckCredentials(user, password)){
    if(user.getEmail_code() != null &&
    !user.getEmail_code().equals("not_confirmed"))
        user.setEmail_code(null);
    user.setEmail_attempt(0);
    user.setToken(UUID.randomUUID().toString());
    userDAO.update(user, user.getId());
    res.getWriter().write("0: " + user.getToken());
    System.out.println("access allowed");
}
else{
    res.getWriter().write( "1: access denied");
    System.out.println("access denied");
}
```

Пароль у базі зберігається у вигляді хешу (SHA1) у який додається сіль (за стандартом PBCs)

1. Код створення хешу SHA1

```
public String hash(String data){
    try {
        MessageDigest md = MessageDigest.getInstance( "SHA" );
        byte[] hash = md.digest( data.getBytes() );
        StringBuilder sb = new StringBuilder();
        for( byte b : hash ) {
            sb.append( String.format("%02x", b & 0xFF ) );
        }
    }
}
```

```

        return sb.toString() ;
    }
    catch( NoSuchAlgorithmException ex ) {
        System.out.println( ex.getMessage() ) ;
        return null ;
    }
}

```

2. Генерування сілі та створення хешу паролю

```

private String genSalt(){
    return hashService.hash(UUID.randomUUID().toString());
}

private String makePasswordHash(String password, String salt){
    return hashService.hash(salt + password + salt);
}

```

Важливою частиною додатку є додавання та видалення постів, які передаються у вигляді двох потоків даних перший з яких є медіа-даними а другий це решта інформації про пост

Додавання посту

```

try {
    if ("POST".equals(req.getMethod()) &&
        req.getContentType().contains("multipart/form-data")) {
        req.setAttribute(Request.__MULTIPART_CONFIG_ELEMENT, new
            MultipartConfigElement(""));
    }
    Part imagePart = req.getPart("media");

    List<MimeTypeService.MediaType> mediaTypes = new ArrayList<>();
    mediaTypes.add(MimeTypeService.MediaType.IMAGE);
    mediaTypes.add(MimeTypeService.MediaType.VIDEO);
    mediaTypes.add(MimeTypeService.MediaType.AUDIO);

    String path = uploadService.Upload(imagePart,
        req.getServletContext().getRealPath("/"),
        mediaTypes);

    Part postPart = req.getPart("otherInfo");
    byte[] postBytes;
    InputStream is = postPart.getInputStream();
    postBytes = IOUtils.readAllBytes(is);
}

```

```

String postSource = new String(postBytes, StandardCharsets.UTF_8);
JSONObject jpost = new JSONObject(postSource);

jpost.put("id", UUID.randomUUID().toString());
jpost.put("addDate", LocalDateTime.now().
    format(DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss")));
jpost.put("postponePublication", jpost.get("postponePublication"));
jpost.put("mediaUrl", path);
Post post = new Post(jpost);
postDAO.add(post);

if(!jpost.isNull("taggedPeople")){
    String tmp = jpost.getString("taggedPeople");
    String[] logins = tmp.split(" ");
    taggedPeopleDAO.setTaggedPeople(jpost.getString("id"), logins);
}
} catch (Exception ex){
    res.getWriter().write("1: " + ex.getMessage());
    return;
}
res.getWriter().write("successfully added");

```

Видалення посту

```

res.setHeader("Access-Control-Allow-Origin", "");
JSONObject body = bodyParseService.parseBody(req);
String token = null;
String postId = null;
try {
    token = body.getString("token");
    postId = body.getString("postId");
}
catch (JSONException ex){
    res.getWriter().write("1: internal error, " + ex.getMessage());
    return;
}

User user = userDAO.getUserByToken(token);

if(user == null){
    res.getWriter().write("2: access denied");
    return;
}

Boolean has = postDAO.authorHasPost(postId, user.getLogin());

if(has == null){

```

```

        res.getWriter().write("1: internal error");
        return;
    }
    else if (has || user.getRole() == 'a'){
        if(postDAO.deletePostById(postId))
            res.getWriter().write("0: successfully deleted");
        else
            res.getWriter().write("1: internal error");
        return;
    }

    res.getWriter().write("2: access denied");

```

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

```

@Singleton
public class MimeService {

    public enum MediaType {IMAGE, VIDEO, AUDIO};
    private Map<String, String> imageTypes;
    private Map<String, String> videoTypes;
    private Map<String, String> audioTypes;

    private Map<MediaType, Map<String, String>> checker;
    public MimeService() {
        imageTypes = new HashMap<>();
        videoTypes = new HashMap<>();
        audioTypes = new HashMap<>();
        imageTypes.put("bmp", "image/bmp" );
        imageTypes.put("gif", "image/gif" );
        imageTypes.put("jpg", "image/jpeg");
        imageTypes.put("jpeg", "image/jpeg");
        imageTypes.put("png", "image/png" );
        imageTypes.put("webp", "image/webp");
        videoTypes.put("mp4", "video/mp4");
        audioTypes.put("mp3", "audio/mpeg");
        audioTypes.put("mpeg", "audio/mpeg");
        checker = new HashMap<>();
        checker.put(MediaType.IMAGE, imageTypes);
        checker.put(MediaType.VIDEO, videoTypes);
        checker.put(MediaType.AUDIO, audioTypes);
    }

    public boolean checkMimeType(MediaType mediaType, String extension){

```

```
        return checker.get(mediaType).containsKey(extension.toLowerCase());
    }

    public boolean checkMimeTypes(List<MediaType> mediaTypes, String
extension){
        for (MimeType.MediaType mediaType : mediaTypes) {
            if(checkMimeType(mediaType, extension)){
                return true;
            }
        }
        return false;
    }

    public String getMimeType(String extension){
        if(imageTypes.containsKey(extension)){
            return imageTypes.get(extension);
        }
        return null;
    }
}
```

З повним кодом проекту можна ознайомитись у репозиторії <https://github.com/drDrake6/ClickshotBackend>

РОЗДІЛ 3

РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Результатом компіляції програмного коду клієнтської частини являється (WAR) файл додатку, що може бути задеплоєний на сервері. Головна серія випробувань, звіт з якої наведено далі, проведена на сервері «Tomcat 8.5.79».

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. Зовнішній вигляд екранів реєстрації та входу наведено на рис. 1.

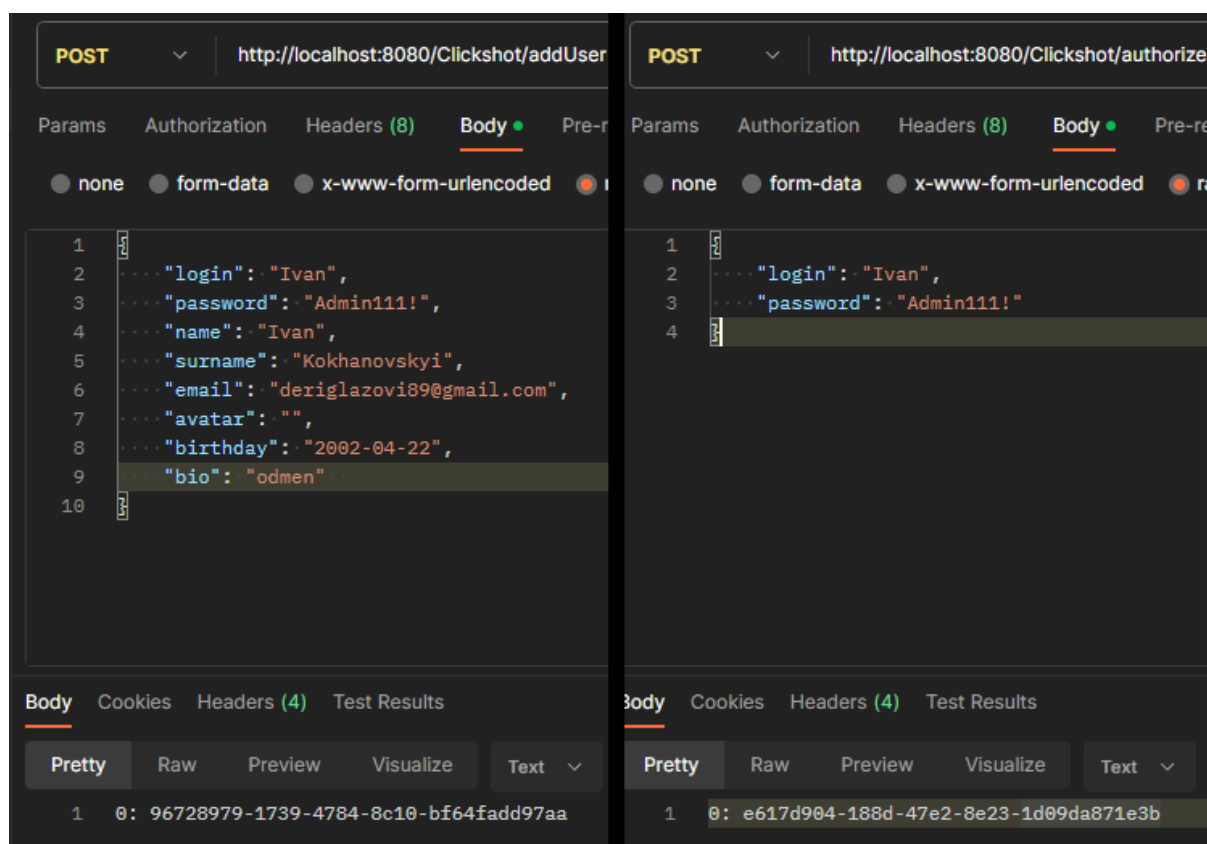


Рисунок 1 - Реєстрація користувача та вхід

Після входу до системи користувач отримує доступ до основних функцій системи. Одними з таких функцій є додавання та видалення поста (см. рис. 2, рис. 3)

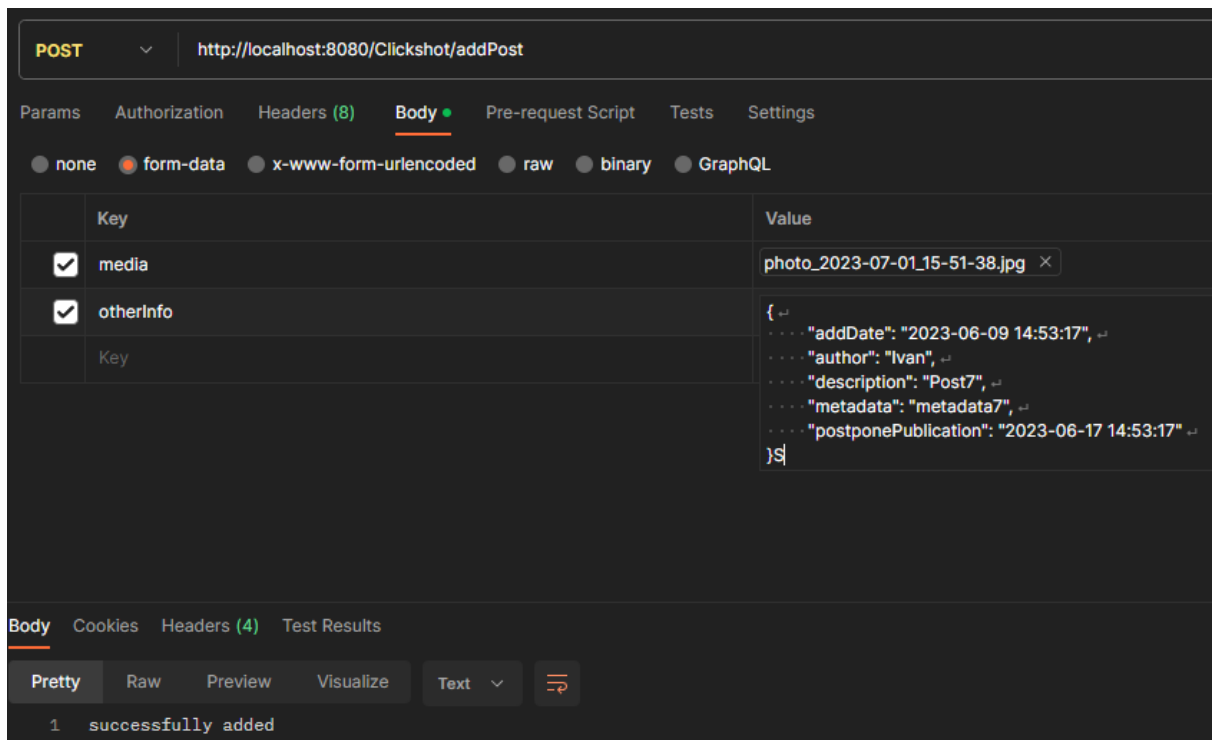


Рисунок 2 - Додавання поста

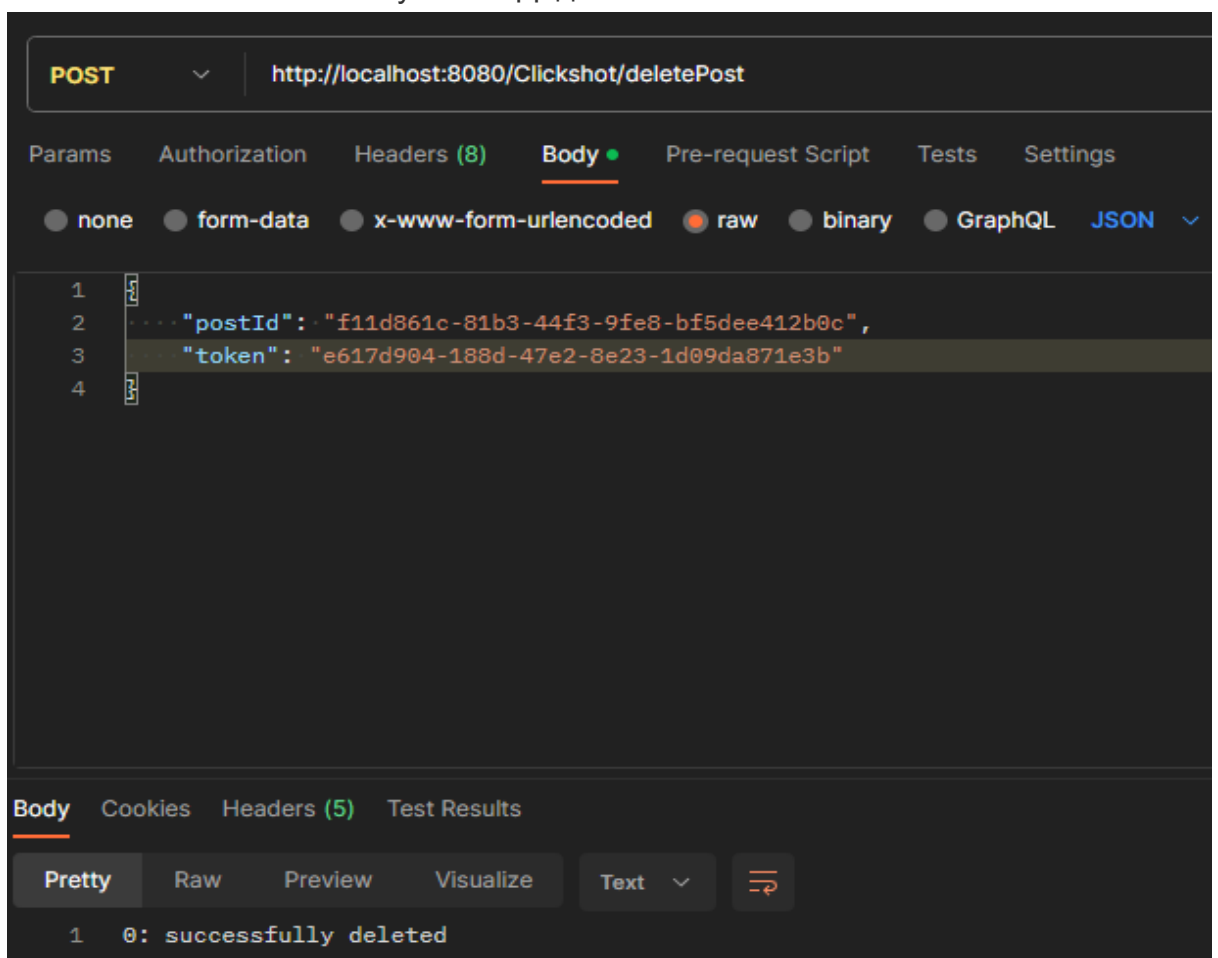


Рисунок 3 - Видалення поста

Після додавання посту інформаційна система фіксує її, зберігають дані у базі даних. Далі користувач має змогу переглядати стрічку постів як наведено на рис. 4.

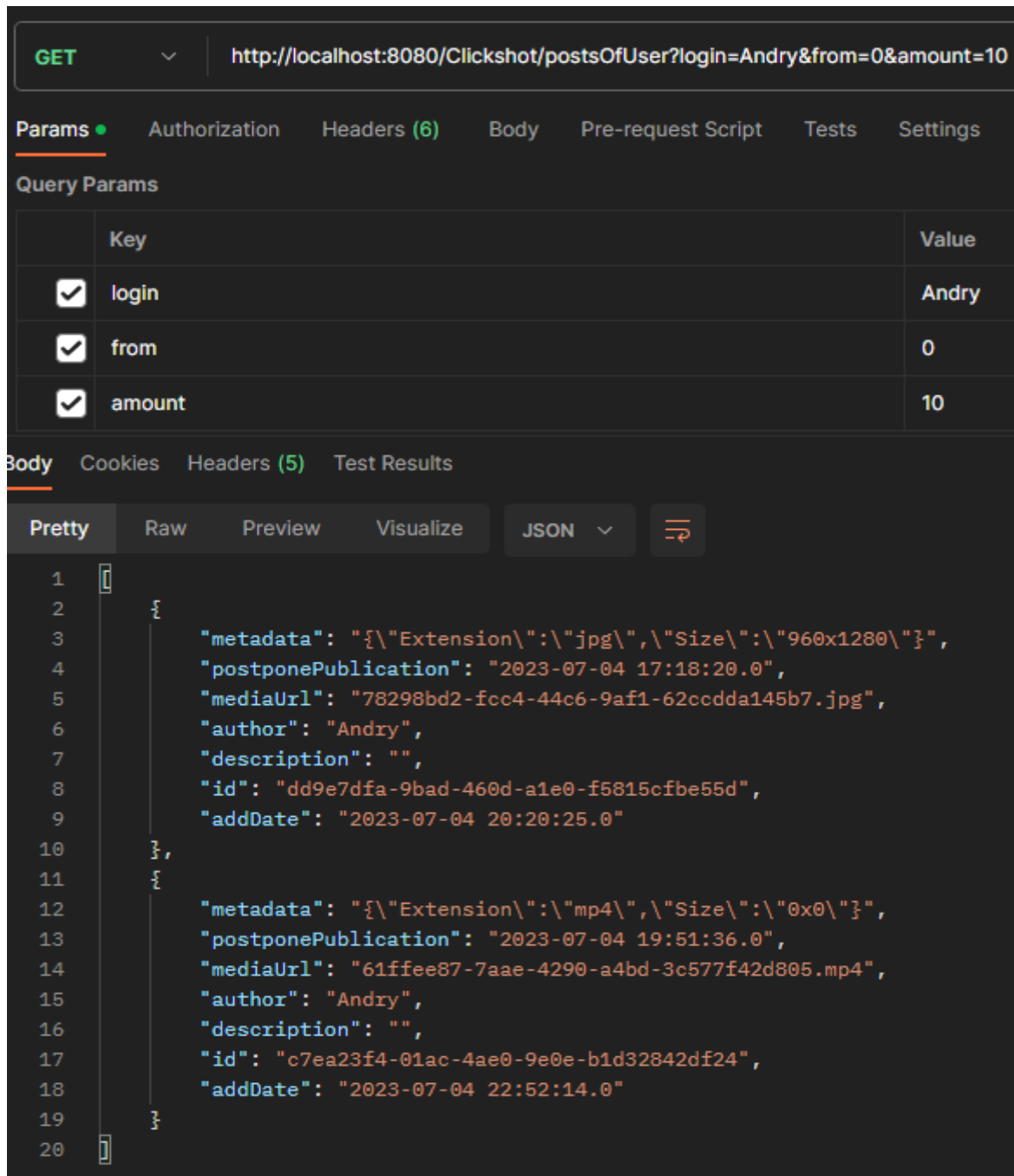


Рисунок 4 - отримання перших 10-ти постів у стрічці

У підсумку, всі функції клієнтської частини, що були регламентовані технічним завданням було реалізовано та випробувано. Поточна версія додатку, що реалізує серверну частину, доступна у репозиторії <https://github.com/drDrake6/ClickshotBackend>

ВИСНОВКИ

Проведено огляд інформаційних систем, що призначені до обміну медіаконтентом, виявлено, що суттєве поширення мають системи, які пов'язані з обміном графічним контентом. Але систем, які націлені на обмін будь якими медіафайлами (не тільки відео але й аудио). Проект був націлений на розробку такого додатку, який дозволяє обмінюватися будь-якими медіафайлами, та який можна буде легко підтримувати та додавати нові типи медіафайлів у майбутньому.

Визначено програмні засоби (інтерфейси API), які дозволяють завантажувати файли та відправляти їх користувачеві, перевагу надано засобам від Sun Microsystems: Java Servlet API, Бібліотека Retrofit. Дані засоби було вжито при створенні системи.

Спроектовано, реалізовано та випробувано систему прийняття та відправлення медіафайлів користувачів. Система складена за розподіленою клієнт-серверною архітектурою та складається з трьох 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» <https://github.com/MiroslavaBo/clickshot>

У систему впроваджено засоби покращення інформаційної безпеки: додаткове шифрування даних, що передаються комунікаційним каналом, за алгоритмом SHA1. 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. Олександр Смичніков. Параноя або сучасні реалії: як зберегти конфіденційність у мережі. - ФЬЮЧЕР МЕДІА. URL: <https://mind.ua/openmind/20183329-paranoya-abo-suchasni-realiyi-yak-zberegiti-konfidencijnist-u-merezhi> (дата звернення 30.05.2022)
4. Oracle: Документація javax.servlet. URL: <https://docs.oracle.com/javaee/7/api/javax/servlet/Servlet.html>
5. Google: Документація guice. URL: <https://github.com/google/guice/wiki/Motivation>
6. 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.
7. 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
8. Булгакова О.С. Інформатика: візуальне програмування. Навчальний посібник (стереотипне видання) / Булгакова О.С., Зосімов В.В., Броницька Н.А., Танкова Н.В. - Херсон. - 2020. - 312 с. ISBN: 978-966-289-039-6
9. Гоменюк С.І. Математичне моделювання геометричних об'єктів у паралельних комп'ютерних системах. Монографія / Гоменюк С.І., Чопоров С.В., Аль-Атамнех Б.Г. - Херсон. - 2019. - 112 с. ISBN: 978-966-916-714-9
10. Булгакова О.С. Методи та системи штучного інтелекту: теорія та практика. Навчальний посібник / Булгакова О.С., Зосімов В.В., Поздєєв В.О. - Херсон. - 2020. - 356 с. ISBN: 978-966-289-364-9
11. Покотилова В.І. Використання інформаційних технологій в теорії прийняття рішень. Навчальний посібник / Покотилова В.І., Фомішина В.М., Лугінін О.Є. - Херсон. - 2019. - 240 с. ISBN: 978-966-289-277-2
12. Величко О.М. Основи системного аналізу і прийняття оптимальних рішень. Підручник / Величко О.М., Гордієнко Т.Б. - Херсон. - 2021. - 672 с. ISBN: 978-966-289-553-7
13. Чайковська М.П. Концептуально-методологічні засади управління маркетинговими ІТ-проєктами в умовах цифрових трансформацій. Монографія / Чайковська М.П. - Херсон. - 2021. - 370 с. ISBN: 978-966-289-537-7