

## Swing MVP minta

- **Model:** Az üzleti logika, adatkezelés és a szerverrel való kommunikáció (token-kezelés, barátkezelés, üzenetküldés).
- **View:** A Swing UI komponensek, amelyek passzívan csak a felhasználói interakciót jelenítik meg.
- **Presenter:** Az összekötő réteg, amely kezeli a View eseményeit, hívja a Model műveleteit, és frissíti a View-t. Ez a réteg független a Swing implementációtól, így könnyen cserélhető más UI-ra is (pl. mobilon).

## Előnyök

- Tiszta elválasztás UI és logika között
  - Egyszerűbb Swinghez, könnyen karbantartható
  - Későbbi mobil portolásnál a Presenter és Model kód újrahasznosítható
  - Minimalizálja a Swing-specifikus kód kiterjedését
- 

## Java kliens terve a „Java-Projekt.pdf” alapján

- **Technológia:** Java 17+, Swing UI
  - **Funkciók:**
    - Felhasználók regisztrációja és bejelentkezése token alapú autentikációval, automatikus tokenfrissítéssel
    - Barátok hozzáadása, törlése, elfogadása vagy elutasítása
    - Üzenetek küldése szöveg, kép és hang formátumban egyetlen REST API végponton keresztül
    - Üzenetek és barátkérések egyesített lekérése automatikus visszaigazolással
    - Médiafájlok előfeldolgozása (tömörítés, átméretezés)
  - **Architektúra:**
    - Core (üzleti logika és adatkezelés) modul, teljesen UI-független
    - Swing UI modul MVP mintával, amely a Core modult használja
  - **Adattárolás:** Helyi SQLite adatbázis vagy fájlok kezelése, absztrahálva a Core modulban
  - **Kommunikáció:** REST API hívások kezelése a Core-ban, kivéve UI-t
- 

## MVP és Java kliens együttműködése

- A Presenter kommunikál a Core modullal és frissíti a Swing View-t.
  - A Core kezeli az API hívásokat és az üzleti állapotot.
  - A Swing UI a View implementálásával megjeleníti az adatokat, a Presenter pedig kezeli az eseményeket.
-

# Projekt architektúra

- **Java kliens:**
  - Kezeli a felhasználók regisztrációját, bejelentkezését (token alapú autentikációval).
  - Automatikusan tárolja és használja a felhasználónév-jelszó párost a token frissítéshez.
  - Barátok hozzáadását, törlését és elutasítását kezeli.
  - Üzenetek (szöveg, hang, kép) küldése egyetlen API végponton keresztül, előzetes kliens oldali média optimalizálással (átméretezés, tömörítés).
  - Egyesített lekérést végez az üzenetekre és barátkérésekre, automatikus átvétel visszaigazolással.
- **PHP szerver (REST API):**
  - Felhasználói regisztráció, bejelentkezés, token generálás és ellenőrzés (lejárattal).
  - Jelszóemlékeztető email küldés.
  - Barátok kezelése (hozzáadás, törlés, elutasítás).
  - Üzenetek és média fájlok kezelése, validálása (méret, formátum, biztonság).
  - Egyetlen `/api/sendMessage` végpont az összes üzenet és média fogadására és feldolgozására.
  - Egyesített `/api/getMessages` végpont üzenet- és barátkérés fogadására, confirm visszaigazolással.
  - MySQL adatbázis használata.

---

## Chat kiszolgáló szerver MySQL adatbázis modell

Tábla	Mezők és leírás
users	id, nickname, email, password_hash, reg_date, status
friends	id, user_id, friend_id
friend_requests	id, from_user_id, to_user_id, request_date
messages	id, sender_id, receiver_id, msg_type (text/audio/image), content (szöveg vagy fájlút), sent_date, delivered, read_status

media_files	id, owner_id, media_type, file_path, file_size, created_at
-------------	--

## API végpontok részletesen

Végpont	Módsze r	Funkció
/api/registerLogin	POST	Regisztráció/bejelentkezés, token kiadás
/api/forgotPassword	POST	Jelszóemlékeztető email küldése
/api/changeCredenti als	POST	Email és jelszó módosítása
/api/addFriend	POST	Barát hozzáadása nickname alapján
/api/deleteFriend	POST	Barát törlése vagy barátkérés elutasítása
/api/sendMessage	POST	Üzenetek fogadása (szöveg, hang, kép) egyetlen végponton
/api/getMessages	POST	Üzenetek és barátkérések lekérése, előző üzenetek confirmálása

## Folyamatok

### Hitelesítés és token kezelés

- A szerver tokeneket ad ki időkorláttal, amelyeket minden API hívásban ellenőriz.
- Ha lejár a token, a kliens a háttérben a tárolt név-jelszó párossal újra bejelentkezik a `/api/registerLogin` végponton.

## Üzenetküldés

- A kliens előkészíti a médiát (tömörítés, átméretezés), és az egységes `/api/sendMessage` végponton küldi el az üzeneteket.
- A szerver tárolja az üzeneteket, médiafájlokat validál és tárol, majd az üzenet rekordban tárolja a média hivatkozást.

## Üzenet- és barátkérés lekérése

- A kliens periodikusan (pl. 10 mp-ként) hívja a `/api/getMessages` végpontot.
- A kérés tartalmazhat konfirmálandó, előzőleg átvett üzenetek azonosítóit.
- A válaszban visszakapja az új üzeneteket és barátkéréseket.
- A szerver az átvett üzeneteket törli vagy állapotukat frissíti.

## Barátkezelés

- Barátok hozzáadása a `/api/addFriend` végponton, nickname alapján.
- Barátok törlése vagy barátkérések elutasítása a `/api/deleteFriend` végponton.