1. Buatlah high level design architecture atas project mobile apps ini.
   Jawaban :

   Lapisan Klien (Mobile App):

   - **Cross-Platform :** Framework seperti React Native atau Flutter dapat dipertimbangkan untuk pengembangan yang lebih cepat di kedua platform (*iOS dan android*), karena pengembangan native sering menawarkan kinerja yang lebih baik dan fitur khusus platform.
   - **Fitur:** Registrasi pengguna, login, manajemen profil, formulir aplikasi pinjaman, pelacakan status pinjaman, pengingat pembayaran, riwayat transaksi, tampilan notifikasi.

   API Gateway:

   - Bertindak sebagai titik masuk tunggal untuk semua permintaan klien.
   - Menangani autentikasi, otorisasi, pembatasan tingkat permintaan (rate limiting), dan perutean permintaan ke berbagai microservice.
   - Menyediakan lapisan keamanan dan abstraksi untuk layanan backend.

   Microservices Backend:

   - **Layanan Manajemen Pengguna (User Management Service):**
     - Menangani registrasi pengguna, login (hashing password, integrasi biometrik), pembaruan profil, verifikasi KTP.
     - Berintegrasi dengan penyedia autentikasi (misalnya, OAuth 2.0).
   - **Layanan Aplikasi Pinjaman (Loan Application Service):**
     - Mengelola pengajuan aplikasi pinjaman, validasi (jumlah pinjaman, tenor), dan pemrosesan.
     - Berintegrasi dengan mesin penilaian kredit (internal atau eksternal).
     - Menangani logika persetujuan/penolakan pinjaman.
   - **Layanan Pembayaran & Penagihan (Payment & Billing Service):**
     - Mengelola pencairan pinjaman, pembuatan tagihan bulanan, pelacakan pembayaran, dan perhitungan saldo terutang.
     - Berintegrasi dengan gateway pembayaran (misalnya, bank, e-wallet).
   - **Layanan Notifikasi (Notification Service):**
     - Mengirim notifikasi email dan SMS untuk pembaruan status pinjaman, pengingat pembayaran, dll.
     - Berintegrasi dengan penyedia layanan email (ESP) dan gateway SMS.

   Lapisan Database:

   - **Database Relasional (misalnya, PostgreSQL, MySQL):** Untuk data terstruktur seperti profil pengguna, aplikasi pinjaman, catatan pembayaran, dan riwayat transaksi. Memastikan integritas dan hubungan data.

- **Cache (misalnya, Redis, Memcached):** Untuk data yang sering diakses untuk mengurangi beban database dan meningkatkan waktu respons (misalnya, data sesi pengguna, syarat pinjaman yang sering dilihat).

Infrastruktur & Penyebaran (Deployment):

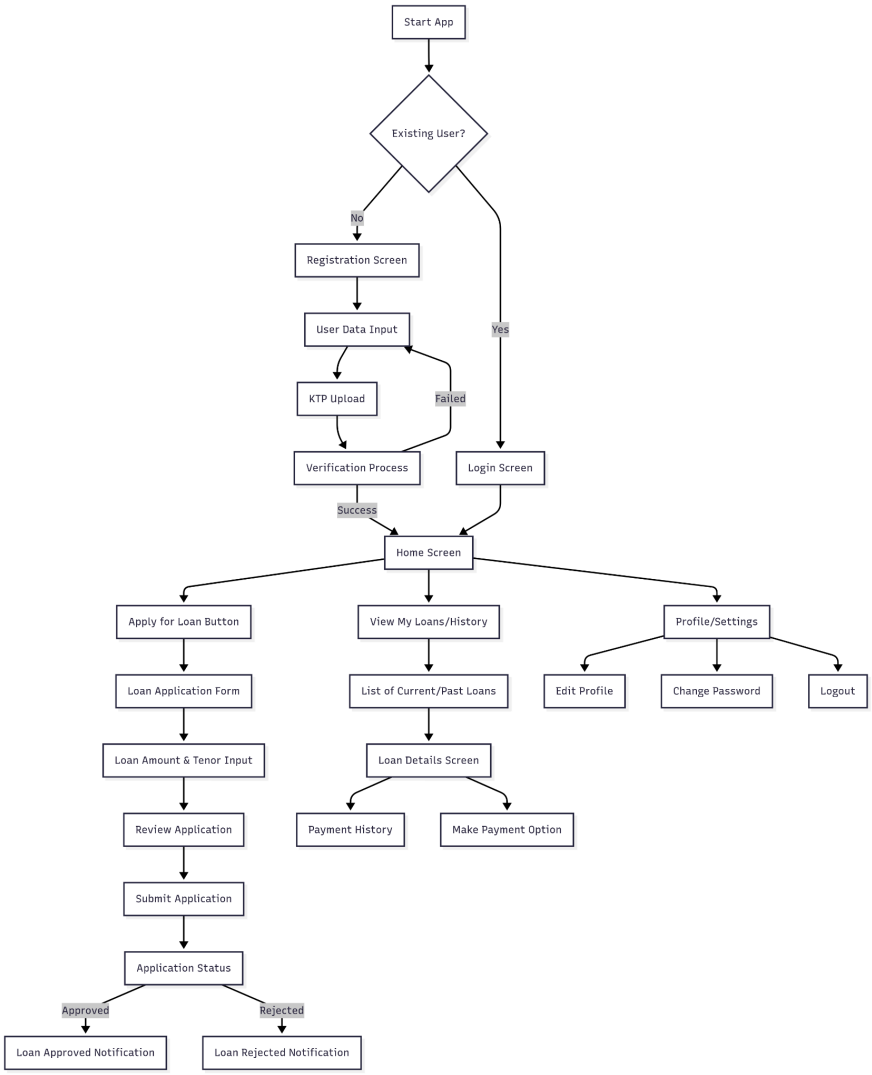
- **Platform Cloud (misalnya, AWS, GCP, Azure):** Menyediakan skalabilitas, keandalan, dan berbagai layanan terkelola.
- **Containerization (Docker):** Untuk mengemas aplikasi dan dependensinya, memastikan konsistensi di seluruh lingkungan.
- **Orkestrasi (misalnya, Kubernetes):** Untuk mengelola dan menyebarkan aplikasi dalam kontainer dalam skala besar.
- **CI/CD Pipeline (GitLab CI/CD):** Untuk pengujian otomatis, pembangunan, dan penyebaran aplikasi.

Keamanan:

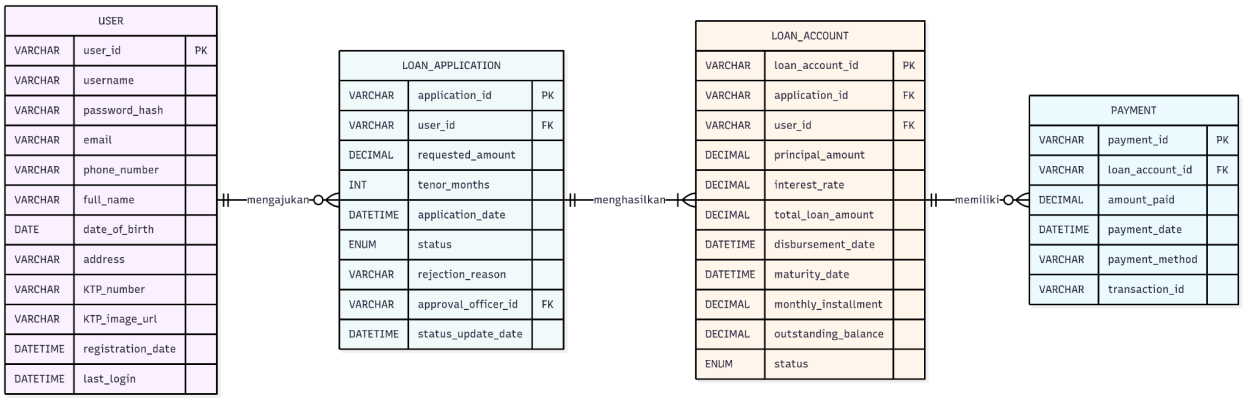
- **HTTPS/SSL:** Untuk komunikasi yang aman antara klien dan server.
- **Enkripsi Data:** Mengenkripsi data sensitif saat tidak aktif (at rest) dan saat transit.
- **Validasi Input:** Mencegah kerentanan web umum seperti injeksi SQL dan XSS.
- **Audit Keamanan Reguler:** Untuk mengidentifikasi dan mengatasi potensi kerentanan.

2. Spesifikasikan design screen flow dan ERD atas rancangan yang ingin anda buat.

   Jawaban :



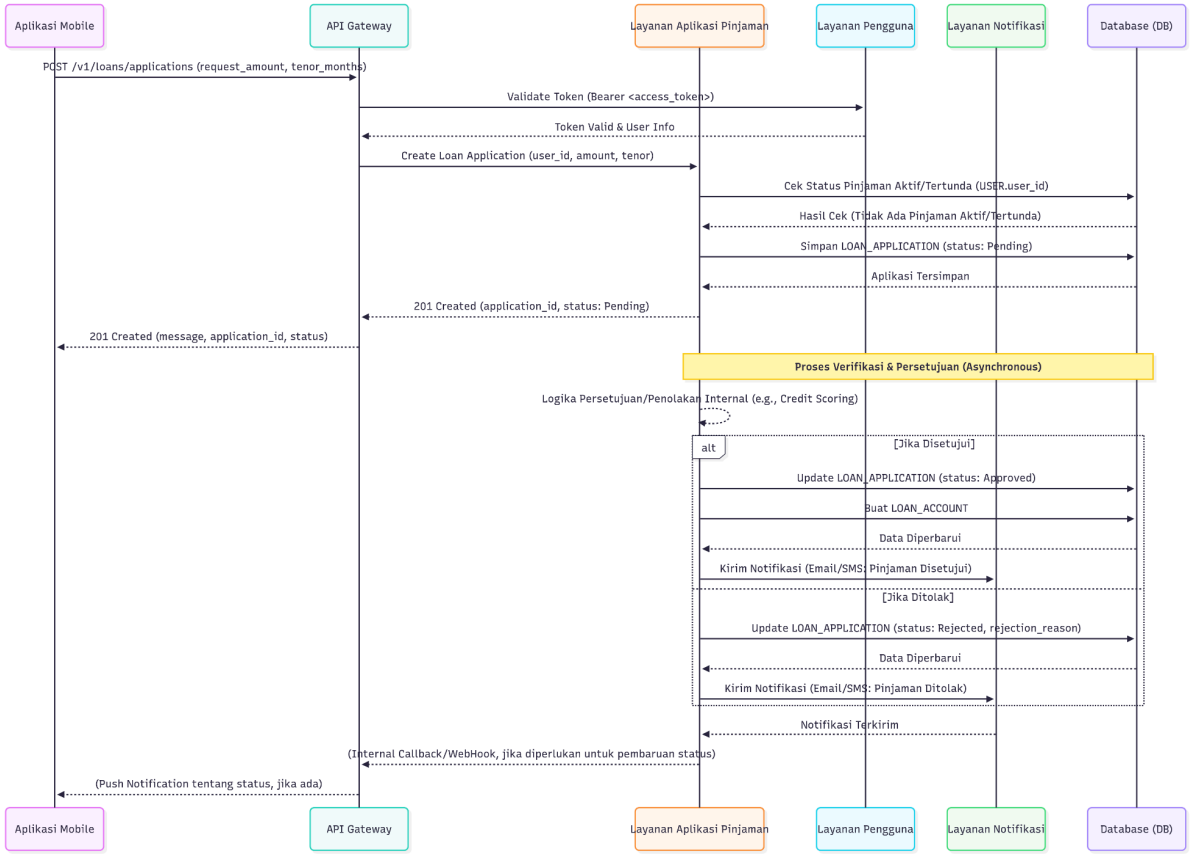
Screenflow Apps



ER Diagram

3. Buatlah detail design untuk API dengan menggunakan tools design seperti UML, ERD,flowchart etc.

   Jawaban :



4. Buatlah detail design untuk screen behavior dari mobile apps berdasarkan screen flow diatas.

   Jawaban :

   1. Perilaku Layar Registrasi:

   - **Keadaan Awal:**
     - Bidang: Nama Lengkap, Email, Nomor Telepon, Kata Sandi, Konfirmasi Kata Sandi, Nomor KTP.
     - Tombol: "Daftar".
     - Tautan: "Sudah punya akun? Masuk."
   - **Input Pengguna:**
     - **Validasi Email/Nomor Telepon:** Validasi real-time untuk format dan keunikan (melalui panggilan API). Jika sudah terdaftar, tampilkan pesan kesalahan: "Email/Nomor telepon sudah terdaftar. Silakan masuk atau gunakan yang lain."

    - **Persyaratan Kata Sandi:** Tampilkan kriteria (misalnya, minimal 8 karakter, 1 huruf besar, 1 angka, 1 karakter khusus). Tampilkan umpan balik visual (misalnya, tanda centang hijau) saat kriteria terpenuhi.
    - **Konfirmasi Kata Sandi:** Perbandingan real-time dengan bidang kata sandi. Jika tidak cocok, tampilkan kesalahan: "Kata sandi tidak cocok."
    - **Validasi Nomor KTP:** Validasi format dasar (misalnya, panjang).
- **Alur Unggah KTP:**
    - Setelah input data awal, transisi ke layar "Unggah KTP" atau bagian.
    - **Instruksi:** "Harap unggah foto KTP Anda yang jelas."
    - **Opsi:** "Ambil Foto" (membuka kamera) dan "Pilih dari Galeri" (membuka pemilih foto).
    - **Pratinjau:** Setelah memilih, tampilkan pratinjau foto KTP.
    - **Konfirmasi:** Tombol "Konfirmasi KTP".
- **Pengajuan:**
    - Saat tombol "Daftar" ditekan:
- **Panggilan API:** Kirim data registrasi (termasuk URL gambar KTP setelah berhasil diunggah ke penyimpanan cloud) ke `/auth/register`.
- **Berhasil:** Setelah 201 Created:
    - Tampilkan pesan keberhasilan (misalnya, "Registrasi berhasil! Silakan masuk.").
    - Secara otomatis navigasi ke Layar Login.
- **Gagal:** Saat terjadi kesalahan 4xx:
    - Sembunyikan indikator pemuatan.
    - Tampilkan pesan kesalahan spesifik (misalnya, "Nomor KTP tidak valid," "Kesalahan server. Silakan coba lagi.").
    - Biarkan pengguna di layar registrasi, memungkinkan mereka untuk mengoreksi input.

2. Perilaku Layar Login:

- **Keadaan Awal:**
    - Form: Nama Pengguna (Email/Telepon), Kata Sandi.
    - Tombol: "Login", "Login dengan Biometrik" (jika didukung).
    - Link: "Lupa Kata Sandi?", "Belum punya akun? Daftar."
- **Input Pengguna:**
    - **Validasi:** Validasi sisi klien dasar untuk bidang kosong.
- **Proses Login (Kata Sandi):**
    - Saat "Login" ditekan:
- **Indikator Pemuatan.**
- **Panggilan API:** Kirim kredensial ke `/auth/login`.
- **Berhasil (200 OK):**

    - Simpan `access_token` dengan aman.
    - Navigasi ke Layar Utama.
- **Gagal (401 Unauthorized, dll.):**
    - Sembunyikan indikator pemuatan.
    - Tampilkan pesan kesalahan: "Nama pengguna atau kata sandi tidak valid."
- **Proses Login (Biometrik):**
    - Jika perangkat mendukung biometrik (Face ID/Sidik Jari):
- Tampilkan tombol "Login dengan Biometrik".
- Saat ditekan: Memicu prompt autentikasi biometrik perangkat.
- Setelah autentikasi biometrik berhasil (sisi klien):
    - **Panggilan API:** Kirim token/pengidentifikasi biometrik spesifik (yang sudah terdaftar dengan akun pengguna sebelumnya) ke endpoint login biometrik khusus di backend.
    - **Berhasil:** Simpan `access_token` dan navigasi ke Layar Utama.
    - **Gagal:** Tampilkan kesalahan: "Login biometrik gagal. Silakan coba login kata sandi."

3. Perilaku Layar Utama:

- **Keadaan Awal:**
    - Tampilkan nama pengguna (misalnya, "Halo, John!").
    - Tampilkan tombol/kartu "Ajukan Pinjaman" yang menonjol.
    - Tampilkan bagian atau tombol "Lihat Pinjaman Saya", mungkin dengan ringkasan status pinjaman terbaru.
    - Bilah navigasi/tab: Beranda, Pinjaman Saya, Profil.
- **Konten Dinamis:**
    - Ambil status aplikasi pinjaman terbaru atau ringkasan saldo terutang dari backend (misalnya, `/loans/applications?status=Pending` atau `/loans/accounts`).
    - Jika tidak ada pinjaman aktif, tombol "Ajukan Pinjaman" harus menonjol.
    - Jika ada aplikasi yang sedang berjalan, tampilkan statusnya dengan jelas.
    - Jika ada pinjaman aktif, tampilkan saldo terutang dan tanggal jatuh tempo berikutnya.
- **Navigasi:**
    - Mengetuk "Ajukan Pinjaman" menavigasi ke Formulir Aplikasi Pinjaman.
    - Mengetuk "Lihat Pinjaman Saya" menavigasi ke layar Daftar Pinjaman.
    - Mengetuk "Profil" menavigasi ke layar Profil.

4. Perilaku Formulir Aplikasi Pinjaman:

- **Keadaan Awal:**

    - Bidang input untuk "Jumlah yang Diminta" (pemilih angka atau input teks dengan validasi) dan "Tenor (Bulan)" (misalnya, dropdown dari 1 hingga 12).
    - Tampilkan jumlah pinjaman maksimum yang diizinkan (Rp 12.000.000) dan tenor (1 tahun) sebagai pedoman.
    - Tombol: "Lanjutkan" atau "Tinjau Aplikasi".
- **Input Pengguna:**
    - **Input Jumlah:** Batasi hanya angka, pastikan tidak melebihi Rp 12.000.000. Berikan pesan kesalahan yang jelas jika dilanggar.
    - **Input Tenor:** Pastikan antara 1 hingga 12 bulan.
- **Perilaku Layar Tinjau Aplikasi:**
    - Tampilkan ringkasan pinjaman yang diminta: "Jumlah: Rp X", "Tenor: Y bulan".
    - Hitung dan tampilkan perkiraan angsuran bulanan. (Ini bisa berupa perhitungan sisi klien berdasarkan tingkat bunga yang telah ditentukan, atau panggilan API untuk mendapatkan penilaian awal).
    - **Tombol:** "Konfirmasi dan Kirim".
    - **Tombol:** "Edit" atau "Kembali" untuk kembali dan mengubah.
- **Pengajuan:**
    - Saat "Konfirmasi dan Kirim" ditekan:
- **Indikator Pemuatan.**
- **Panggilan API:** Kirim data ke `/loans/applications`.
- **Berhasil (201 Created):**
    - Tampilkan pesan keberhasilan: "Aplikasi pinjaman berhasil diajukan! Kami akan memberitahu Anda tentang statusnya."
    - Navigasi ke layar "Status Aplikasi Pinjaman (Tertunda)" atau kembali ke layar Utama dengan status yang diperbarui.
- **Gagal:**
    - Sembunyikan indikator pemuatan.
    - Tampilkan kesalahan: "Gagal mengajukan aplikasi. Harap periksa input Anda atau coba lagi nanti." (misalnya, jika pengguna memiliki pinjaman yang ada: "Anda memiliki proses pinjaman yang sedang berjalan atau pinjaman yang belum dilunasi. Tidak dapat mengajukan pinjaman baru.")

5. Perilaku Layar Status Pinjaman / Pinjaman Saya:

- **Keadaan Awal:**
    - Tampilkan daftar semua aplikasi/akun pinjaman (aktif, disetujui, ditolak, tertunda).
    - Setiap item menampilkan: ID Aplikasi/ID Pinjaman, Jumlah, Tenor, Status, Tanggal.
- **Interaksi:**
    - Mengetuk entri pinjaman tertentu akan menavigasi ke "Layar Detail Pinjaman".

- **Pembaruan Dinamis:**
  - Aplikasi idealnya harus menyegarkan daftar ini saat diluncurkan atau ditarik untuk menyegarkan (pull-to-refresh) untuk mencerminkan status terbaru.
  - **Notifikasi:** Ketika status pinjaman berubah (Disetujui/Ditolak), notifikasi push harus terpicu, dan mengetuknya harus deep-link ke layar detail pinjaman yang relevan.

6. Perilaku Layar Detail Pinjaman:

- **Keadaan Awal:**
  - Tampilkan detail komprehensif dari pinjaman yang dipilih:
    - ID Akun Pinjaman/ID Aplikasi
    - Jumlah Pokok, Tingkat Bunga, Total Jumlah Pinjaman
    - Tanggal Pencairan, Tanggal Jatuh Tempo
    - Angsuran Bulanan
    - Saldo Terutang Saat Ini
    - Status (Aktif, Tertutup, Tertunda, Disetujui, Ditolak)
    - Tanggal Jatuh Tempo Berikutnya (jika aktif)
    - Alasan Penolakan (jika ditolak)
  - Tombol: "Lihat Riwayat Pembayaran" (jika pinjaman aktif/tertutup).
  - Tombol: "Lakukan Pembayaran" (jika pinjaman aktif dengan saldo terutang).
- **Alur Pembayaran (saat "Lakukan Pembayaran" ditekan):**
  - **Dialog Konfirmasi:** "Apakah Anda yakin ingin melakukan pembayaran sebesar Rp X?"
  - **Pemilihan Metode Pembayaran:** Jika ada beberapa opsi (misalnya, Virtual Account, E-wallet), izinkan pengguna memilih.
  - **Panggilan API:** Memulai pembayaran melalui `/loans/accounts/{loan_account_id}/payments`.
  - **Integrasi Eksternal:** Jika menggunakan gateway pembayaran, aplikasi mungkin perlu membuka webview atau mengalihkan ke aplikasi gateway pembayaran.
  - **Callback/Pembaruan Status:** Setelah pembayaran, aplikasi harus mendengarkan callback atau secara berkala memeriksa status pembayaran di backend untuk memperbarui saldo terutang.
  - **Pesan Keberhasilan:** "Pembayaran berhasil! Saldo terutang Anda telah diperbarui."
  - **Pesan Kegagalan:** "Pembayaran gagal. Silakan coba lagi atau hubungi dukungan."

Perilaku Umum Aplikasi Mobile:

- **Mode Offline/Caching:** Untuk konten statis atau data yang baru dilihat, implementasikan caching untuk memberikan pengalaman yang lebih lancar saat offline atau dengan konektivitas yang buruk.
- **Notifikasi Push:**
  - Dipicu oleh backend untuk perubahan status pinjaman (Disetujui/Ditolak).
  - Dipicu untuk pengingat pembayaran.
  - Mengetuk notifikasi harus menavigasi ke layar yang relevan.
- **Indikator Pemuatan:** Gunakan status pemuatan yang sesuai (spinner, skeleton screen) untuk semua panggilan API untuk meningkatkan kinerja yang dirasakan.
- **Penanganan Kesalahan:** Tampilkan pesan kesalahan yang ramah pengguna untuk masalah jaringan, kesalahan server, atau input tidak valid.
- **Validasi Formulir:** Validasi sisi klien untuk umpan balik instan, diikuti dengan validasi sisi server untuk integritas data yang kuat.
- **Integrasi Biometrik:** Mengintegrasikan dengan mulus dengan biometrik perangkat untuk login di mana tersedia dan diaktifkan oleh pengguna.
- **Aksesibilitas:** Pastikan aplikasi dapat digunakan oleh orang-orang dengan disabilitas (misalnya, ukuran teks yang tepat, kontras, kompatibilitas pembaca layar).
- **Keamanan:** Terapkan penyimpanan aman untuk token, komunikasi HTTPS yang tepat, dan sanitasi input.