

Test Application Design (Soal)

PT. XYZ adalah sebuah perusahaan fintech yang ingin mengembangkan mobile apps mereka, dalam upaya menjangkau pengguna yang lebih luas mereka ingin mengembangkan aplikasi pinjaman online.

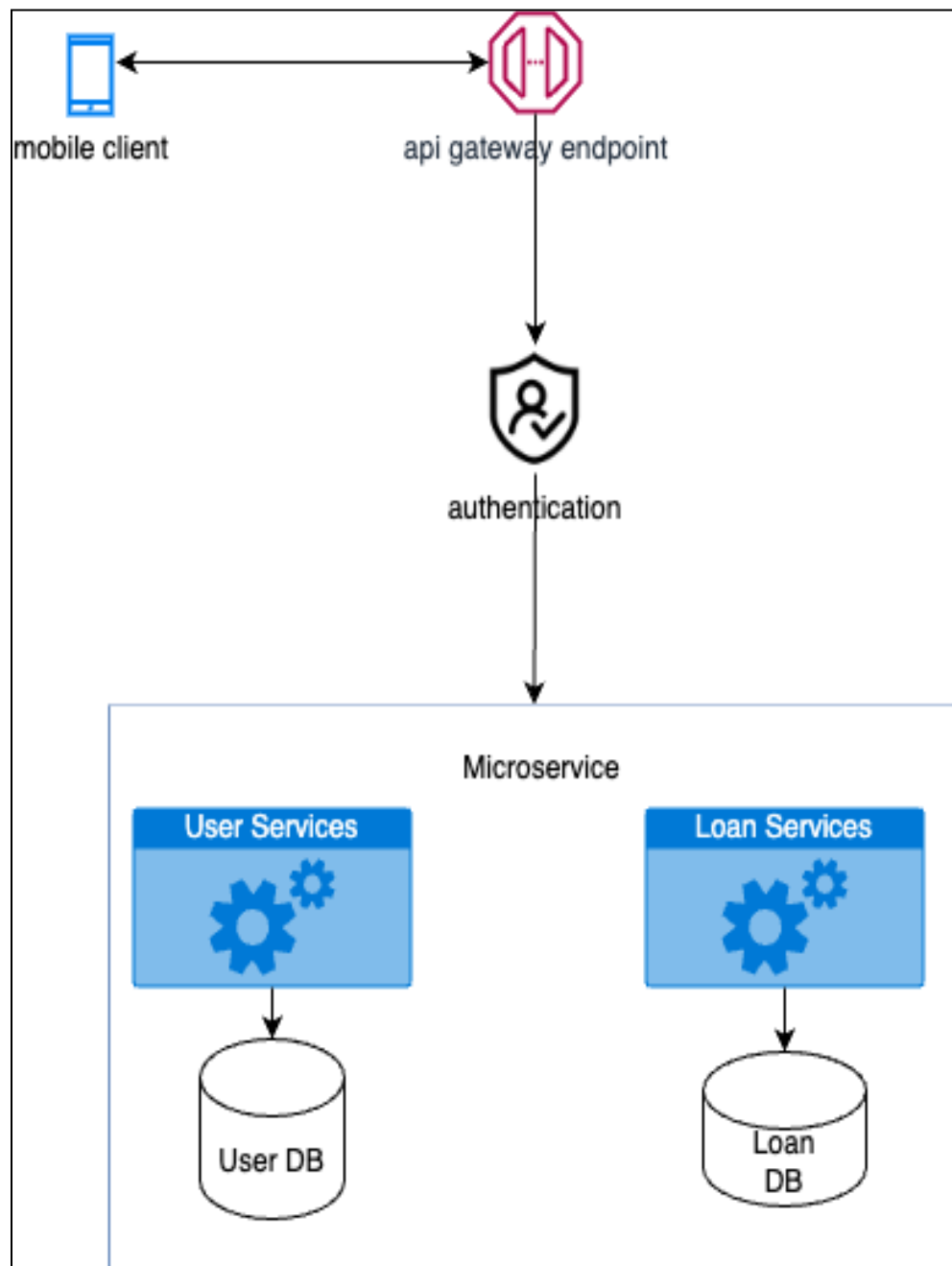
Potential High Level User Story:

- User melakukan registrasi dengan data diri, email, nomor telepon dan upload foto beserta KTP
- User dapat login dengan password atau biometric (jika ada di perangkat mobilya)
- User dapat melihat Sisa hutang dan tagihan perbulan yang harus di bayarkan (Jika ada)
- User dapat meminjam uang paling besar Rp. 12.000.000 dengan tenor maksimal 1 taun.
- User dalam proses peminjaman akan di proses dengan hasil diterima atau ditolak
- Jika pinjaman diterima maka akan ada notifikasi lewat email dan nomot telepon yang terdaftar
- User tidak dapat melakukan peminjaman uang jika sedang ada proses peminjaman dan belum di lunaskan.

Tugas Anda:

1. Buatlah high level design architecture atas project mobile apps ini.
2. Spesifikasikan design Screen Flow dan ERD atas rancangan yang ingin anda buat.
3. Buatlah detail design untuk API dengan menggunakan tools design seperti UML, ERD, flowchart etc.
4. Buatlah detail design untuk screen behavior dari mobile apps berdasarkan screen flow diatas.

1. High level design architecture atas project mobile apps ini.

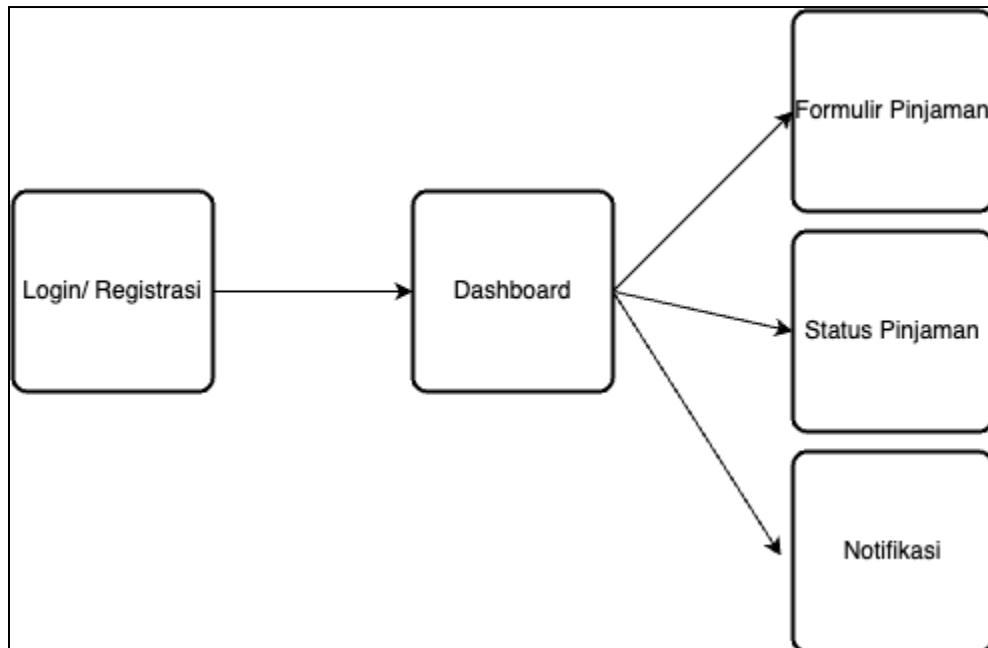


Berikut adalah penjelasan singkat tentang bagian-bagian tersebut:

1. **Mobile Client:** Aplikasi mobile yang akan digunakan oleh pengguna.
2. **API Gateway:** Sebagai pintu gerbang ke sistem backend dan dapat melakukan beberapa hal seperti throttling, routing, dan lainnya.
3. **Authentication Service:** Layanan yang bertanggung jawab untuk otentikasi dan otorisasi pengguna. Ini akan memastikan bahwa pengguna yang mencoba untuk mengakses adalah pengguna yang sah dan memiliki hak untuk melakukan operasi yang diminta.
4. **Microservices / Serverless Functions:** Layanan-layanan yang secara khusus mengurus suatu aspek dari aplikasi. Dalam kasus ini, kita memiliki dua layanan:
 - a. **User Service:** Mengurus semua hal yang berkaitan dengan pengguna, seperti manajemen profil dan data pengguna.
 - b. **Loan Service:** Mengurus semua hal yang berkaitan dengan pinjaman, seperti mengurus permohonan pinjaman, pengecekan kelayakan, dan manajemen hutang.
5. **User DB:** Basis data yang menyimpan informasi tentang pengguna. Ini akan diurus oleh User Service.
6. **Loan DB:** Basis data yang menyimpan informasi tentang pinjaman. Ini akan diurus oleh Loan Service.

2. Spesifikasi design Screen Flow dan ERD atas rancangan yang ingin dibuat.

a. Screen Flow

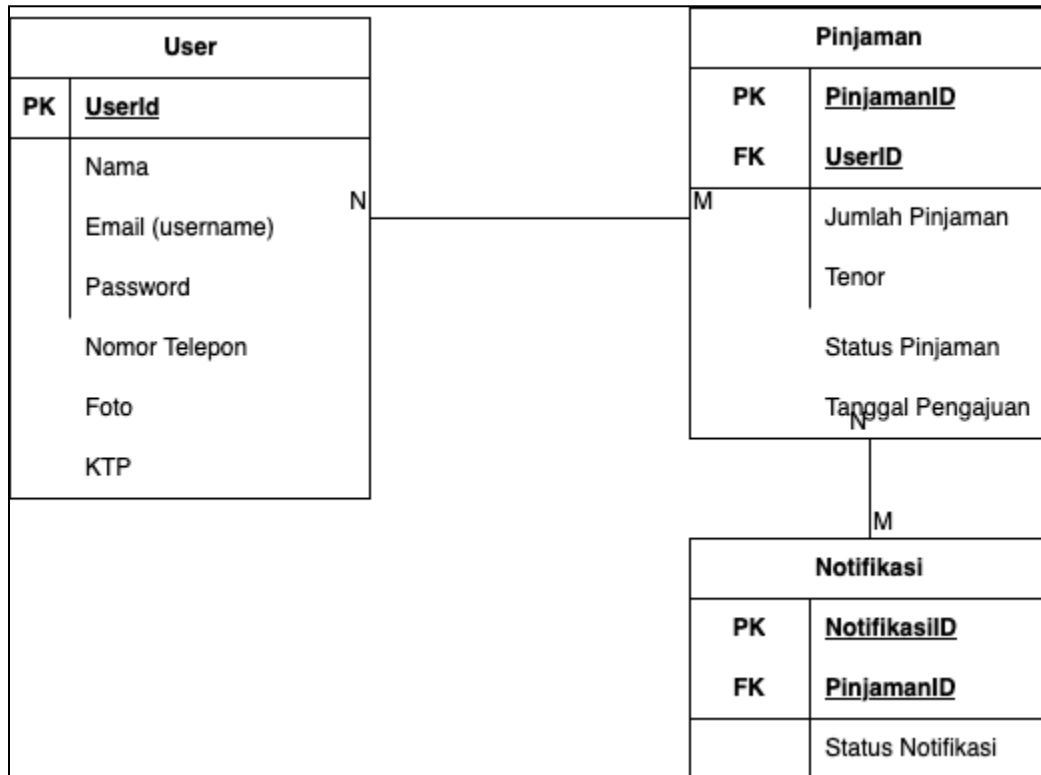


Berikut adalah penjelasan singkat tentang bagian-bagian tersebut:

1. **Login / Registrasi:** Pengguna melakukan registrasi atau login. Jika registrasi, mereka harus mengisi data diri, email, nomor telepon, dan mengunggah foto serta KTP.
2. **Dashboard:** Setelah login, pengguna dibawa ke halaman dashboard yang menunjukkan informasi tentang pinjaman mereka, jika ada. Informasi ini mencakup jumlah pinjaman, sisa hutang, dan tagihan bulanan.
3. **Formulir Pinjaman:** Pengguna dapat pergi ke halaman ini untuk mengajukan pinjaman baru. Mereka dapat memilih jumlah dan tenor pinjaman, dengan batas maksimal Rp. 12.000.000 dan 1 tahun.

4. **Status Pinjaman:** Setelah pengajuan, pengguna akan dibawa ke halaman status pinjaman, di mana mereka bisa melihat apakah pinjaman mereka diterima atau ditolak.
5. **Notifikasi:** Jika pinjaman diterima, pengguna akan menerima notifikasi melalui email dan nomor telepon yang terdaftar.

b. ERD



Berikut adalah penjelasan singkat tentang bagian-bagian tersebut:

1. **Entity User:** Memiliki atribut ID Pengguna, Nama, Email (digunakan sebagai username), Password (biasanya disimpan sebagai hash password, bukan plaintext), Nomor Telepon, Foto, dan KTP. Dalam hal ini, satu Pengguna masih bisa memiliki banyak Pinjaman, dan satu Pinjaman hanya bisa dimiliki oleh satu Pengguna.
2. **Entity Pinjaman:** Memiliki atribut ID Pinjaman, ID User (kunci asing), Jumlah Pinjaman, Tenor, Status Pinjaman, dan Tanggal Pengajuan.
3. **Entity Notifikasi:** Memiliki atribut ID Notifikasi, ID Pinjaman (kunci asing), dan Status Notifikasi.

Hubungan antara entitas tersebut adalah sebagai berikut:

- Satu User bisa memiliki banyak Pinjaman, namun satu Pinjaman hanya bisa dimiliki oleh satu User. Hubungan ini adalah one-to-many dari User ke Pinjaman.
- Satu Pinjaman bisa memiliki banyak Notifikasi, namun satu Notifikasi hanya bisa berhubungan dengan satu Pinjaman. Hubungan ini adalah one-to-many dari Pinjaman ke Notifikasi.

3. Design untuk API dengan menggunakan tools design

Dalam konteks ini, kita akan fokus pada dua layanan utama yaitu *User Service* dan *Loan Service*.

1. User Service

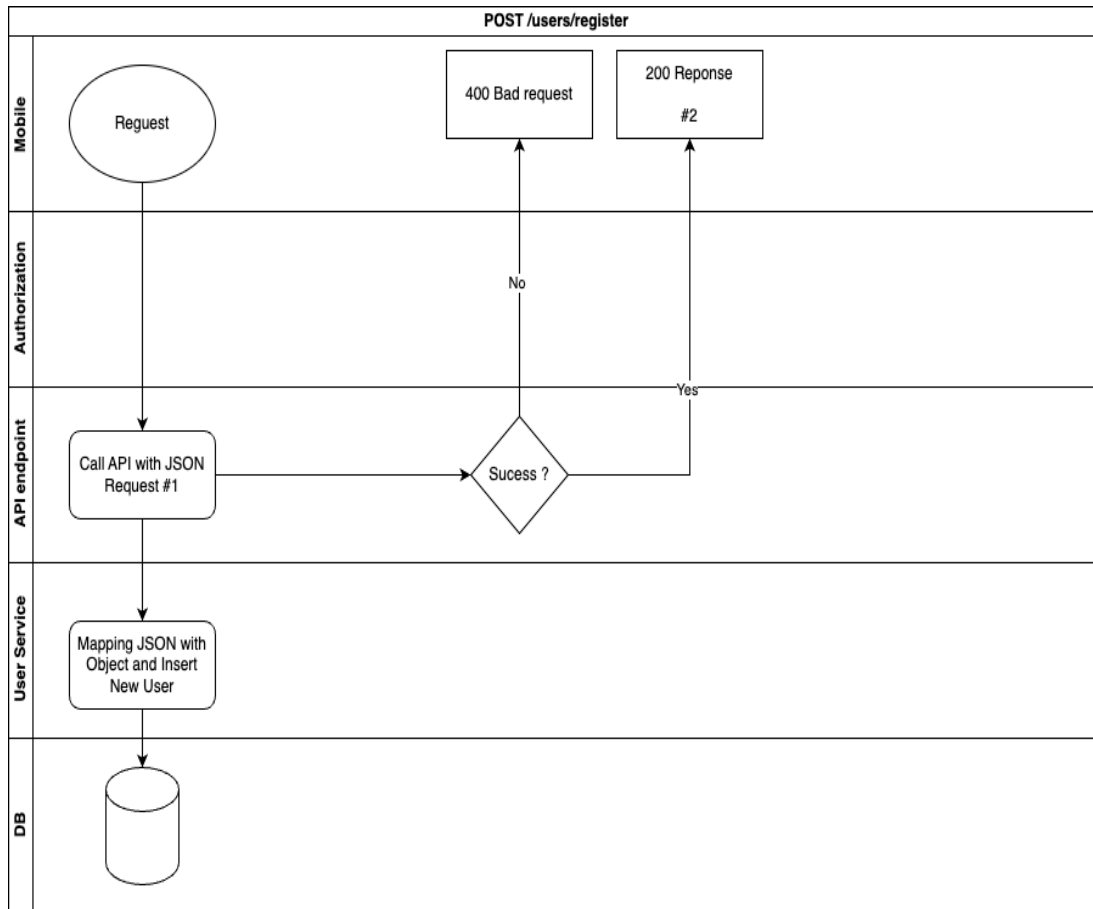
- Register API (POST /users/register)
- Login API (POST /users/login)
- Get User Profile API (GET /users/{userId})

2. Loan Service

- Create Loan API (POST /loans)
- Get Loan API (GET /loans/{loanId})

a. **Flow Chart Register API (POST /users/register)**

Menerima data pengguna (nama, email, nomor telepon, foto, dan KTP) sebagai input dan mengembalikan ID pengguna yang baru dibuat sebagai output.



Register Request #1

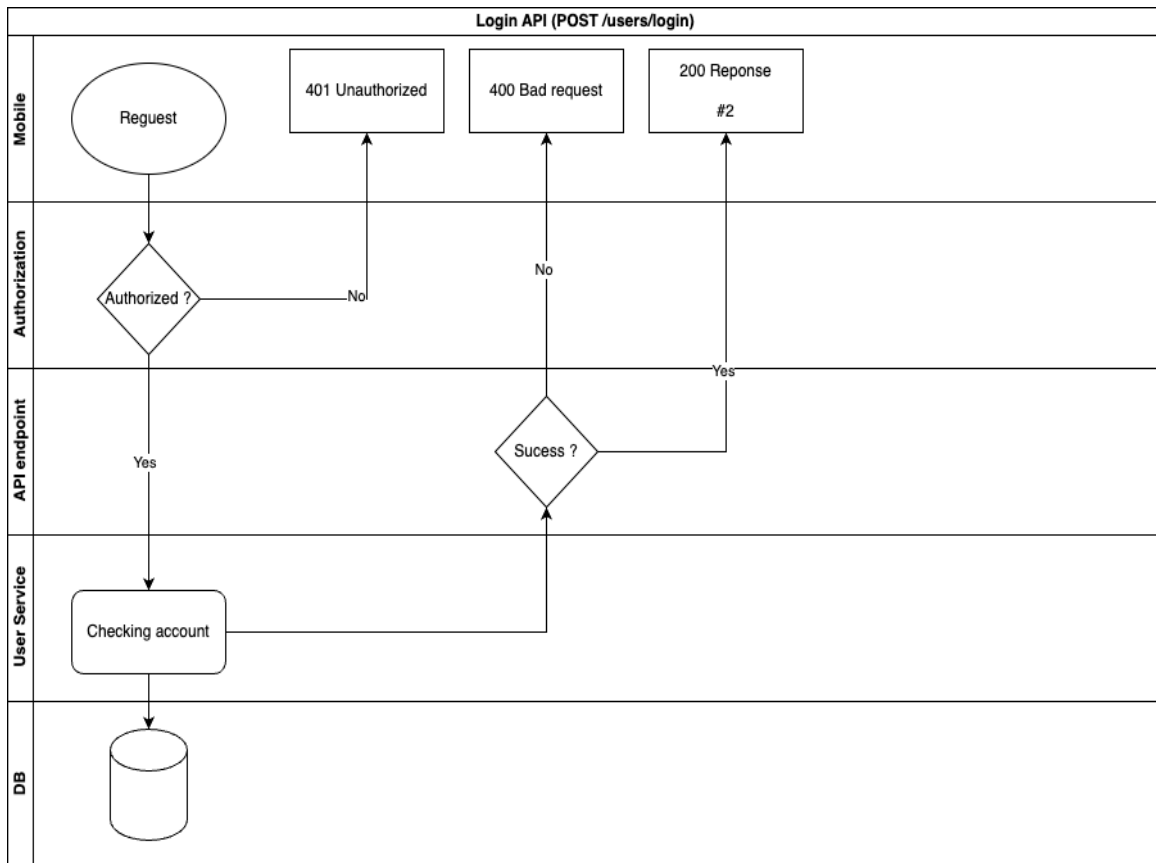
```
{
  "name": "John Doe",
  "email": "john@example.com",
  "phone": "08123456789",
  "password": "password",
  "photo": "<base64_encoded_photo>",
  "ktp": "<base64_encoded_ktp>"
}
```

Register Response #2

```
{
  "userId": "123"
}
```


b. Flow Chart **Login API (POST /users/login)**

Menerima email dan password sebagai input dan mengembalikan token autentikasi yang bisa digunakan untuk mengakses API lainnya.



Login Request #1

```

{
  "email": "john@example.com",
  "password": "password"
}

```

Login Response #2

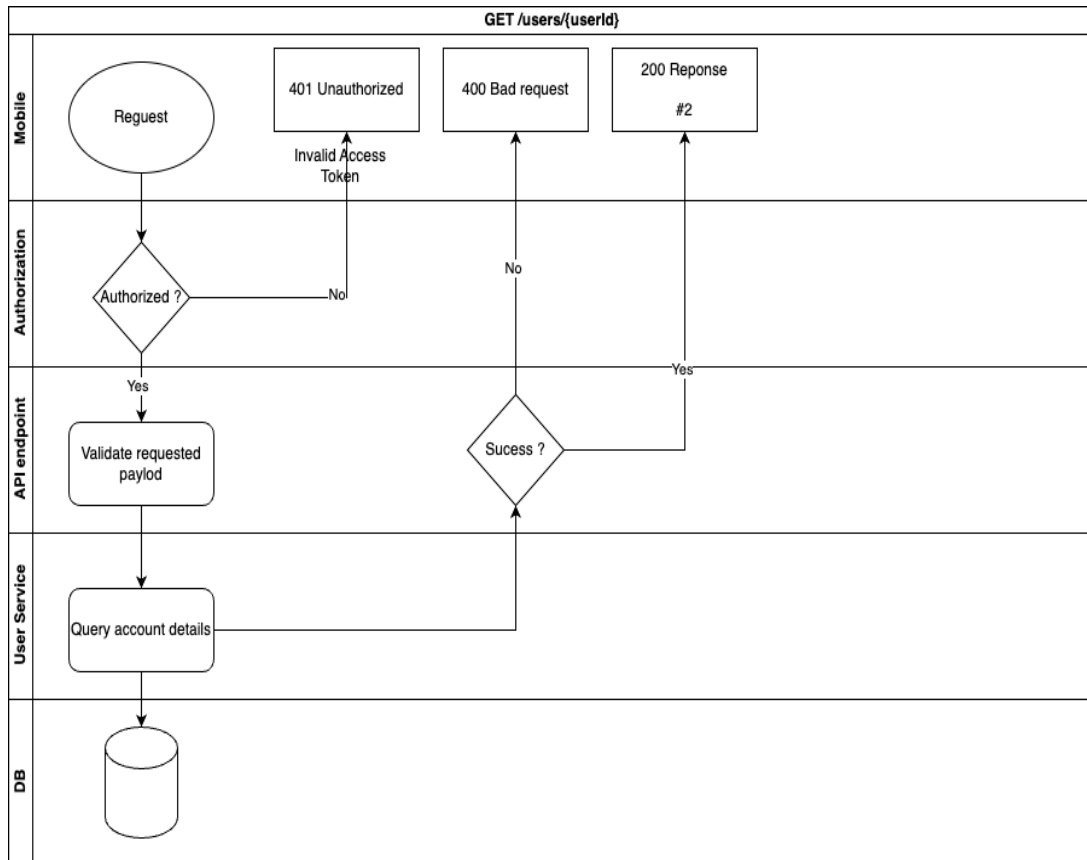
```

{
  "token": "abc123"
}

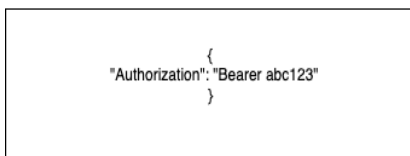
```

c. Flow Chart **Get User Profile API (GET /users/{userId})**

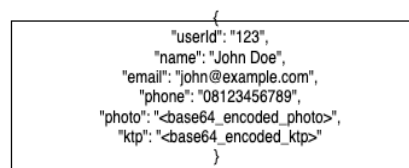
Menerima token autentikasi dan ID pengguna sebagai input dan mengembalikan data profil pengguna.



User Profile Request #1

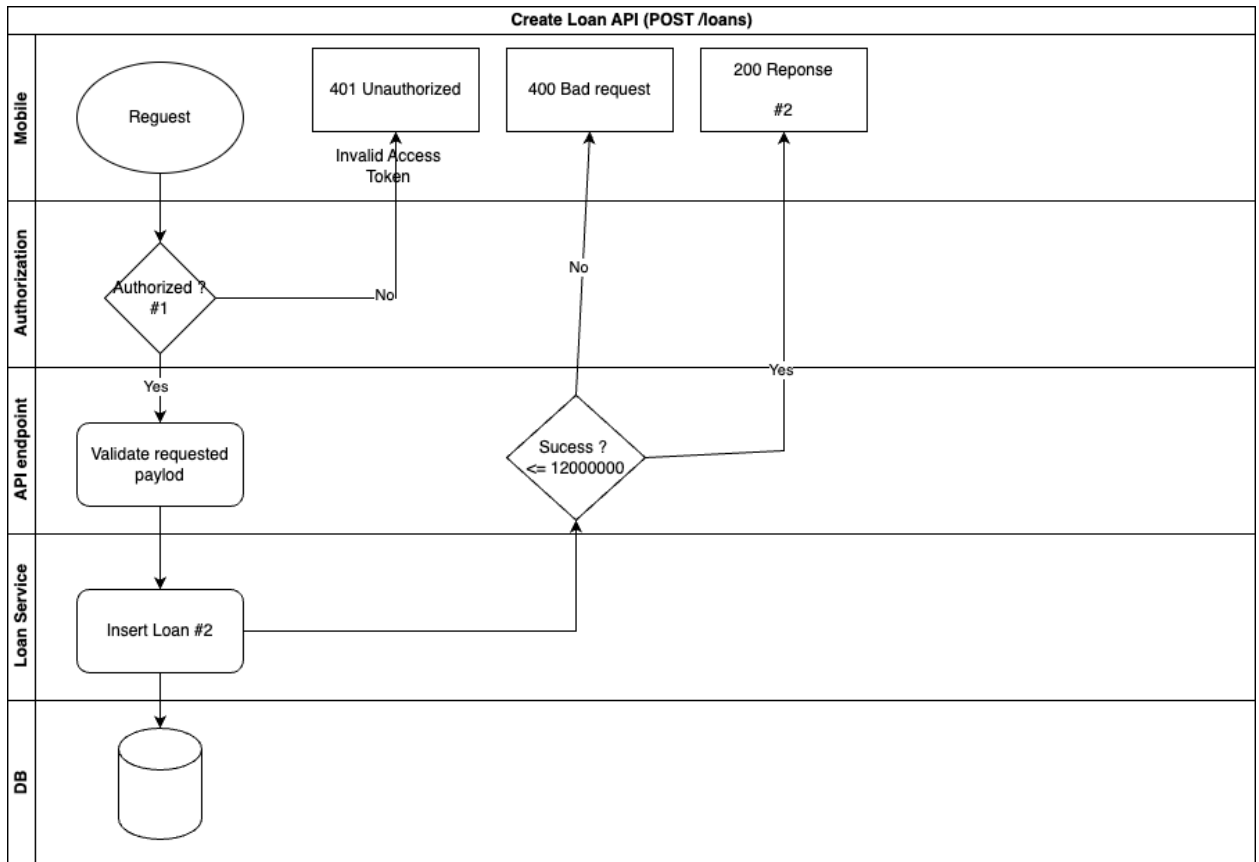


User Profile Response #2



d. Flow Chart **Create Loan API (POST /loans)**

Menerima token autentikasi, ID pengguna, jumlah pinjaman, dan tenor sebagai input dan mengembalikan ID pinjaman yang baru dibuat sebagai output.



Create Loan Request #1

```

{
  "Authorization": "Bearer abc123"
}

```

Create Loan Response #2

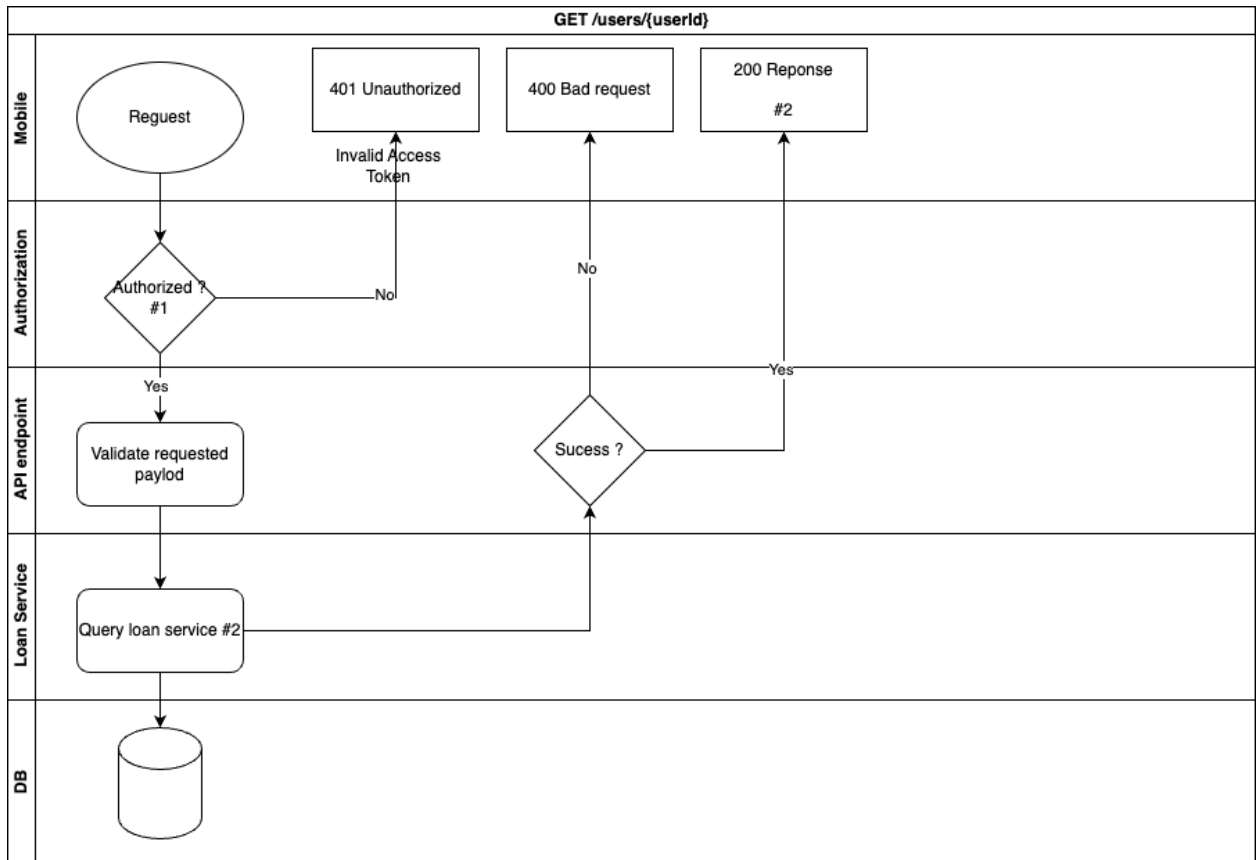
```

{
  "userId": "123",
  "amount": 12000000,
  "tenor": 12
}

```

e. Flow Chart **Get Loan API (GET /loans/{loanId})**

Menerima token autentikasi dan ID pengguna sebagai input dan mengembalikan daftar pinjaman pengguna.



Loan Request #1

```

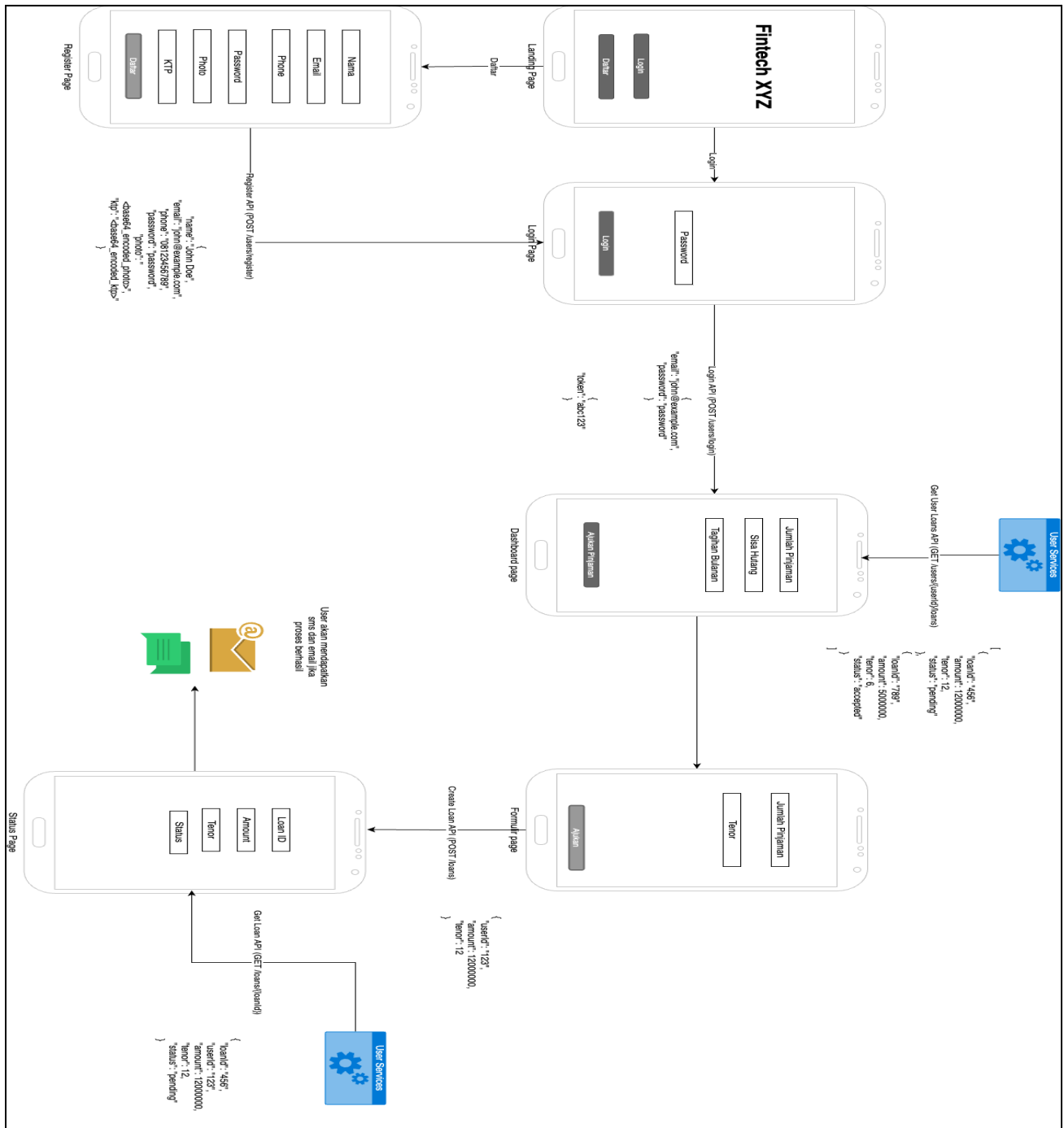
{
  "Authorization": "Bearer abc123"
}
  
```

Loan Response #2

```

{
  "loanId": "456",
  "userId": "123",
  "amount": 12000000,
  "tenor": 12,
  "status": "pending"
}
  
```

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



Berikut adalah rincian perilaku layar berdasarkan alur layar yang telah kita tentukan:

1. Login / Registrasi:

- Ketika pengguna membuka aplikasi, mereka dihadapkan dengan layar login.
- Jika pengguna baru, mereka bisa pilih "Daftar" untuk pergi ke layar registrasi.
- Di layar registrasi, pengguna diharapkan untuk mengisi semua detail yang diperlukan dan mengunggah foto mereka dan KTP. Setelah menekan "Daftar", informasi ini dikirim ke backend melalui Register API. Jika berhasil, pengguna dialihkan ke layar login.
- Di layar login, pengguna memasukkan email dan password mereka. Informasi ini dikirim ke backend melalui Login API. Jika berhasil, pengguna diberikan token dan diarahkan ke layar dashboard.

2. Dashboard:

- Ini adalah layar utama setelah pengguna berhasil login.
- Di layar ini, pengguna dapat melihat detail pinjaman mereka seperti jumlah pinjaman, sisa hutang, dan tagihan bulanan. Informasi ini diperoleh dari backend melalui Get User Loans API.
- Jika pengguna tidak memiliki pinjaman yang aktif, mereka dapat melihat opsi untuk membuat pinjaman baru.

3. Formulir Pinjaman:

- Di layar ini, pengguna dapat mengajukan pinjaman baru.
- Mereka perlu memasukkan jumlah dan tenor pinjaman. Setelah menekan "Ajukan", informasi ini dikirim ke backend melalui Create Loan API.
- Jika berhasil, pengguna diberikan ID pinjaman dan dialihkan ke layar Status Pinjaman.

4. Status Pinjaman:

- Di layar ini, pengguna dapat melihat status pinjaman mereka, apakah diterima atau ditolak. Status ini diperoleh dari backend melalui Get Loan API.
- Jika pinjaman diterima, pengguna akan menerima notifikasi melalui email dan SMS.