

## Test Application Design

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.

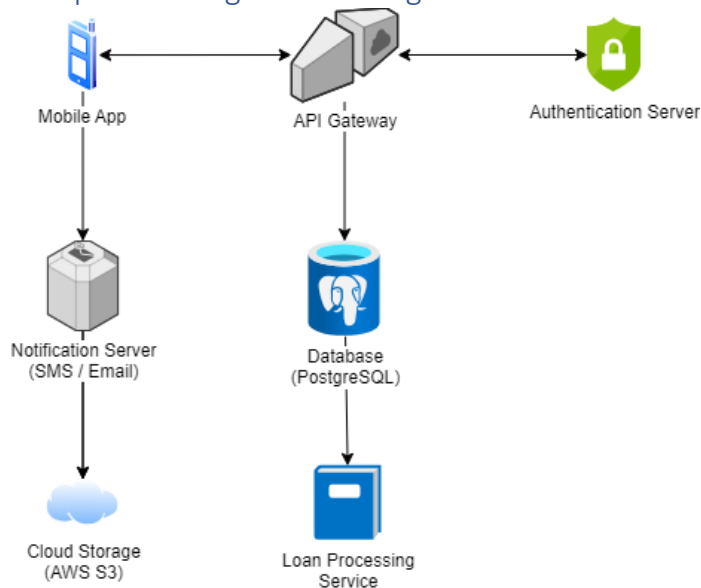
Potential High Level User Story:

- 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 mobilynya)
- 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.

Components High Level Design Architecture:



**a. Mobile Application:**

- i. Platform: iOS and Android.
- ii. Language: Swift (iOS) and Kotlin (Android) or Flutter (Dart)

**b. Backend Server:**

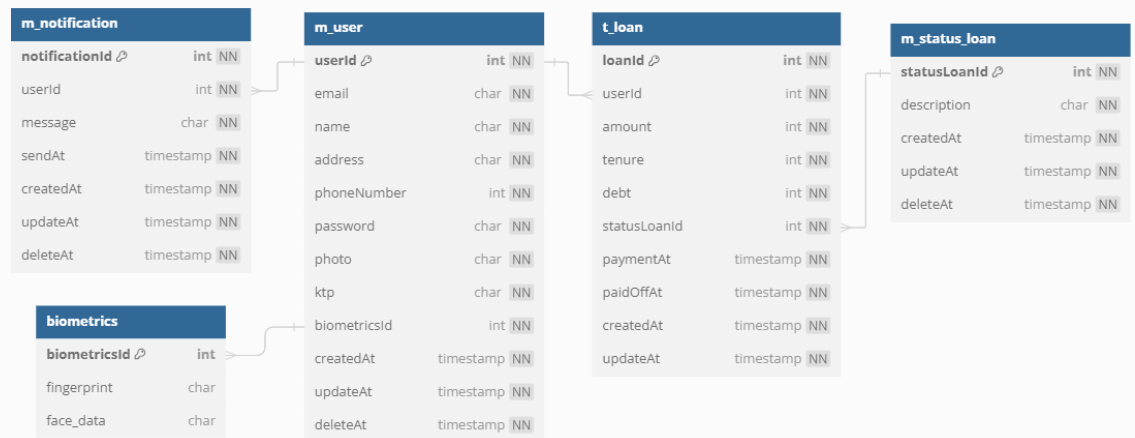
- i. Server: Node.js with Express.js framework.

- ii. Database: PostgreSQL.
  - iii. Authentication: JWT for token-based authentication.
  - iv. Cloud Storage: AWS S3 for storing user photos and KTP images.
- c. Third-Party Integrations:**
- i. SMS and Email Service: Twilio or SendGrid for notifications.
  - ii. Biometric Authentication: Integrated with device capabilities.
- d. Security:**
- i. HTTPS for secure communication.
  - ii. Encryption for sensitive data (e.g., KTP images, passwords).
2. Spesifikasikan design Screen Flow dan ERD atas rancangan yang ingin anda buat.

#### Screen Flow:

1. Welcome Screen: Options to Register or Login.
2. Registration Screen: Form for data input (email, phone number, photo, KTP upload).
3. Login Screen: Options for password or biometric login.
4. Dashboard: Displays outstanding loans and monthly bills.
5. Loan Application Screen: Form to apply for a loan (amount, tenure).
6. Loan Status Screen: Displays status of loan applications (approved, rejected, pending, paid, outstanding bills).
7. Notifications: Display email and SMS notifications for loan approval.

#### ERD (Entity-Relationship Diagram):

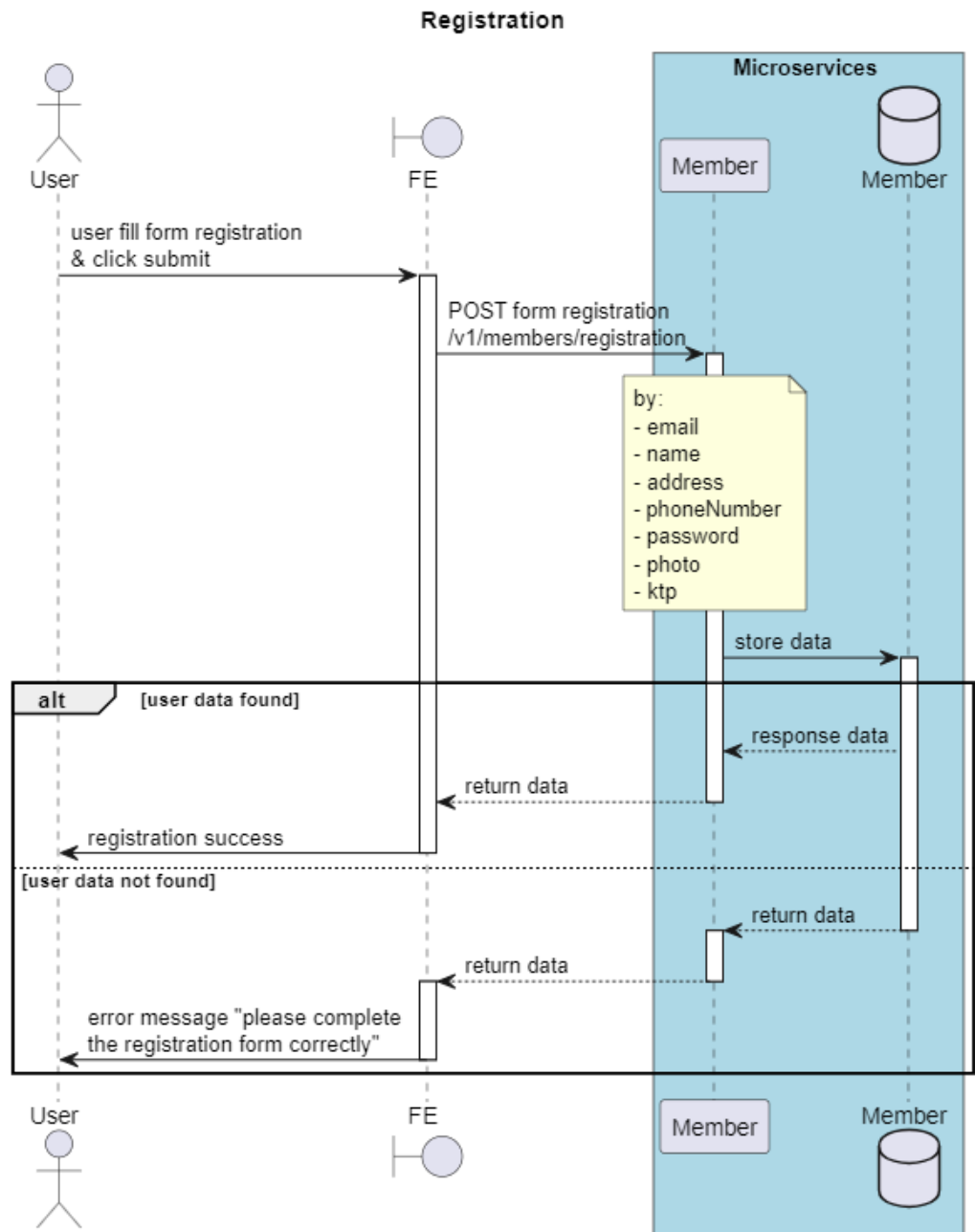


#### Note :

- o Status loan description : approved, rejected, pending
3. Buatlah detail design untuk API dengan menggunakan tools design seperti UML, ERD, flowchart etc.

## User Registration

### Sequence Diagram

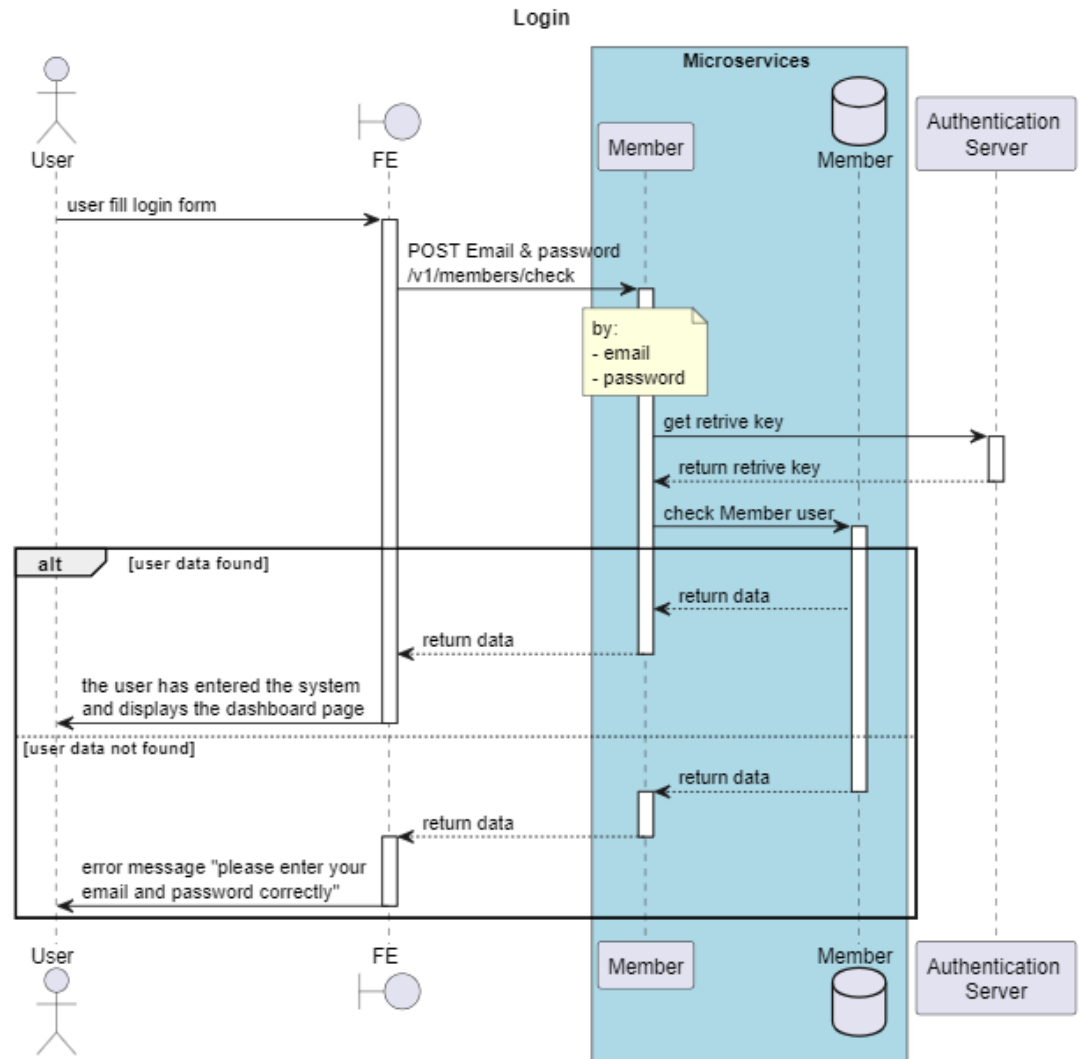


### API Contract

Method	POST
Endpoint	{{host}}ms-members/v1/members/registration
Request	Header X-localization: {{LANGUAGE}} Content-Type:application/json x-api-Key: {{API_KEY}}

	channel: {{CHANNEL}}
	Body <pre> {   "email": "user@example.com",   "phoneNumber": "08123456789",   "password": "password123",   "photo": "base64-encoded-image",   "ktp": "base64-encoded-image" } </pre>
Response	<b>success</b> <pre> {   "status": true,   "code": "RU001",   "message": true,   "errorMessage": null,   "errorData": null,   "data": {     "message": "Registration successful",     "userId": "12345"   } } </pre> <b>failed</b> <pre> {   "status": true,   "code": "RU002",   "message": null,   "errorMessage": "Registration failed",   "errorData": {     "errorTitle": "Registration Failed",     "errorDesc": "Please check your email/password"   },   "data": null } </pre>

## User Auth Login Sequence Diagram



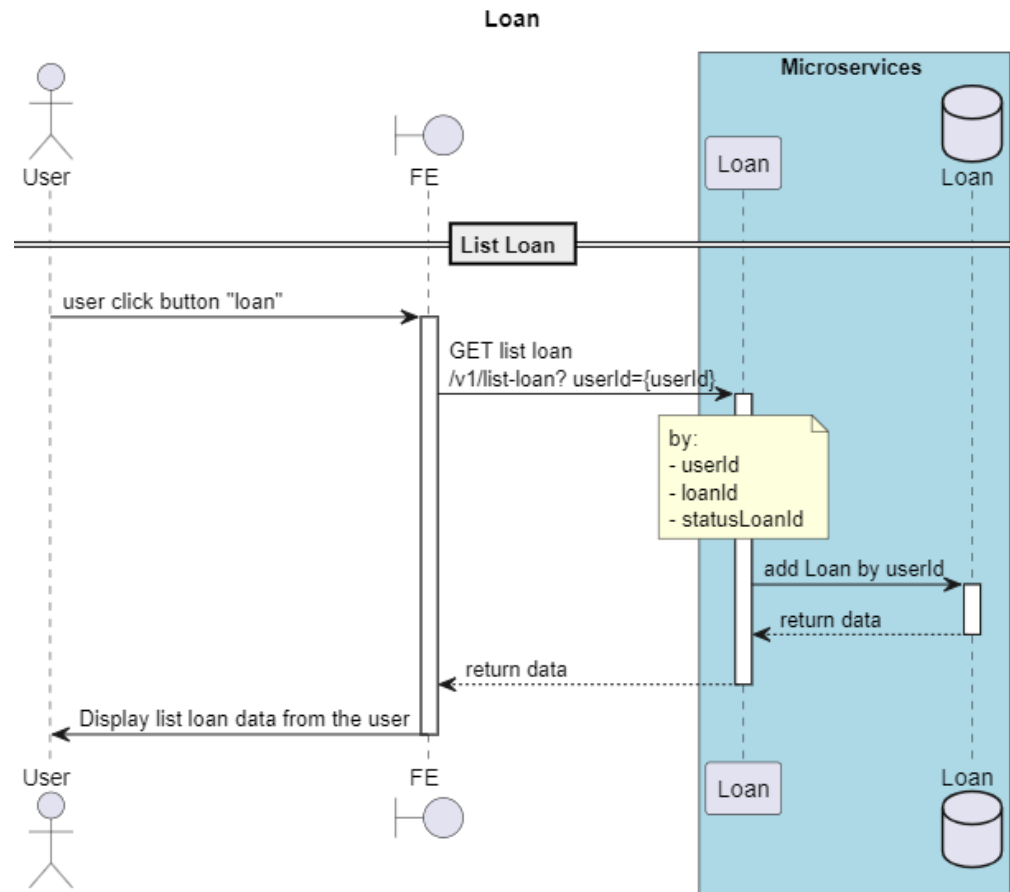
### API Contract

Method	POST
Endpoint	{{host}}ms-members/v1/members/check
Request	Header X-localization: {{LANGUAGE}} Content-Type:application/json x-api-Key: {{API_KEY}} channel: {{CHANNEL}}
	Body { "email": "user@example.com", "password": "password123", "biometric": "biometric123"

	}
Response	<p><b>success</b></p> <pre>{   "status": true,   "code": "CU001",   "message": true,   "errorMessage": null,   "errorData": null,   "data": {     "message": "Registration successful",     "userId": "12345"   } }</pre> <p><b>failed</b></p> <pre>{   "status": true,   "code": "CU002",   "message": null,   "errorMessage": "Login failed",   "errorData": {     "errorTitle": "Login Failed",     "errorDesc": "Please check your email/password"   },   "data": null }</pre>

## Get List User Loans

### Sequence Diagram



### API Contract

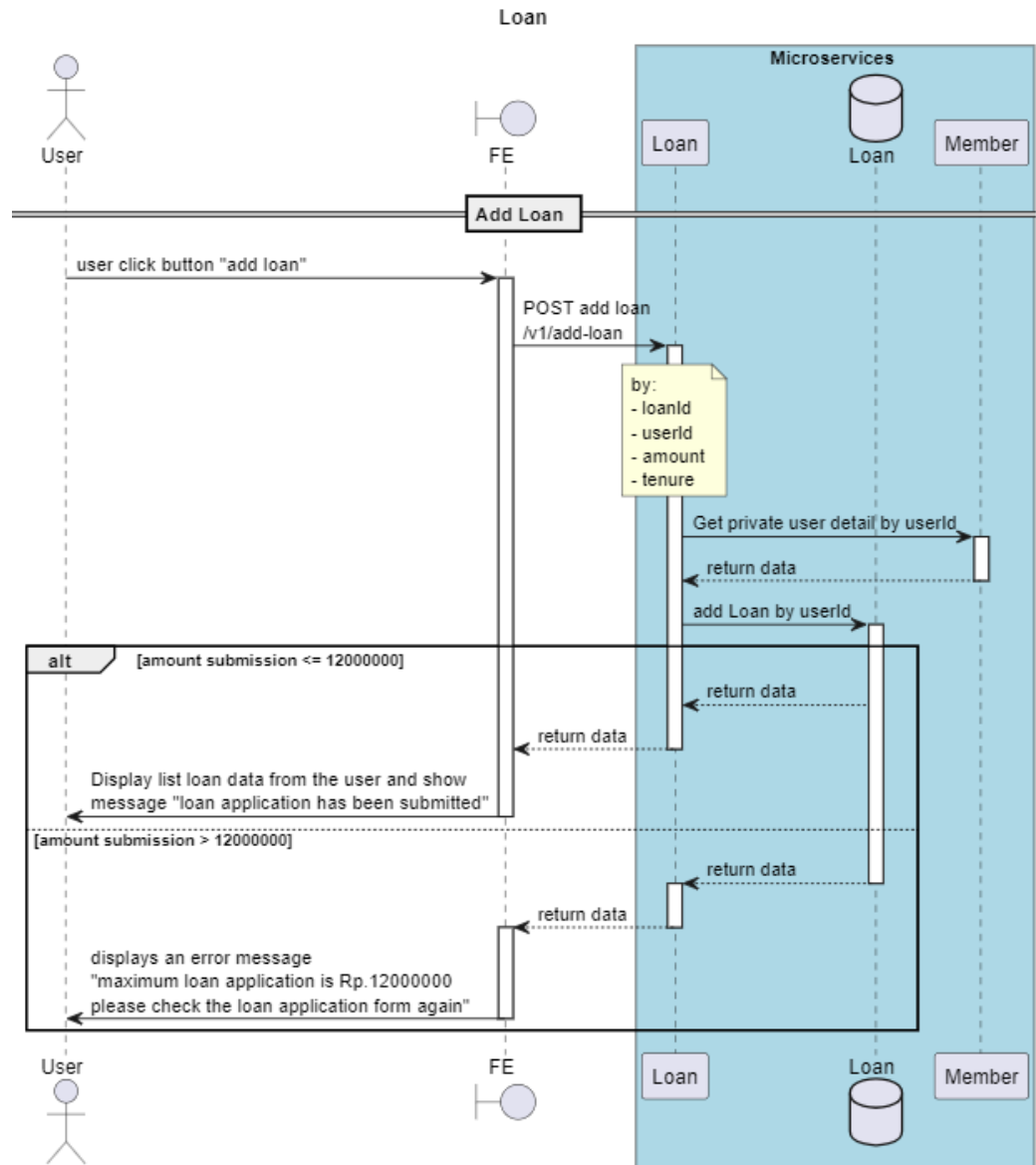
Method	GET
Endpoint	{{host}}ms-loan /v1/list-loan? userId={userId}
Request	Header X-localization: {{LANGUAGE}} Authorization: Bearer jwt-token Content-Type:application/json x-api-Key: {{API_KEY}} channel: {{CHANNEL}}
	Body -
Response	<b>success</b> { "status": true, "code": "LU001", "message": true, "errorMessage": null,

	<pre>"errorData": null, "data": {   "loans": [     {       "loanId": "1",       "amount": 5000000,       "tenure": 12,       "status": {         "statusId": "1",         "description": "approved"       },       "createdAt": "2023-06-01T00:00:00Z"     },     {       "loanId": "2",       "amount": 5000000,       "tenure": 12,       "status": {         "statusId": "2",         "description": "rejected"       },       "createdAt": "2023-06-01T00:00:00Z"     },     {       "loanId": "3",       "amount": 5000000,       "tenure": 12,       "status": {         "statusId": "3",         "description": "pending"       },       "createdAt": "2023-06-01T00:00:00Z"     }   ] }</pre>
--	---



## Add Loan

### Sequence Diagram

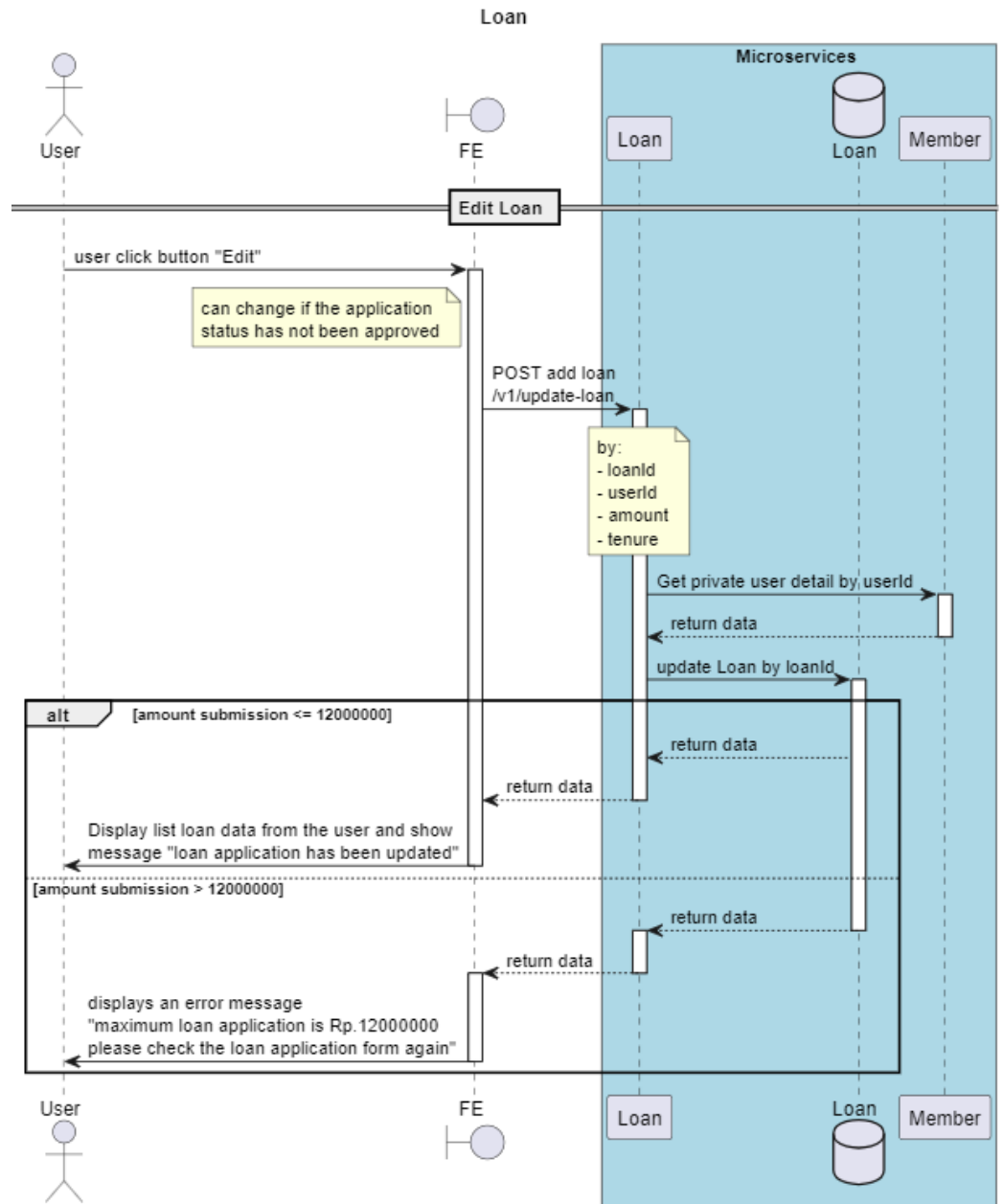


### API Contract

Method	POST
Endpoint	{{host}}ms-loan/v1/add-loan
Request	<div>Header</div> <div> X-localization: {{LANGUAGE}}  Authorization: Bearer jwt-token  Content-Type:application/json  x-api-Key: {{API_KEY}}  channel: {{CHANNEL}}h </div> <div>Body</div>

	<pre>{   "member":{     "userId":1,     "nama": "sukijan",     "email": "user@example.com",     "phoneNumber": "08123456789",     "password": "password123",     "photo": "base64-encoded-image",     "ktp": "base64-encoded-image"   },   "loanId": 1,   "amount": 10000000,   "tenure": 12 }</pre>
Response	<pre><b>success</b> {   "status": true,   "code": "AL001",   "message": true,   "errorMessage": null,   "errorData": null,   "data": {     "message": "Loan application submitted",     "loanId": "2"   } }  <b>failed</b> {   "status": true,   "code": "LU002",   "message": null,   "errorMessage": "Loan application not submitted",   "errorData": {     "errorTitle": "Loan application not submitted",     "errorDesc": "Please check your loan application"   },   "data": null }</pre>

## Update Loan Sequence Diagram



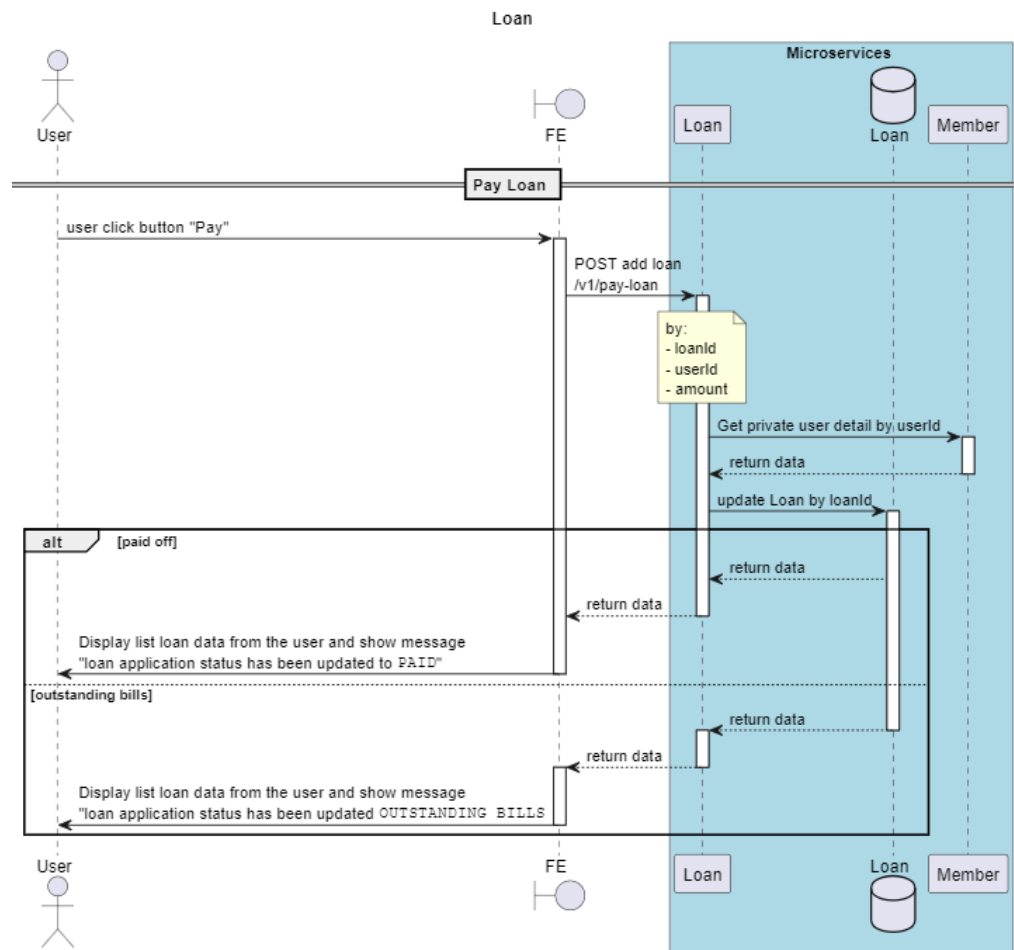
### API Contract

Method	POST
Endpoint	{{host}}ms-loan/v1/update-loan
Request	Header X-localization: {{LANGUAGE}} Authorization: Bearer jwt-token Content-Type: application/json x-api-Key: {{API_KEY}}

	<p>channel: {{CHANNEL}}</p> <pre> Body {   "member":{     "userId":1,     "nama": "sukijan",     "email": "user@example.com",     "phoneNumber": "08123456789",     "password": "password123",     "photo": "base64-encoded-image",     "ktp": "base64-encoded-image"   },   "loanId": 1,   "amount": 10000000,   "tenure": 12 } </pre>
Response	<pre> <b>success</b> {   "status": true,   "code": "AL001",   "message": true,   "errorMessage": null,   "errorData": null,   "data": {     "message": "Loan application submitted",     "loanId": "2"   } }  <b>failed</b> {   "status": true,   "code": "LU002",   "message": null,   "errorMessage": "Loan application not submitted",   "errorData": {     "errorTitle": "Loan application not submitted",     "errorDesc": "Please check your loan application"   },   "data": null } </pre>

## Pay Loan

### Sequence Diagram



### API Contract

Method	POST
Endpoint	{{host}}ms-loan/v1/pay-loan
Request	<div>Header</div> <div>X-localization: {{LANGUAGE}}</div> <div>Authorization: Bearer jwt-token</div> <div>Content-Type: application/json</div> <div>x-api-Key: {{API_KEY}}</div> <div>channel: {{CHANNEL}}</div> <div>Body</div> <div>{</div> <div> <div>"member":{</div> <div> <div>"userId":1,</div> <div>"nama": "sukijan",</div> <div>"email": "user@example.com",</div> <div>"phoneNumber": "08123456789",</div> </div> </div>

	<pre>         "password": "password123",         "photo": "base64-encoded-image",         "ktp": "base64-encoded-image"       },       "loanId": 1,       "amount": 10000000,       "tenure": 12     } </pre>
Response	<pre> <b>success</b> {   "status": true,   "code": "PL001",   "message": true,   "errorMessage": null,   "errorData": null,   "data": {     "message": "Loan application paid, remaining loan amount 1000000 with tenure 10 months",     "loanId": "2"   } }  <b>failed</b> {   "status": true,   "code": "PU002",   "message": null,   "errorMessage": "Loan application not paid",   "errorData": {     "errorTitle": "Loan application not paid",     "errorDesc": "Please check your loan application"   },   "data": null } </pre>

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

#### Detail design

No.	Name Screen	Behavior
1.	Welcome Screen	<ul style="list-style-type: none"> <li>• Display buttons for "Register" and "Login".</li> <li>• Navigate to Registration Screen or Login Screen based on user selection.</li> </ul>

2.	Registration Screen	<ul style="list-style-type: none"> <li>• Collect user data (email, phone number, password, photo, KTP).</li> <li>• Validate inputs.</li> <li>• On submission, call User Registration API.</li> <li>• Display success or error message based on API response.</li> </ul>
3.	Login Screen	<ul style="list-style-type: none"> <li>• Allow login via password or biometric.</li> <li>• On submission, call User Login API.</li> <li>• Store JWT token on successful login.</li> <li>• Navigate to Dashboard.</li> </ul>
4.	Dashboard	<ul style="list-style-type: none"> <li>• Fetch user loans using Get User Loans API.</li> <li>• Display outstanding loans and monthly bills.</li> <li>• Provide option to apply for a new loan.</li> <li>• Navigate to Loan Application Screen on selection.</li> </ul>
5.	Loan Application Screen	<ul style="list-style-type: none"> <li>• Collect loan amount and tenure from the user.</li> <li>• Validate inputs (e.g., maximum amount, maximum tenure).</li> <li>• On submission, call Apply for Loan API.</li> <li>• Display success or error message based on API response.</li> </ul>
6.	Loan Status Screen	<ul style="list-style-type: none"> <li>• Display loan application status (approved, rejected, pending).</li> <li>• Fetch loan status using Get Loan Status API.</li> </ul>
7.	Notifications	<ul style="list-style-type: none"> <li>• Display notifications for loan approval/rejection.</li> <li>• Fetch notifications from the server and display them to the user.</li> </ul>