*Software Requirements Specification*

for

AppsMarket

Version 1.0 approved

Prepared by

20220801051 – Filbert Chiuman

20220801267 - Vincent Etwin Mangapul

31-03-2024

Table of Contents

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

1.2 Audien yang Dituju dan Pembaca yang Disarankan

1.3 Batasan Produk

1.4 Definisi dan Istilah

1.5 Refrensi

2. Deskripsi Keseluruhan

2.1 Deskripsi Produk

2.2 Fungsi Produk

2.3 Penggolongan Karakterik Pengguna

2.4 Lingkungan Operasi

2.5 Batasan Desain dan Implementasi

2.6 Dokumentasi Pengguna

3. Kebutuhan Antarmuka Eksternal

3.1 User Interfaces

3.2 Hardware Interface

3.3 Software Interface

3.4 Communication Interface

4. Functional Requirement

4.1 Use Case Diagram

4.2 Activity Diagram

4.3 Class Diagram

4.4 Communication Diagram

4.5 State Machine Diagram

4.6 Sequence Diagram

4.7 Component Diagram

4.8 Deployment Diagram

4.9 Composite Diagram

4.10 Object Diagram

5. Non Functional Requirements

**6. Mock Up…………………………………………………………………………………….**

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SRS) untuk AppsMarket bertujuan untuk memberikan gambaran komprehensif tentang fungsionalitas, batasan, dan interaksi pengguna yang diharapkan dari aplikasi ini. Dokumen ini bertujuan untuk:

* Menjabarkan secara detail kebutuhan fungsional dan non-fungsional aplikasi AppsMarket.
* Menyediakan panduan yang jelas bagi pengembang dan desainer untuk membangun dan mendesain aplikasi sesuai dengan kebutuhan pengguna dan harapan pasar.
* Mengidentifikasi batasan dan asumsi yang terkait dengan pengembangan aplikasi.
* Melayani sebagai perjanjian formal antara pemangku kepentingan tentang fitur dan fungsi yang akan dihadirkan oleh aplikasi.

## Audien yang Dituju dan Pembaca yang Disarankan

Dokumen ini ditujukan untuk berbagai audiens, termasuk:

* Tim Pengembangan: Untuk memahami secara mendalam apa yang perlu dibangun, termasuk arsitektur sistem, teknologi yang digunakan, dan detail implementasi.
* Desainer UI/UX: Agar dapat merancang antarmuka pengguna yang intuitif dan memenuhi kebutuhan pengguna.
* Tim QA dan Pengujian: Untuk mengetahui aspek apa saja yang perlu diuji, termasuk fungsionalitas, keamanan, dan kinerja aplikasi.
* Pemangku Kepentingan: Termasuk investor, mitra, dan manajemen, untuk memahami visi produk dan ekspektasi pasar.
* Pengguna Akhir (sebagai informasi): Untuk mendapatkan pemahaman tentang apa yang dapat mereka harapkan dari aplikasi, meskipun mereka bukan pembaca utama dokumen ini.

## Batasan Produk

Batasan produk untuk AppsMarket mencakup:

* Platform dan Teknologi: AppsMarket akan dikembangkan untuk sistem operasi iOS dan Android, menggunakan bahasa pemrograman Swift dan Kotlin.
* Integrasi Pihak Ketiga: Aplikasi akan mengintegrasikan API untuk metode pembayaran, analisis pengguna, dan layanan iklan.
* Jangkauan Geografis: Awalnya, AppsMarket akan tersedia hanya dalam beberapa bahasa dan negara tertentu, dengan rencana ekspansi berdasarkan respons pasar.
* Regulasi dan Kepatuhan: AppsMarket akan mematuhi GDPR, CCPA, dan regulasi lokal lainnya terkait privasi data pengguna.

## Definisi dan Istilah

* SRS (Software Requirements Specification): Dokumen yang mencakup semua kebutuhan dan spesifikasi untuk pengembangan perangkat lunak.
* UI/UX: User Interface dan User Experience, yang berkaitan dengan desain visual dan interaksi pengguna dengan aplikasi.
* API (Application Programming Interface): Antarmuka pemrograman yang memungkinkan perangkat lunak berinteraksi dengan perangkat lunak lain.
* GDPR (General Data Protection Regulation): Regulasi perlindungan data dan privasi untuk semua individu dalam Uni Eropa.
* CCPA (California Consumer Privacy Act): Undang-undang negara bagian California yang bertujuan untuk meningkatkan hak privasi dan perlindungan konsumen bagi penduduk California.

## Refrensi

1.) General Data Protection Regulation (GDPR):

Situs Resmi Uni Eropa tentang GDPR. Memberikan informasi lengkap mengenai regulasi, panduan untuk organisasi, dan hak-hak individu.

<https://gdpr.eu/>

2.)California Consumer Privacy Act (CCPA):

Situs resmi negara bagian California yang menyediakan informasi tentang CCPA, termasuk teks undang-undang, panduan untuk bisnis, dan hak-hak konsumen.

<https://oag.ca.gov/privacy/ccpa>

3.)Prinsip Dasar UI/UX Design:

Artikel oleh Nielsen Norman Group, sebuah otoritas terkemuka dalam penelitian UX, yang menjelaskan prinsip-prinsip dasar desain UI/UX.

<https://www.nngroup.com/articles/ten-usability-heuristics/>

4.) Panduan Pengembangan Android:

Dokumentasi resmi untuk pengembang Android oleh Google. Berisi panduan, tutorial, dan referensi API untuk pengembangan aplikasi Android.

<https://developer.android.com/docs>

5.) Panduan Pengembangan iOS:

Dokumentasi resmi Apple untuk pengembang iOS. Menyediakan sumber daya, panduan, dan tutorial untuk membuat aplikasi untuk perangkat iOS.

<https://developer.apple.com/documentation/>

# Deskripsi Keseluruhan

## Deskripsi Produk

AppsMarket adalah platform distribusi digital yang memungkinkan pengguna untuk menjelajahi, mengunduh, dan mengelola aplikasi untuk perangkat mobile mereka. Didesain dengan antarmuka pengguna yang intuitif, AppsMarket menyediakan akses mudah ke berbagai aplikasi, baik gratis maupun berbayar, yang meliputi kategori seperti permainan, produktivitas, pendidikan, gaya hidup, dan banyak lagi. Platform ini juga menawarkan fitur rekomendasi personalisasi, ulasan pengguna, dan pembaruan aplikasi otomatis, bertujuan untuk menciptakan pengalaman pengguna yang kaya dan memuaskan.

## Fungsi Produk

* Penjelajahan dan Pencarian: Pengguna dapat menjelajahi aplikasi berdasarkan kategori, peringkat, atau melalui pencarian langsung menggunakan kata kunci.
* Instalasi dan Pembaruan Aplikasi: Memungkinkan pengguna untuk menginstal aplikasi baru dan memperbarui yang sudah ada secara otomatis atau manual.
* Ulasan dan Rating: Pengguna dapat memberikan rating dan ulasan terhadap aplikasi yang telah digunakan untuk membantu pengguna lain dalam membuat keputusan.
* Manajemen Aplikasi: Fasilitas untuk melihat aplikasi yang telah diinstal, mengelola pembaruan, dan menghapus aplikasi jika diperlukan.
* Rekomendasi Personalisasi: Fitur yang menyediakan rekomendasi aplikasi berdasarkan preferensi dan riwayat penggunaan aplikasi pengguna.

## Penggolongan Karakterik Pengguna

AppsMarket ditujukan untuk berbagai jenis pengguna yang masing-masing memiliki peran dan kebutuhan yang berbeda. Pengguna dapat digolongkan menjadi tiga kategori utama:

(1.) Pengguna (User):

Individu yang menggunakan aplikasi untuk mencari, mengunduh, menginstal, dan mengelola aplikasi mobile.

Karakteristik:

- Mencari aplikasi yang memenuhi kebutuhan spesifik mereka (hiburan, produktivitas, pendidikan, dll.).

- Membutuhkan antarmuka pengguna yang sederhana dan intuitif.

- Mengandalkan ulasan dan rating untuk memutuskan aplikasi mana yang akan diunduh.

- Menginginkan pembaruan aplikasi yang mudah dan otomatis.

- Memiliki berbagai tingkat pengetahuan teknis, dari pemula hingga mahir.

(2.) Pengembang Aplikasi (Developer/Apps Company):

Individu atau perusahaan yang membuat dan mengelola aplikasi yang tersedia di AppsMarket.

Karakteristik:

- Membutuhkan akses ke dashboard pengembang untuk mengunggah dan mengelola aplikasi mereka.

- Menginginkan analitik dan laporan kinerja aplikasi untuk memahami penggunaan dan mengidentifikasi area untuk perbaikan.

- Memerlukan mekanisme untuk merespon ulasan dan feedback dari pengguna.

- Mencari platform yang memudahkan distribusi aplikasi dan memberikan visibilitas tinggi.

- Memahami teknis dan memerlukan dokumentasi API untuk integrasi fitur khusus.

(3.) Administrator (Admin):

Tim atau individu yang bertanggung jawab untuk memelihara dan mengelola keseuruhan platform AppsMarket.

Karakteristik:

- Mengelola konten aplikasi, termasuk persetujuan, penolakan, dan penghapusan aplikasi yang tidak sesuai dengan kebijakan.

- Memantau aktivitas pengguna dan pengembang untuk memastikan kepatuhan terhadap kebijakan dan regulasi.

- Mengelola aspek teknis seperti server backend, database, dan performa sistem.

- Mengidentifikasi dan menyelesaikan masalah teknis atau operasional yang mungkin timbul.

- Membutuhkan akses ke alat analisis dan pelaporan untuk memantau kinerja platform secara keseluruhan.

## Lingkungan Operasi

AppsMarket akan beroperasi pada:

* Perangkat Mobile: Dukungan untuk sistem operasi iOS dan Android.
* Server Backend: Infrastruktur cloud untuk hosting database aplikasi, manajemen pengguna, analitik, dan layanan backend lainnya.
* Koneksi Internet: Diperlukan untuk mengakses, mengunduh, dan memperbarui aplikasi.

## Batasan Desain dan Implementasi

* Ketergantungan pada Sistem Operasi: Fungsionalitas tertentu mungkin terbatas oleh kemampuan sistem operasi.
* Kepatuhan Regulasi: Harus mematuhi regulasi setempat dan internasional terkait privasi data dan transaksi digital.
* Optimisasi Kinerja: Didesain untuk minimalkan penggunaan data dan baterai tanpa mengorbankan pengalaman pengguna.

## Dokumentasi Pengguna

* Panduan Pengguna: Menyediakan instruksi langkah demi langkah tentang cara menggunakan fitur utama AppsMarket.
* FAQ dan Pusat Bantuan: Seksi untuk menjawab pertanyaan umum dan menyediakan solusi untuk masalah yang sering dihadapi pengguna.

# Kebutuhan Antarmuka Eksternal

## User Interfaces

Deskripsi Antarmuka Pengguna

(1.) Karakteristik Logis:

- Tata Letak Layar: Setiap layar dalam AppsMarket harus mengikuti panduan desain UI yang konsisten. Hal ini termasuk posisi header di bagian atas, menu navigasi di bagian bawah, dan area konten di tengah.

- Standar GUI: AppsMarket akan mematuhi standar desain Material Design untuk Android dan Human Interface Guidelines untuk iOS. Ini termasuk penggunaan ikon, tipografi, dan warna yang sesuai dengan masing-masing platform.

- Fungsi Standar: Setiap layar akan memiliki tombol standar seperti “Help”, “Back”, “Home”, dan “Search” yang mudah diakses. Tombol ini akan ditempatkan di lokasi yang konsisten di seluruh aplikasi.

- Pesan Kesalahan: Semua pesan kesalahan harus informatif dan ramah pengguna. Pesan harus mencantumkan apa yang salah dan langkah-langkah yang dapat diambil pengguna untuk memperbaiki masalah.

(2.) Komponen UI yang Dibutuhkan:

- Layar Beranda: Menampilkan kategori aplikasi, aplikasi populer, dan rekomendasi.

- Layar Detail Aplikasi: Menampilkan deskripsi aplikasi, ulasan pengguna, dan tombol untuk mengunduh/instal.

- Layar Pencarian: Fitur pencarian dengan saran otomatis dan filter pencarian.

- Layar Profil Pengguna: Informasi akun pengguna, aplikasi yang diinstal, dan preferensi pengguna.

- Dashboard Pengembang: Akses untuk mengunggah dan mengelola aplikasi, melihat analitik, dan merespon ulasan.

(3.)Contoh Tampilan Layar:

Beranda:

- Header: Logo AppsMarket, tombol pencarian, ikon profil pengguna.

- Konten Utama: Kategori aplikasi, daftar aplikasi populer, rekomendasi personalisasi.

- Navigasi: Menu bawah dengan ikon untuk Beranda, Kategori, Pencarian, dan Profil.

(4.) Detail Aplikasi:

- Header: Nama aplikasi, tombol kembali.

- Konten Utama: Gambar aplikasi, deskripsi, ulasan pengguna, tombol unduh/instal.

- Footer: Informasi pengembang, versi aplikasi, dan izin yang diminta.

## Hardware Interface

Deskripsi Antarmuka Perangkat Keras:

Karakteristik Logis dan Fisik:

- Perangkat yang Didukung: AppsMarket akan mendukung smartphone dan tablet dengan sistem operasi Android (versi 6.0 ke atas) dan iOS (versi 11.0 ke atas).

- Interaksi Data dan Kontrol: Aplikasi akan menggunakan sensor perangkat (misalnya, GPS, akselerometer) sesuai kebutuhan fitur tertentu, seperti rekomendasi berbasis lokasi.

- Protokol Komunikasi: AppsMarket akan berkomunikasi dengan perangkat keras melalui API standar yang disediakan oleh Android dan iOS. Ini termasuk komunikasi dengan penyimpanan internal, kamera, dan layanan lokasi.

## Software Interface

(1.) Koneksi dengan Komponen Perangkat Lunak Lainnya:

- Sistem Operasi: Android (versi 6.0 ke atas), iOS (versi 11.0 ke atas).

- Database: AppsMarket akan menggunakan Firebase untuk autentikasi pengguna dan penyimpanan data aplikasi.

- APIs: Integrasi dengan API pihak ketiga untuk pembayaran (misalnya, Stripe, PayPal), analitik (misalnya, Google Analytics, Firebase Analytics), dan layanan iklan (misalnya, AdMob).

(2.) Pesan dan Data:

- Masuk: Data pengguna (informasi login, preferensi), permintaan pencarian aplikasi.

- Keluar: Hasil pencarian, rekomendasi aplikasi, pemberitahuan pembaruan aplikasi.

(3.) Layanan yang Dibutuhkan:

- Autentikasi Pengguna: API Firebase untuk login dan manajemen akun.

- Pembayaran: API pembayaran untuk transaksi in-app.

- Analitik: API untuk pelacakan penggunaan aplikasi dan kinerja.

- Mekanisme Berbagi Data: Data pengguna dan aplikasi akan disimpan di basis data cloud dan diakses melalui API REST.

## Communication Interface

Fungsi Komunikasi:

- Protokol Komunikasi: AppsMarket akan menggunakan HTTPS untuk komunikasi aman antara aplikasi dan server backend.

- Standar Komunikasi: Akan mengikuti standar komunikasi seperti RESTful API untuk interaksi dengan server backend.

- Keamanan dan Enkripsi: Semua data yang dikirimkan antara klien dan server akan dienkripsi menggunakan SSL/TLS untuk memastikan privasi dan integritas data.

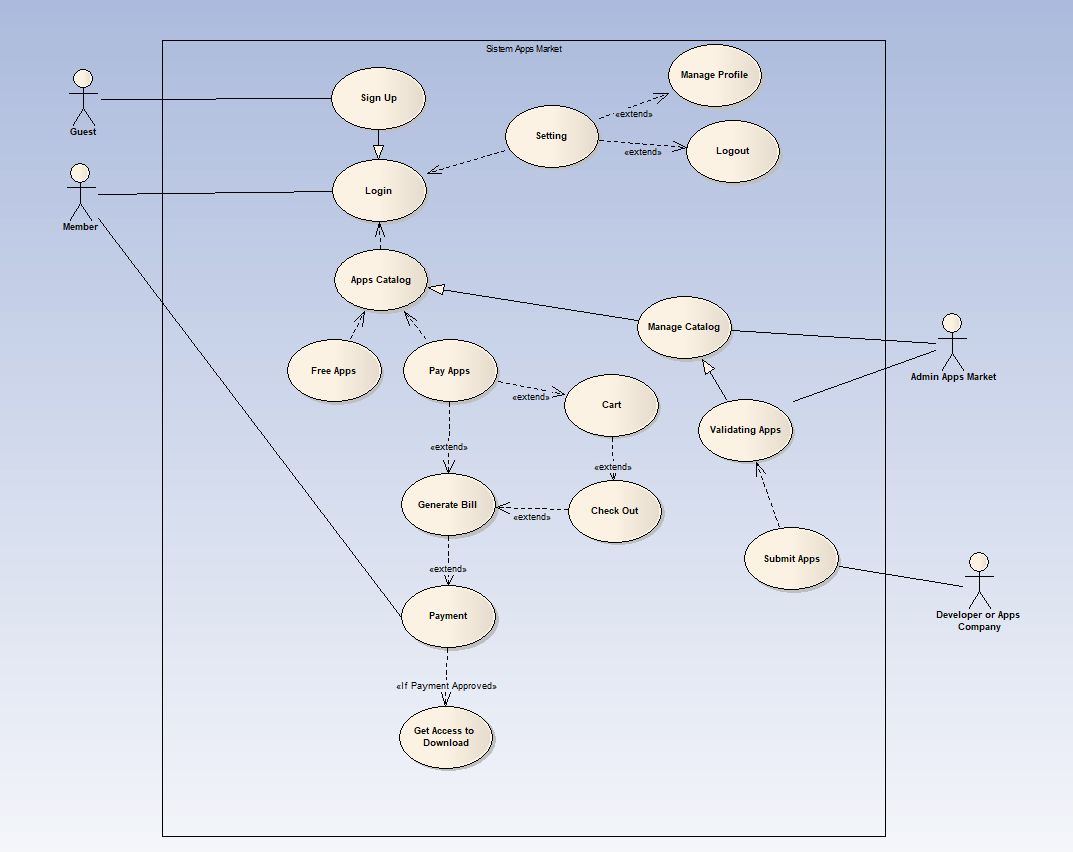
- Kecepatan Transfer Data: Koneksi akan dioptimalkan untuk mendukung transfer data yang cepat dan efisien, dengan target latensi minimal.

- Mekanisme Sinkronisasi: Aplikasi akan menggunakan mekanisme sinkronisasi real-time untuk data pengguna dan pembaruan aplikasi, memastikan bahwa informasi terbaru selalu tersedia bagi pengguna.

# Functional Requirement

| **ID** | **Kebutuhan Fungsional** | **Penjelasan** |
| --- | --- | --- |
| 1 | User Registration | Pengguna harus dapat mendaftar ke aplikasi dengan menggunakan email atau akun media sosial. |
| 2 | User Login | Pengguna harus dapat login ke aplikasi menggunakan email dan kata sandi atau akun media sosial. |
| 3 | Browse and Search Apps | Pengguna harus dapat mencari dan menjelajahi aplikasi berdasarkan kategori, popularitas, dan kata kunci. |
| 4 | App Download and Installation | Pengguna harus dapat mengunduh dan menginstal aplikasi dari AppsMarket ke perangkat mereka. |
| 5 | App Ratings and Reviews | Pengguna harus dapat memberikan rating dan ulasan untuk aplikasi yang telah mereka unduh. |
| 6 | Developer Account Management | Pengembang harus dapat mendaftar, login, dan mengelola akun mereka untuk mengunggah aplikasi. |
| 7 | App Upload and Management | Pengembang harus dapat mengunggah aplikasi, memperbarui deskripsi, dan melacak statistik unduhan. |
| 8 | Admin Dashboard | Admin harus memiliki akses ke dasbor untuk mengelola pengguna, aplikasi, dan ulasan. |
| 9 | Notification System | Pengguna harus menerima notifikasi tentang pembaruan aplikasi, penawaran khusus, dan berita terbaru. |
| 10 | Payment Integration | Pengguna harus dapat melakukan pembelian dalam aplikasi menggunakan berbagai metode pembayaran. |
| 11 | User Profile Management | Pengguna harus dapat mengelola profil mereka, termasuk mengubah informasi pribadi dan pengaturan keamanan. |
| 12 | App Recommendations | Aplikasi harus menyediakan rekomendasi aplikasi berdasarkan preferensi dan riwayat unduhan pengguna. |
| 13 | Reporting and Analytics | Admin dan pengembang harus memiliki akses ke laporan dan analitik mengenai performa aplikasi dan interaksi pengguna. |
| 14 | Customer Support | Pengguna harus dapat menghubungi dukungan pelanggan melalui aplikasi untuk melaporkan masalah atau mendapatkan bantuan. |
| 15 | Content Moderation | Admin harus dapat memoderasi konten, termasuk ulasan pengguna dan aplikasi yang diunggah |
| 16 | Localization Support | Aplikasi harus mendukung berbagai bahasa dan mata uang untuk pengguna internasional. |
| 17 | Data Backup and Recovery | Sistem harus secara otomatis mencadangkan data pengguna dan aplikasi serta menyediakan mekanisme pemulihan data. |
| 18 | Privacy Controls | Pengguna harus memiliki kontrol atas data pribadi mereka, termasuk opsi untuk menghapus akun dan data mereka. |
| 19 | Performance Monitoring | Sistem harus memantau kinerja aplikasi dan melaporkan masalah seperti crash atau lambatnya respon. |
| 20 | Compliance and Regulations | Aplikasi harus mematuhi semua peraturan dan regulasi yang berlaku, termasuk perlindungan data dan kebijakan privasi. |

## Use Case Diagram



Untuk membuat diagram kasus penggunaan (Use Case) untuk aplikasi "Apps Market," kami mengikuti proses yang terstruktur:

1. Wawancara Klien

Kami melakukan wawancara mendetail dengan klien untuk memahami berbagai fungsi dan interaksi pengguna yang mereka inginkan dalam aplikasi.

2. Identifikasi Aktor

Kami mengidentifikasi aktor utama yang terlibat dalam sistem:

- Guest (Tamu): Pengguna yang belum terdaftar atau masuk.

- Member (Anggota): Pengguna yang telah terdaftar dan masuk.

- Admin Apps Market: Administrator yang mengelola katalog aplikasi dan validasi.

- Developer atau Apps Company: Pengembang atau perusahaan yang mengirimkan aplikasi ke Apps Market.

3. Tentukan Kasus Penggunaan

Kami mendefinisikan kasus penggunaan utama untuk setiap aktor berdasarkan fungsi dan interaksi yang dibahas selama wawancara dengan klien. Kasus penggunaan ini mencakup:

- Guest

- Sign Up (Mendaftar)

- Login (Masuk)

- Member

- Login (Masuk)

- Apps Catalog (Katalog Aplikasi)

- Free Apps (Aplikasi Gratis)

- Pay Apps (Aplikasi Berbayar)

- Cart (Keranjang)

- Check Out (Checkout)

- Payment (Pembayaran)

- Get Access to Download (Dapatkan Akses untuk Mengunduh)

- Setting (Pengaturan)

- Manage Profile (Kelola Profil)

- Logout (Keluar)

- Admin Apps Market

- Manage Catalog (Kelola Katalog)

- Validating Apps (Validasi Aplikasi)

- Developer atau Apps Company

- Submit Apps (Kirim Aplikasi)

4. Memetakan Hubungan

Kami memetakan hubungan antara aktor dan kasus penggunaan, memastikan bahwa semua interaksi dan ketergantungan didefinisikan dengan jelas. Berikut ini adalah deskripsi hubungan dari diagram di atas:

- Guest

- Mendaftar (Sign Up)

- Masuk (Login)

- Member

- Masuk (Login)

- Mengakses Katalog Aplikasi (Apps Catalog)

- Melihat Aplikasi Gratis (Free Apps)

- Melihat Aplikasi Berbayar (Pay Apps)

- Menggunakan Keranjang (Cart)

- Melakukan Checkout (Check Out)

- Melakukan Pembayaran (Payment)

- Mendapatkan Akses untuk Mengunduh (Get Access to Download)

- Mengatur Profil (Setting)

- Mengelola Profil (Manage Profile)

- Keluar (Logout)

- Admin Apps Market

- Mengelola Katalog (Manage Catalog)

- Memvalidasi Aplikasi (Validating Apps)

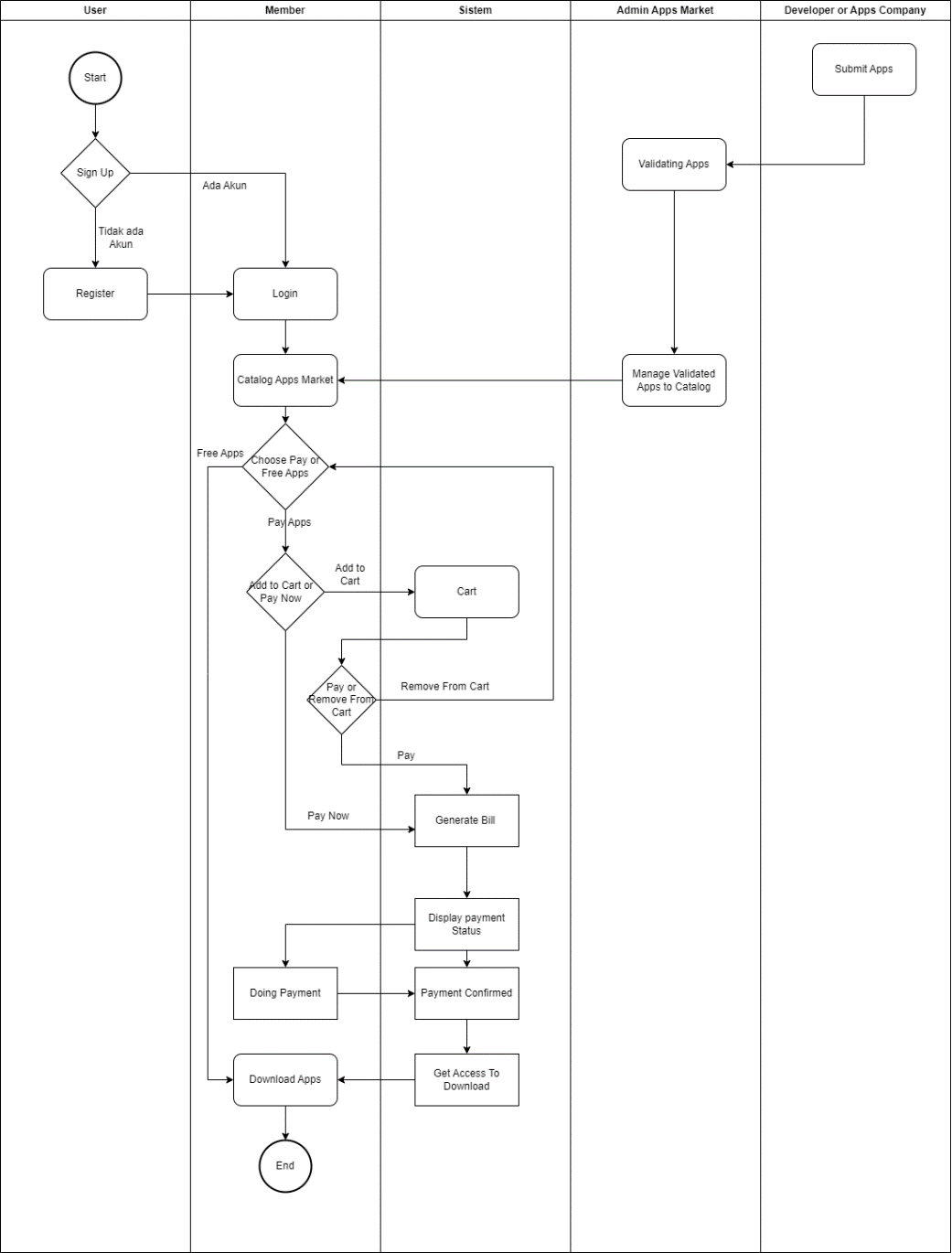
- Developer atau Apps Company

- Mengirim Aplikasi (Submit Apps)

5. Penyempurnaan Iteratif

Melalui tinjauan dan penyempurnaan berulang, kami memastikan bahwa use case diagram secara akurat mewakili fungsi sistem dan interaksi pengguna.

## Activity Diagram



Untuk membuat diagram aktivitas (activity diagram) untuk aplikasi "Apps Market," kami mengikuti proses yang terstruktur:

1. Wawancara Klien dan Pemangku Kepentingan

Kami melakukan wawancara dengan klien dan pemangku kepentingan untuk mengumpulkan persyaratan terperinci dan memahami alur kerja untuk berbagai fungsi dalam aplikasi.

2. Identifikasi Aktivitas Utama

Berdasarkan persyaratan yang telah dikumpulkan, kami mengidentifikasi aktivitas utama yang terlibat dalam setiap kasus penggunaan. Ini termasuk tindakan seperti mendaftar, masuk, mengelola katalog aplikasi, mengelola keranjang, dan mengunduh aplikasi.

3. Pembagian Aktivitas menjadi Langkah-langkah Tindakan Spesifik

Kami membagi setiap aktivitas menjadi langkah-langkah tindakan yang spesifik. Misalnya, untuk aktivitas "Pembayaran", langkah-langkahnya mungkin termasuk "Menghasilkan Tagihan", "Melakukan Pembayaran", "Menampilkan Status Pembayaran", dan "Mengonfirmasi Pembayaran".

4. Pemetaan Alur Kontrol

Kami memetakan alur kontrol di antara langkah-langkah tindakan, mengidentifikasi titik-titik keputusan dan tindakan paralel. Sebagai contoh, dalam aktivitas pembayaran, titik keputusan mungkin adalah "Pembayaran Dikonfirmasi?", yang mengarah ke tindakan selanjutnya yang berbeda berdasarkan hasilnya.

5. Penggunaan Swimlanes

Untuk memperjelas tanggung jawab dan menunjukkan aktor mana yang melakukan setiap tindakan, kami memasukkan swimlanes. Tindakan yang dilakukan oleh pengguna berada di swimlane yang berbeda dari yang dilakukan oleh sistem.

6. Peninjauan dan Penyempurnaan Iteratif

Kami meninjau dan menyempurnakan diagram aktivitas secara berulang dengan umpan balik dari klien dan pemangku kepentingan untuk memastikan keakuratan dan kelengkapannya.

Berikut adalah deskripsi aktivitas yang tergambar dalam Activity Diagram:

- User (Pengguna)

- Start

- Sign Up (Mendaftar)

- Register (Mendaftar)

- Login (Masuk)

- Member (Anggota)

- Login (Masuk)

- Catalog Apps Market (Katalog Aplikasi)

- Choose Pay or Free Apps (Pilih Aplikasi Berbayar atau Gratis)

- Free Apps (Aplikasi Gratis)

- Pay Apps (Aplikasi Berbayar)

- Add to Cart or Pay Now (Tambahkan ke Keranjang atau Bayar Sekarang)

- Pay or Remove from Cart (Bayar atau Hapus dari Keranjang)

- Pay (Bayar)

- Remove from Cart (Hapus dari Keranjang)

- Doing Payment (Melakukan Pembayaran)

- Download Apps (Mengunduh Aplikasi)

- End (Selesai)

- Sistem

- Generate Bill (Menghasilkan Tagihan)

- Display Payment Status (Menampilkan Status Pembayaran)

- Payment Confirmed (Pembayaran Dikonfirmasi)

- Get Access to Download (Dapatkan Akses untuk Mengunduh)

- Admin Apps Market

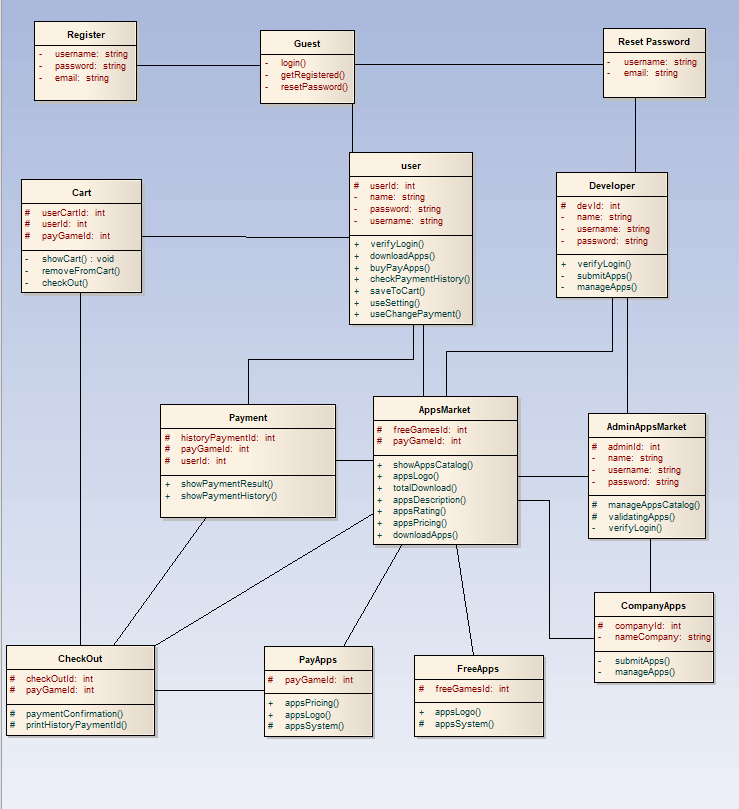
- Validating Apps (Memvalidasi Aplikasi)

- Manage Validated Apps to Catalog (Mengelola Aplikasi yang Divalidasi ke Katalog)

- Developer atau Apps Company

- Submit Apps (Mengirim Aplikasi)

## Class Diagram

****

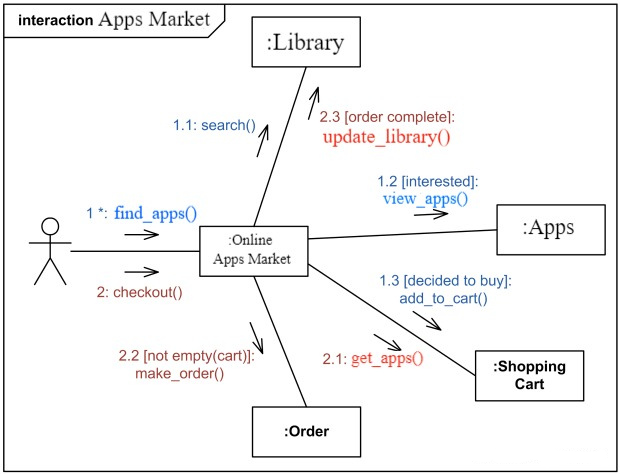
Untuk membuat class diagram aplikasi "Apps Market," kami mulai dengan wawancara terperinci dengan klien untuk mengumpulkan dan memahami kebutuhan mereka. Kami mengidentifikasi fungsi-fungsi utama seperti manajemen pengguna, penanganan aplikasi, pembayaran, dan pengelolaan katalog aplikasi.

Entitas utama yang kami tentukan meliputi Register, Guest, Reset Password, User, Developer, Admin Apps Market, Payment, Cart, CheckOut, AppsMarket, PayApps, FreeApps, dan CompanyApps. Setiap entitas memiliki atribut dan metode yang sesuai untuk mendukung fungsi aplikasi. Misalnya, entitas User memiliki atribut `userId`, `name`, `password`, dan `username`, serta metode seperti `verifyLogin()`, `downloadApps()`, dan `buyPayApps()`.

Hubungan antar entitas juga dipetakan, seperti User yang berhubungan dengan Cart, Payment, dan AppsMarket. Kami memperhitungkan peran pengguna yang berbeda dan izin mereka untuk memastikan kontrol akses yang tepat dalam sistem.

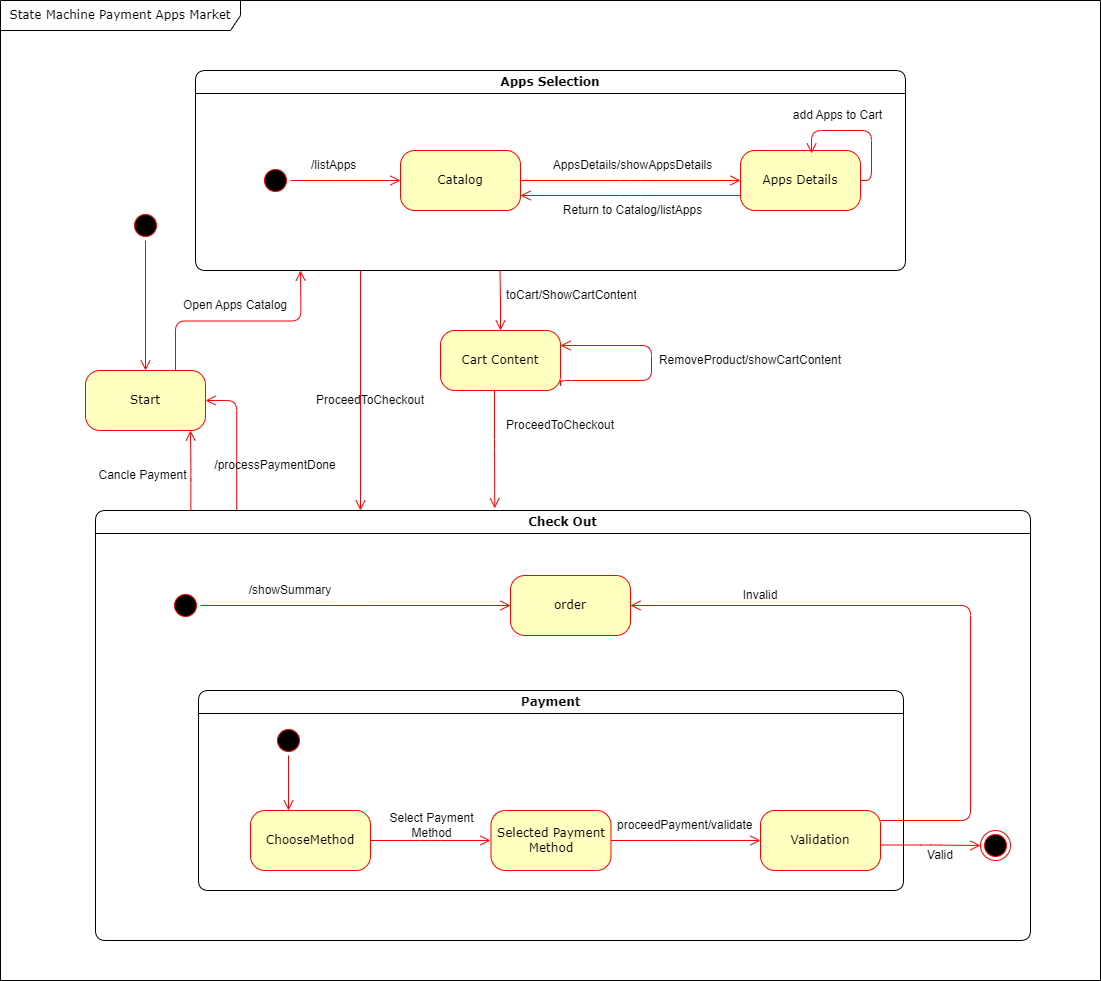
Melalui peninjauan dan penyempurnaan yang berulang, kami mengembangkan class diagram yang komprehensif yang mencerminkan struktur dan interaksi yang diperlukan untuk mendukung fitur-fitur aplikasi "Apps Market".

## Communication Diagram



UML Communication diagram "Apps Market" menunjukkan interaksi antara pengguna, aplikasi, perpustakaan, keranjang belanja, dan pesanan. Pengguna menggunakan metode `find\_apps()` untuk mencari aplikasi di "Online Apps Market". Setelah menemukan aplikasi, pengguna dapat melihat detail aplikasi menggunakan `view\_apps()`. Jika tertarik untuk membeli, aplikasi akan ditambahkan ke keranjang belanja dengan metode `add\_to\_cart()`. "Shopping Cart" mengakses daftar aplikasi dengan metode `get\_apps()`. Jika keranjang belanja tidak kosong, pengguna melanjutkan ke pembayaran dengan metode `checkout()`, kemudian membuat pesanan dengan `make\_order()`. Setelah pesanan selesai, perpustakaan diperbarui dengan metode `update\_library()`.

## State Machine Diagram

****

State Machine Diagram di atas menggambarkan proses pemilihan aplikasi dan pembayaran di "Apps Market". Berikut adalah rincian singkat dari diagram tersebut:

1. Start: Titik awal pengguna membuka katalog aplikasi.

2. Apps Selection:

- Catalog: Pengguna melihat daftar aplikasi dengan metode `listApps`.

- Apps Details: Pengguna melihat detail aplikasi yang dipilih, dapat kembali ke katalog, atau menambahkan aplikasi ke keranjang belanja.

- Cart Content: Pengguna melihat konten keranjang belanja, dapat menghapus produk, atau melanjutkan ke pembayaran dengan metode `ProceedToCheckout`.

3. Check Out:

- order: Pengguna memproses pesanan, melihat ringkasan pesanan dengan metode `showSummary`. Jika pesanan tidak valid, pengguna kembali ke keranjang belanja.

4. Payment:

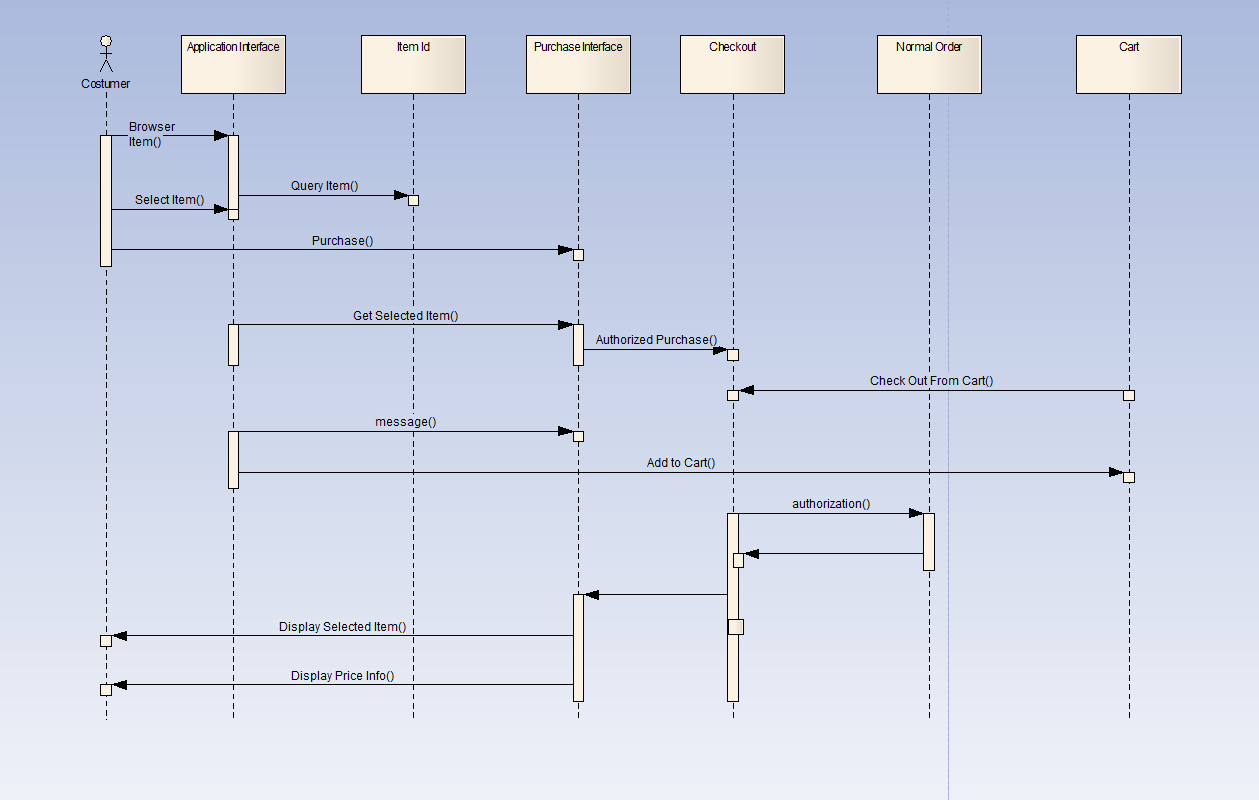
- ChooseMethod: Pengguna memilih metode pembayaran.

- Selected Payment Method: Metode pembayaran yang dipilih diproses.

- Validation: Validasi pembayaran dilakukan. Jika valid, proses pembayaran selesai.

5. End: Proses pembayaran selesai atau dibatalkan.

## Sequence Diagram

****

Sequence diagram di atas menunjukkan proses pembelian langsung atau checkout di "Apps Market". Berikut adalah rincian singkat dari diagram tersebut:

1. Customer:

- Memulai dengan `Browser Item()`, yaitu menjelajahi item yang tersedia di aplikasi.

- Memilih item dengan `Select Item()`.

2. Application Interface:

- Menjalankan `Query Item()` untuk mendapatkan informasi tentang item yang dipilih.

- Mengirim permintaan `Purchase()` ke `Item Id`.

3. Item Id:

- Mengambil item yang dipilih dan mengirimkannya kembali ke `Application Interface` dengan `Get Selected Item()`.

- Mengirim pesan `message()` ke `Purchase Interface` untuk mengotorisasi pembelian.

4. Purchase Interface:

- Melakukan otorisasi pembelian dengan `Authorized Purchase()`.

5. Checkout:

- Proses checkout dari keranjang dengan `Check Out From Cart()`.

- Menambahkan item ke keranjang dengan `Add to Cart()`.

6. Cart:

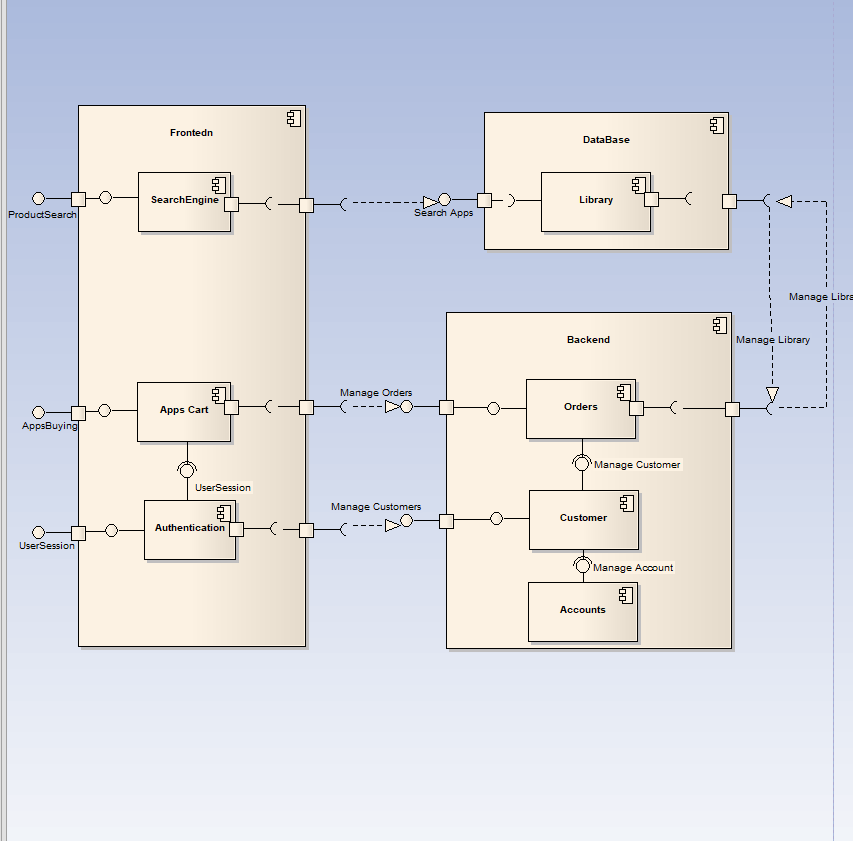
- Melakukan otorisasi dengan `authorization()`.

7. Normal Order:

- Menampilkan item yang dipilih dengan `Display Selected Item()`.

- Menampilkan informasi harga dengan `Display Price Info()`.

## Component Diagram

****

Frontend memiliki komponen SearchEngine untuk mencari aplikasi, Apps Cart untuk menampung aplikasi yang dipilih pengguna, dan Authentication untuk autentikasi pengguna. Backend terdiri dari Orders untuk mengelola pesanan, Customer untuk mengelola data pelanggan, dan Accounts untuk manajemen akun. Database memiliki komponen Library untuk menyimpan dan mengelola aplikasi. Interaksi antar komponen ditunjukkan dengan panah yang mewakili aliran data dan layanan antar komponen tersebut.

## Deployment Diagram

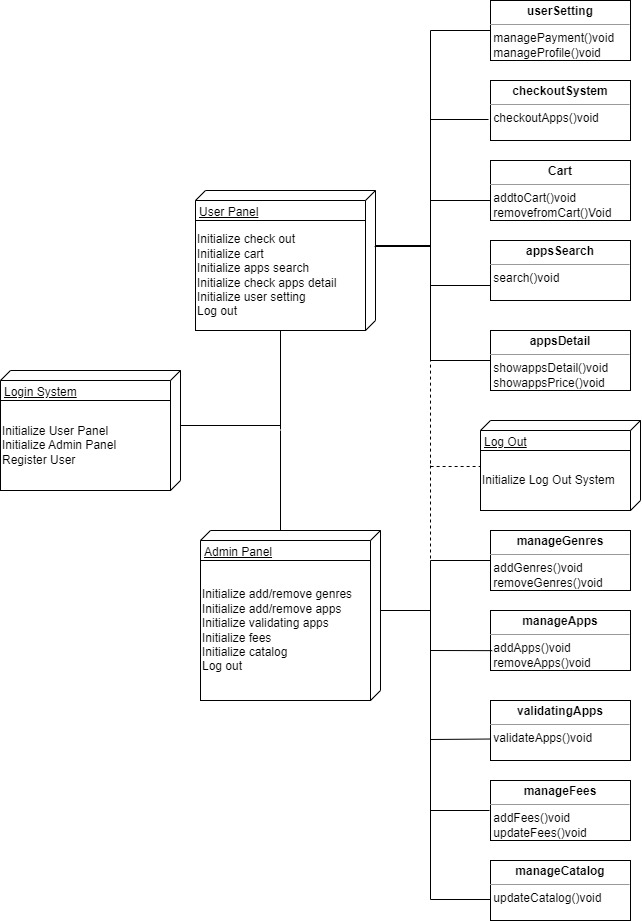
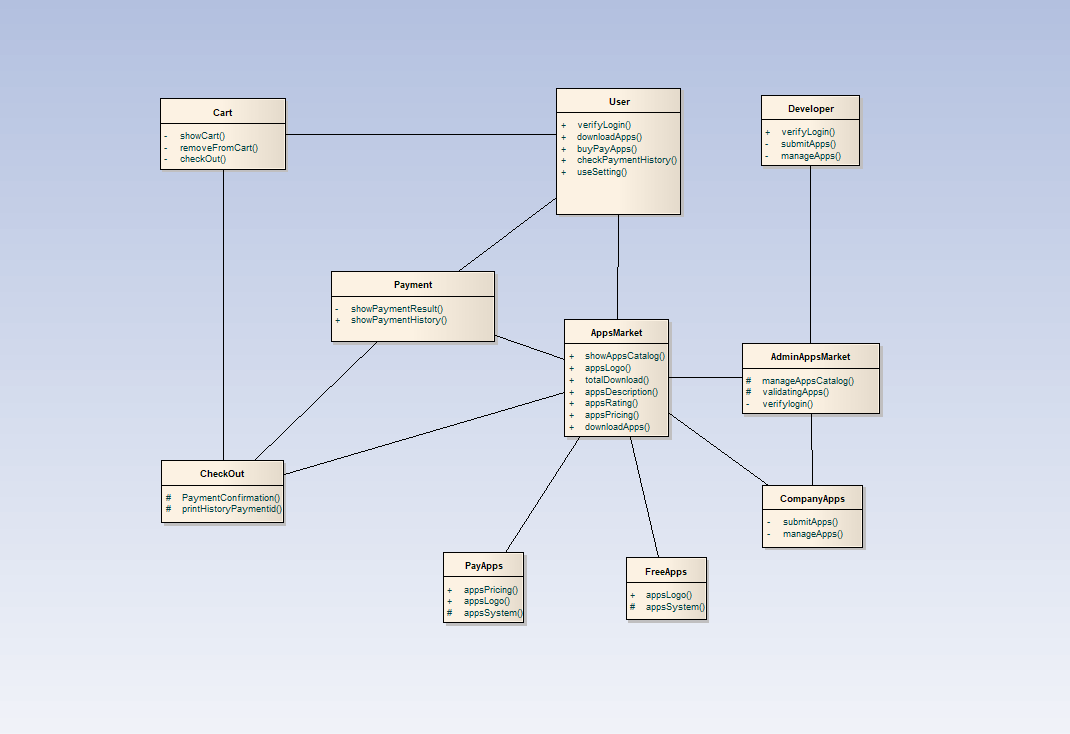


Diagram deployment "Apps Market" menunjukkan distribusi komponen pada infrastruktur. Login System menginisialisasi panel pengguna dan admin. User Panel mengelola checkout, keranjang, pencarian aplikasi, detail aplikasi, dan pengaturan pengguna. Admin Panel mengelola genre, aplikasi, validasi aplikasi, biaya, dan katalog. Log Out menginisialisasi sistem logout.

## Compostie Diagram

****

Composite diagram "Apps Market" menggambarkan hubungan antara komponen utama: User untuk login dan unduh aplikasi, Developer untuk mengelola aplikasi, AppsMarket untuk katalog dan unduhan aplikasi, Cart untuk keranjang belanja, Payment untuk hasil pembayaran, CheckOut untuk konfirmasi pembayaran, AdminAppsMarket untuk mengelola katalog, PayApps dan FreeApps untuk aplikasi berbayar dan gratis, serta CompanyApps untuk aplikasi perusahaan. Semua komponen terhubung dan saling berinteraksi untuk menyediakan layanan aplikasi ke pengguna.

## Object Diagram

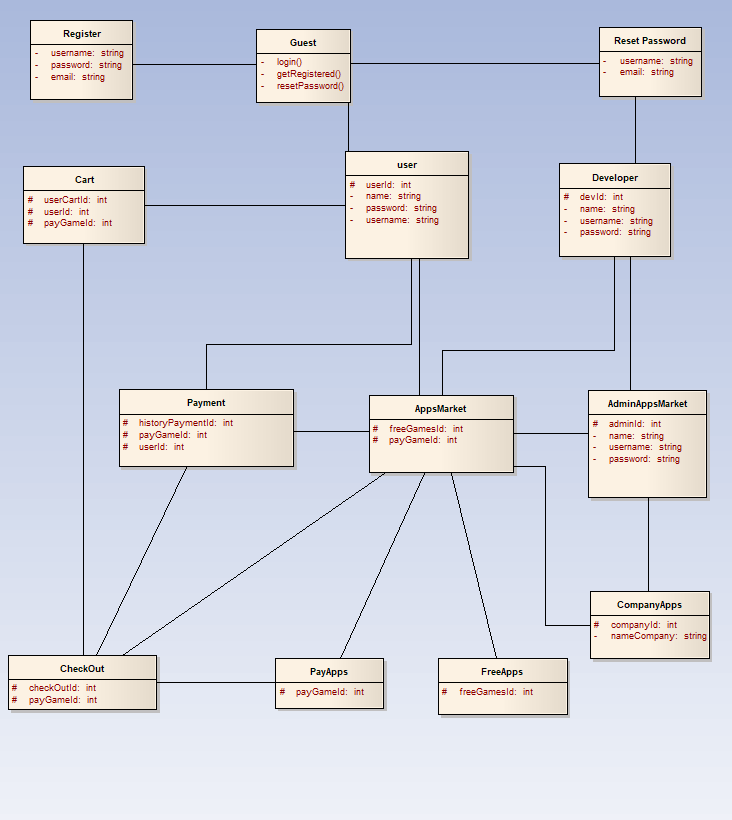
****

Diagram UML "Apps Market" menunjukkan relasi antar entitas seperti `user`, `Register`, `Guest`, `Cart`, `Payment`, `AppsMarket`, `Developer`, `AdminAppsMarket`, `CompanyApps`, `CheckOut`, `PayApps`, dan `FreeApps`. Entitas utama adalah `user` yang dapat mendaftar, login, dan melakukan pembayaran untuk aplikasi di AppsMarket. Admin dan developer juga memiliki peran untuk mengelola aplikasi.

# Non Functional Requirements

| **ID** | **Parameter** | **Kebutuhan** |
| --- | --- | --- |
| 1. | Availability | Aplikasi harus tersedia 24/7 dengan downtime maksimal 0.1% per bulan. |
| 2. | Reliability | Aplikasi harus memiliki uptime 99.9% dengan toleransi kegagalan maksimal 0.1%. |
| 3. | Ergonomy | Antarmuka pengguna harus intuitif dan mudah digunakan, dengan desain responsif untuk berbagai ukuran layer. |
| 4. | Compability | Aplikasi harus dapat berjalan di perangkat Android (versi 6.0 ke atas) dan iOS (versi 11.0 ke atas) |
| 5. | Memory | Aplikasi harus dioptimalkan agar menggunakan tidak lebih dari 200 MB memori saat dijalankan |
| 6. | Response time | Setiap layar harus memuat konten dalam waktu kurang dari 2 detik pada koneksi jaringan standar. |
| 7. | Safety | N/A |
| 8. | Security | Aplikasi harus menggunakan enkripsi SSL/TLS untuk semua komunikasi data antara klien dan server. |
| 9. | Maintainbility | Kode sumber aplikasi harus terstruktur dengan baik dan didokumentasikan agar mudah dipelihara dan diperbarui. |
| 10. | Usability | Aplikasi harus mudah digunakan, dengan antarmuka pengguna yang sederhana dan intuitif serta dokumentasi pengguna yang komprehensif. |

# Mock UP

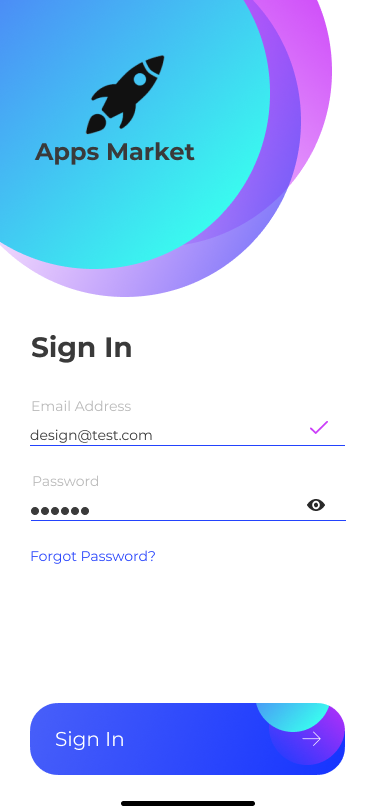
Launch Screen



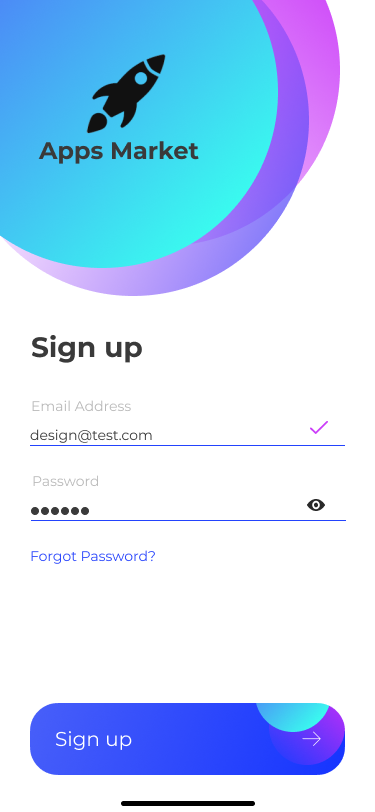
Welcome



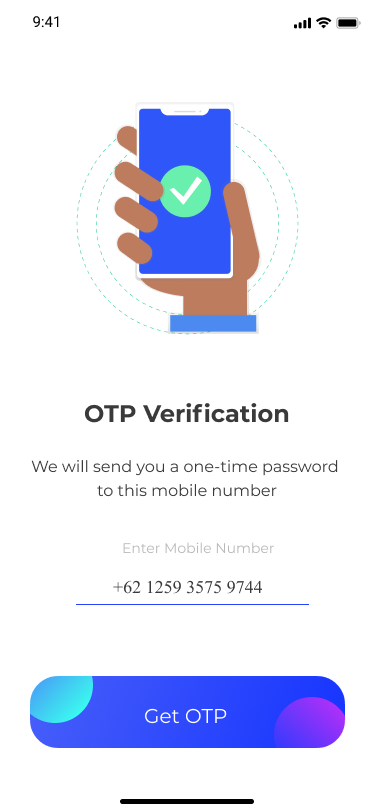
Sign In



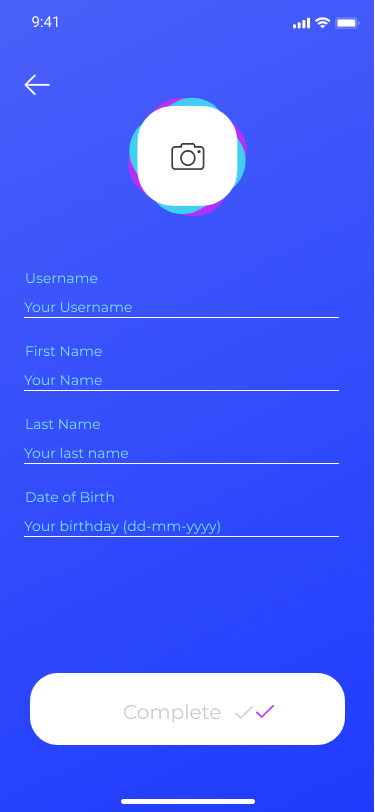
Sign Up



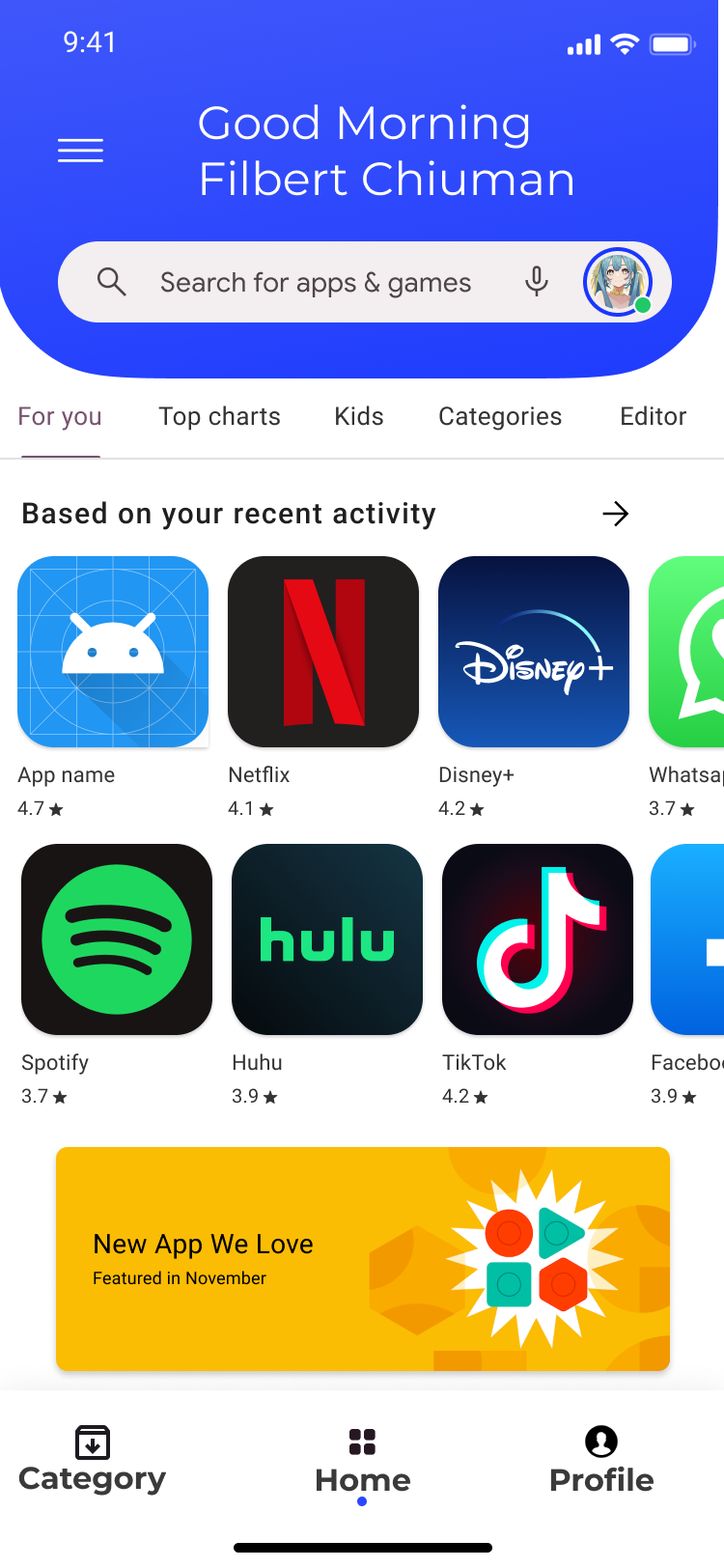
OTP when Sign Up



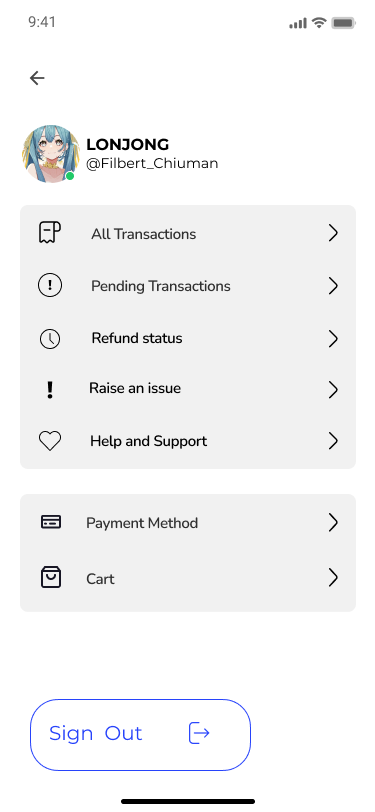
Profile Set up after Sign Up



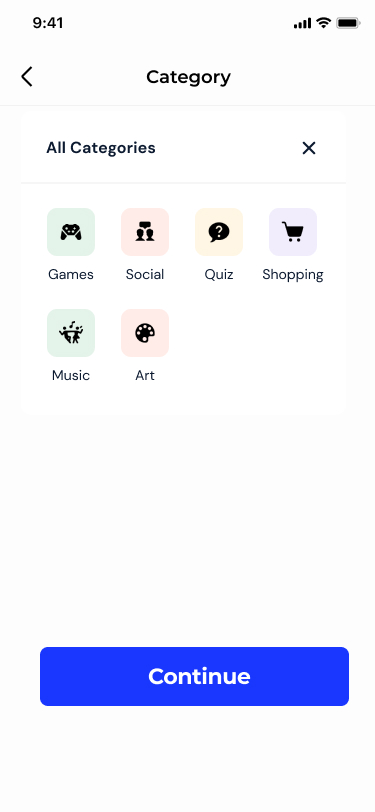
Homepage



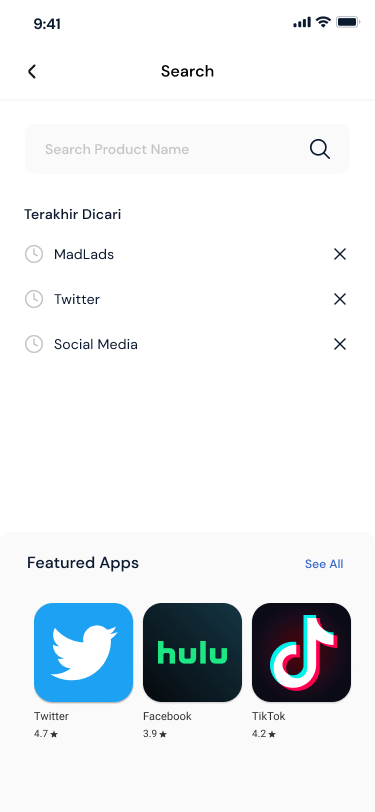
Menu 3 Garis



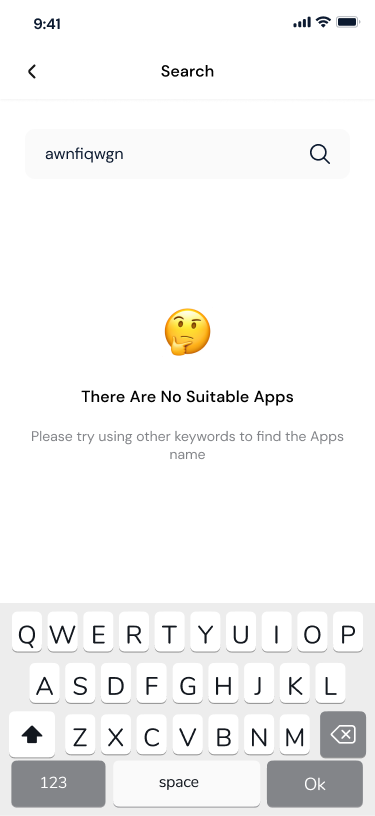
Category



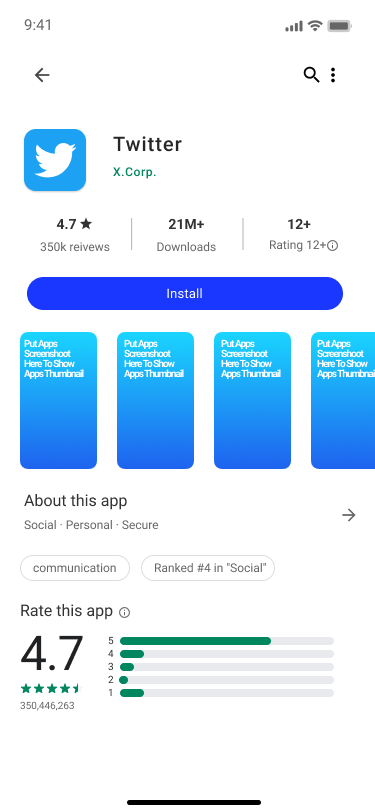
Search



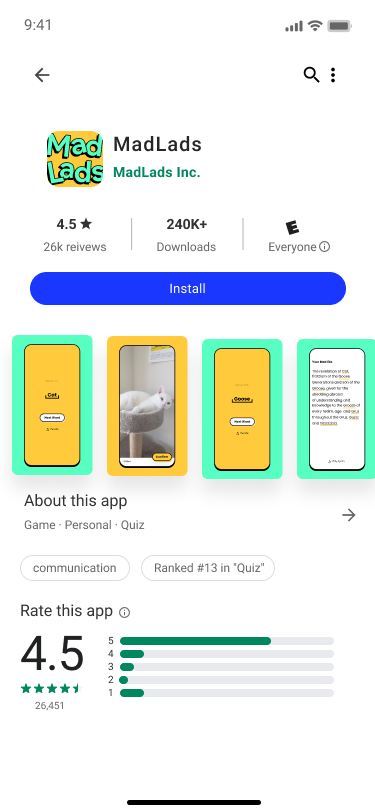
Search Not Found



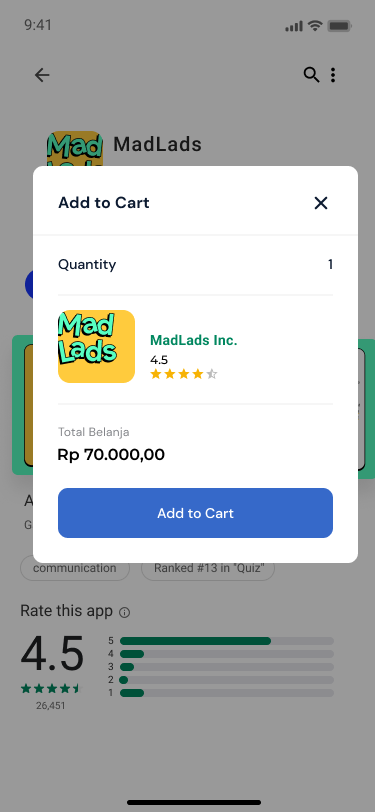
Free Apps



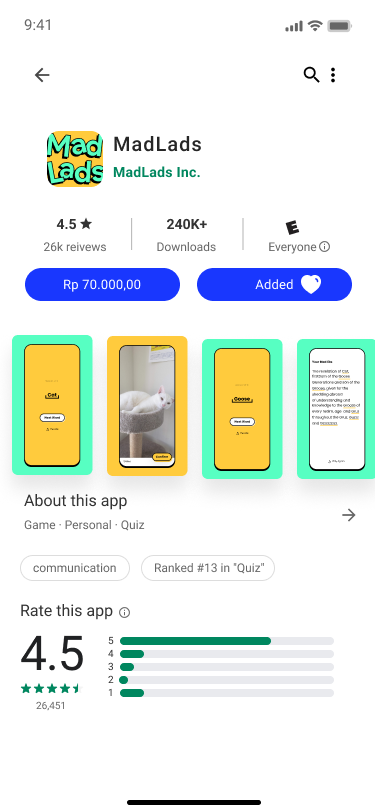
Pay Apps



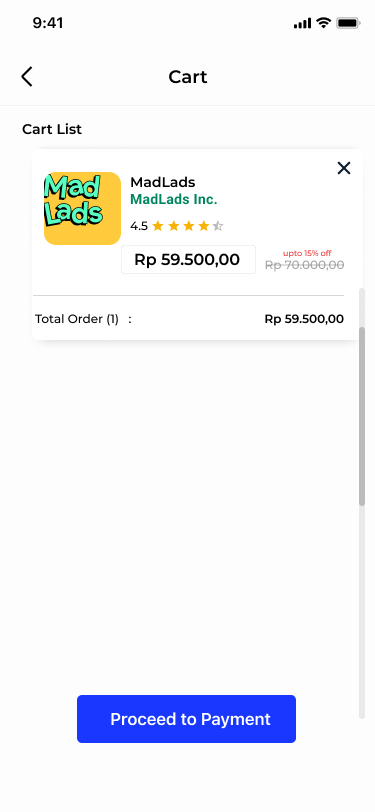
Add to Cart



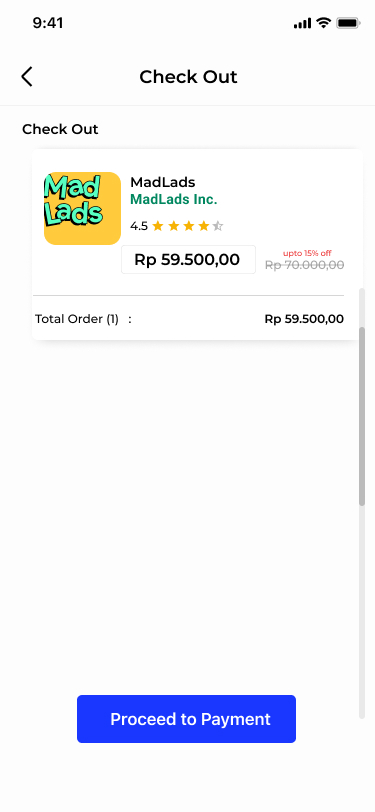
Pay Apps Added to Cart



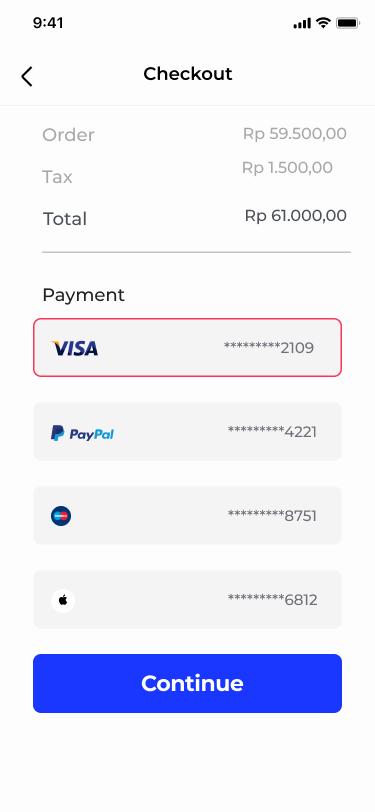
Cart



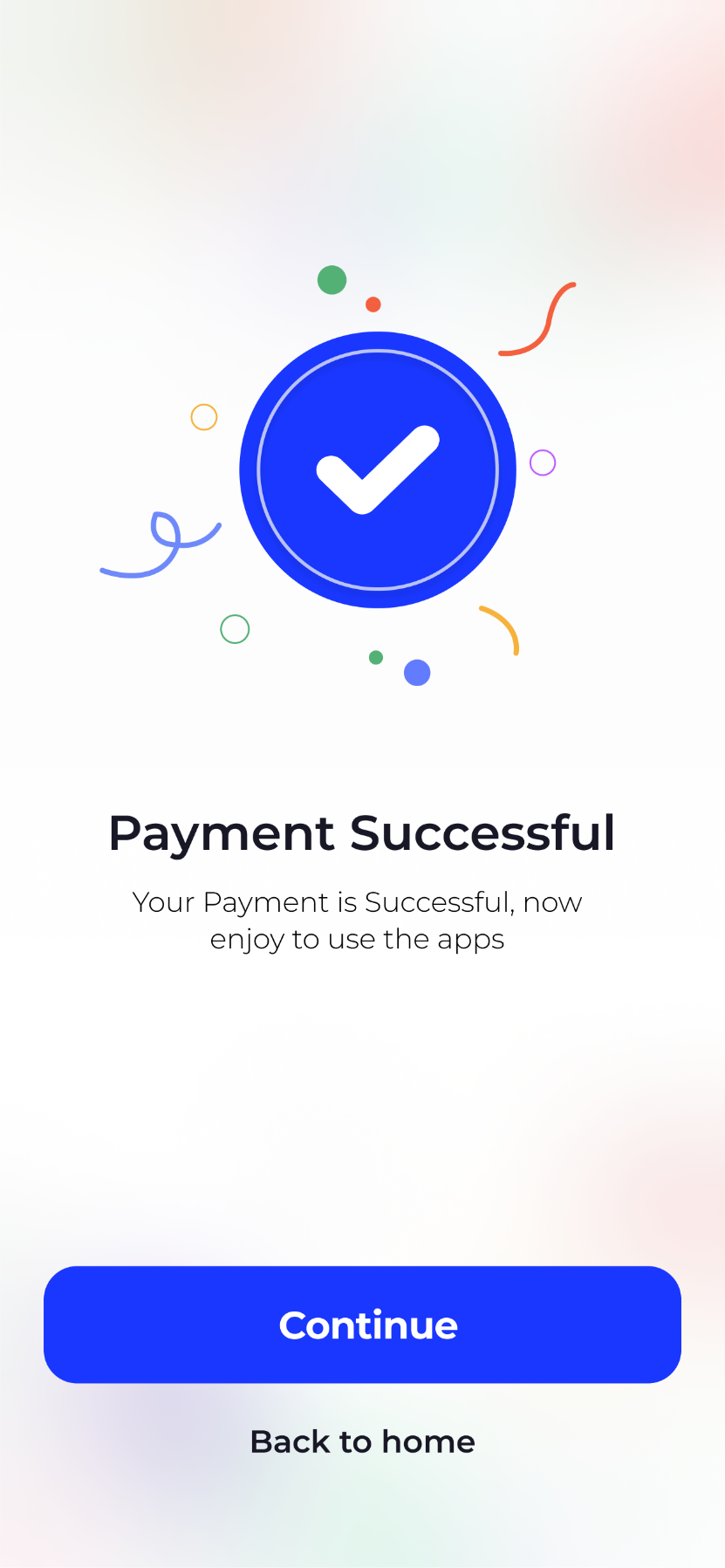
Imidiate Check Out



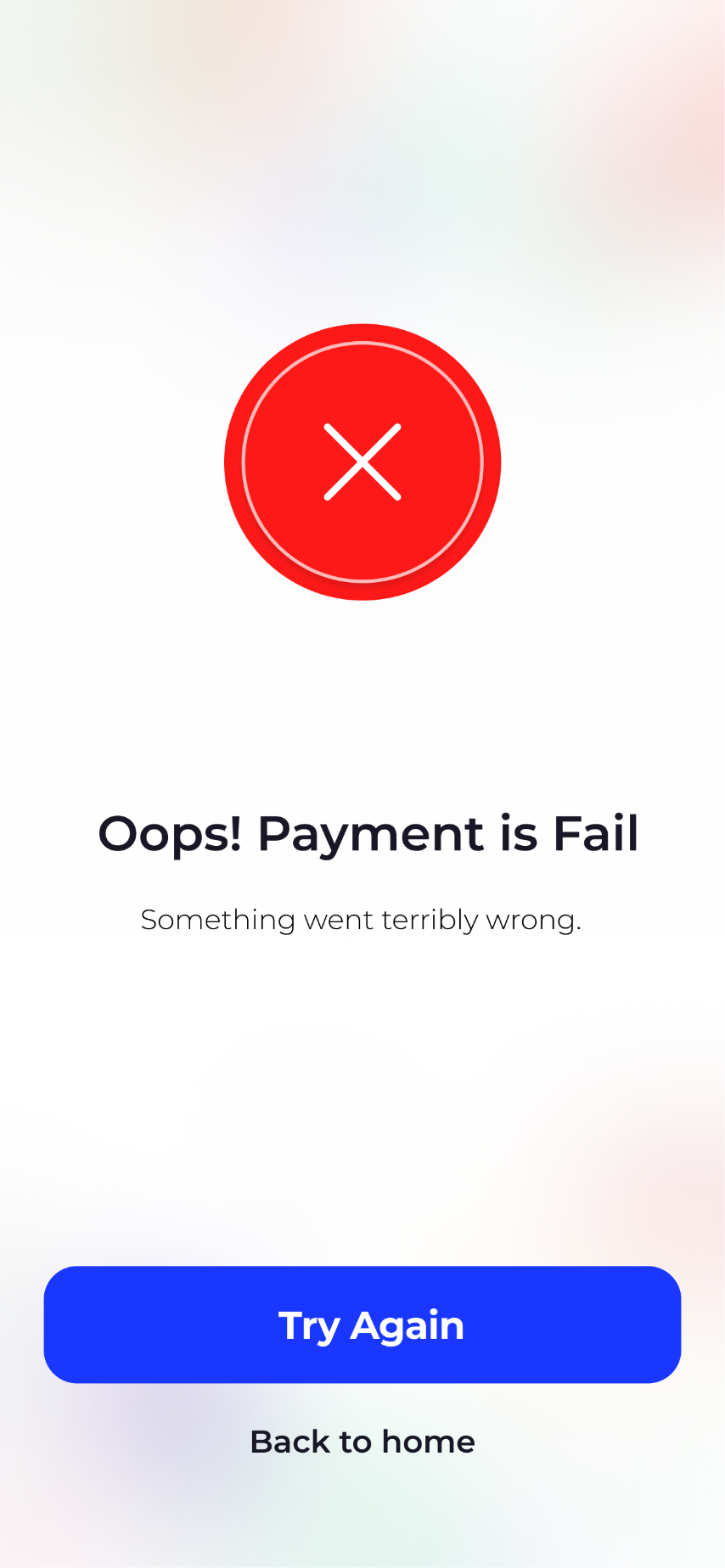
Payment For Checkout



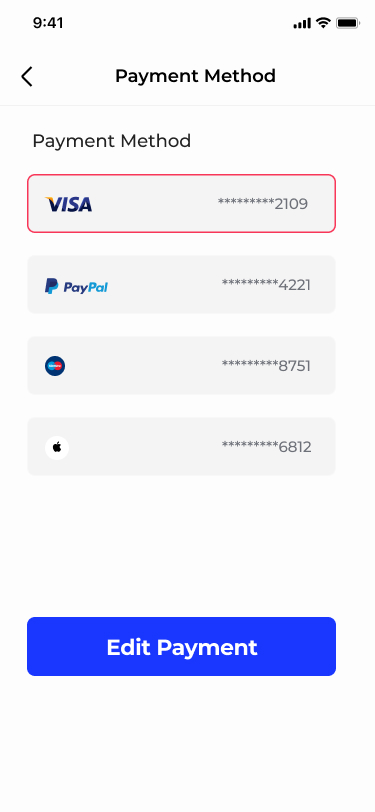
Payment Sucessful



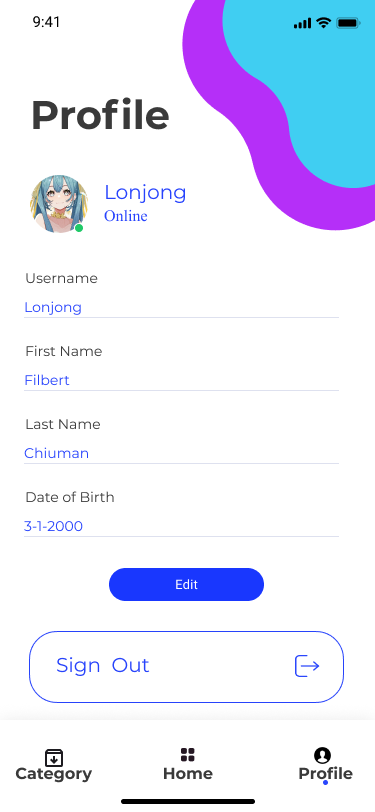
Payment Fail



Change Payment Method



Profile setting/Edit Profile



Gambaran Lengkap Mockupnya : https://www.figma.com/design/hZgnkY327Q2FKUGPxnNpGy/Apps-Market?node-id=9-661&t=U7A8eJ8I6D2qamTy-0