

NAIL SALON APPOINTMENT

PROYEK FINAL

Diajukan untuk melengkapi mata kuliah Perangkat Lunak Berbasis Komponen
Dosen Pengampu: Kurnia Saputra, S.T., M.Sc.

Oleh:

FARRAH FAHIRA MURZANI
STEFHANI ALBA SIREGAR

2008107010062
2008107010057

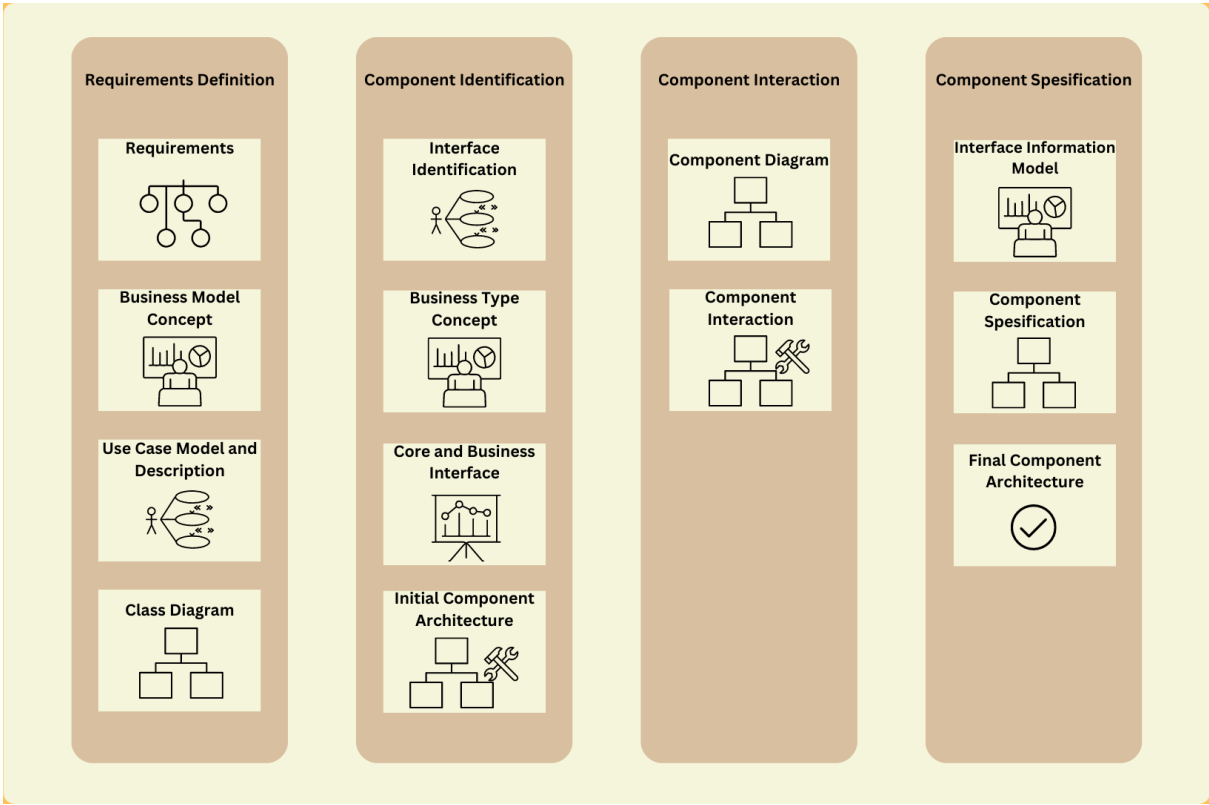


PROGRAM STUDI SARJANA INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA, BANDA ACEH
MEI, 2023

DAFTAR ISI

ARSITEKTUR COMPONENT.....	1
REQUIREMENT DEFINITION.....	2
1. Requirement.....	2
2. Business Model Concept.....	2
3. Use Case Model & Use Case Description.....	3
4. Class Diagram.....	4
COMPONENT IDENTIFICATION.....	5
1. Interface Identification.....	5
2. Business Type Model.....	5
3. Core and Business Interface.....	6
4. Initial Component Architecture.....	6
COMPONENT INTERACTION.....	7
1. Component Diagram.....	7
2. Component Interaction.....	7
COMPONENT SPECIFICATION.....	8
1. Interface Information Model.....	8
2. Component Specification.....	8
3. Final Component Architecture.....	9
OCL.....	10

ARSITEKTUR COMPONENT



REQUIREMENT DEFINITION

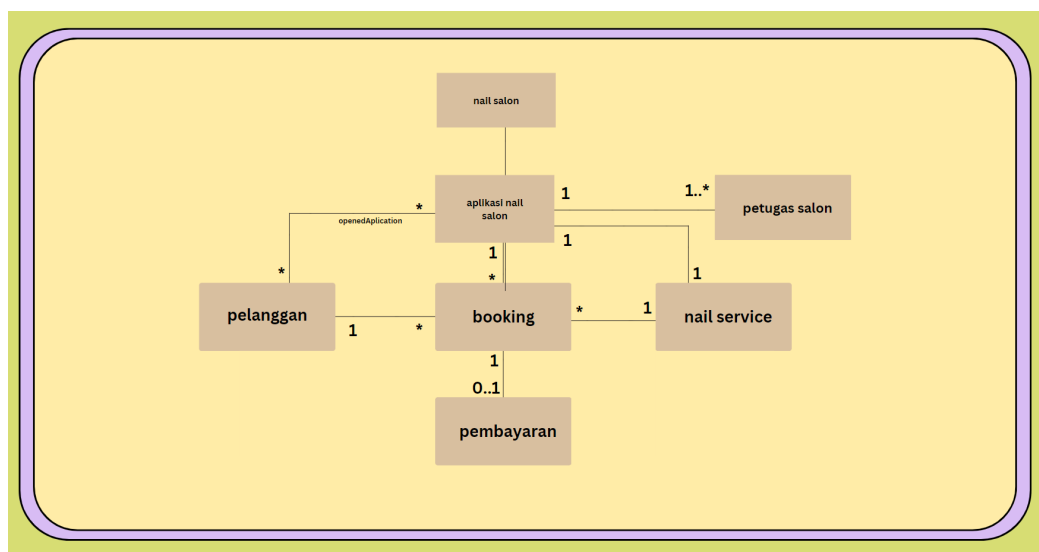
1. Requirement

Berikut adalah requirement dari sistem NailAppointment:

- R01: User dapat memilih servis
- R02: User dapat memilih metode pembayaran

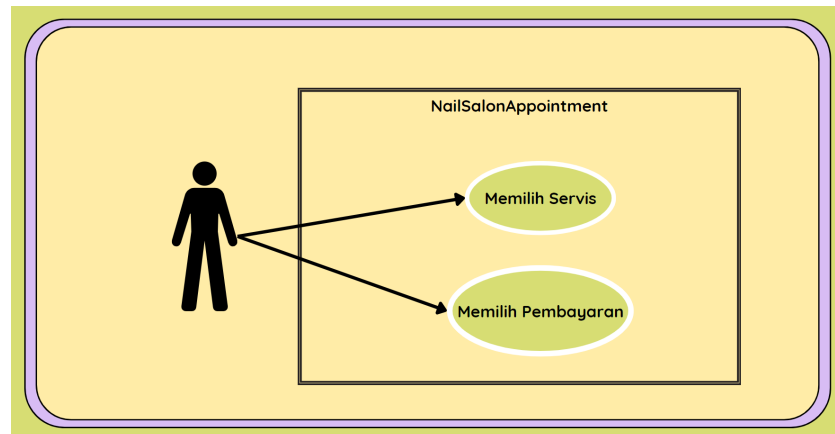
2. Business Model Concept

Berikut adalah Business Model Concept dari sistem NailAppointment:

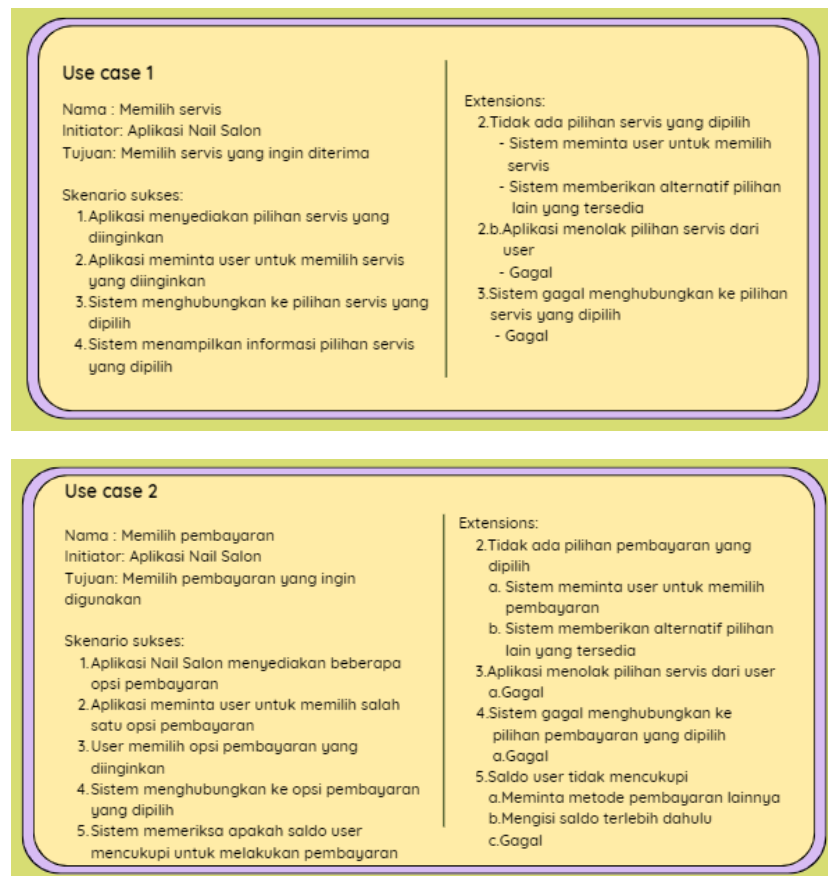


3. Use Case Model & Use Case Description

a. Use Case Model

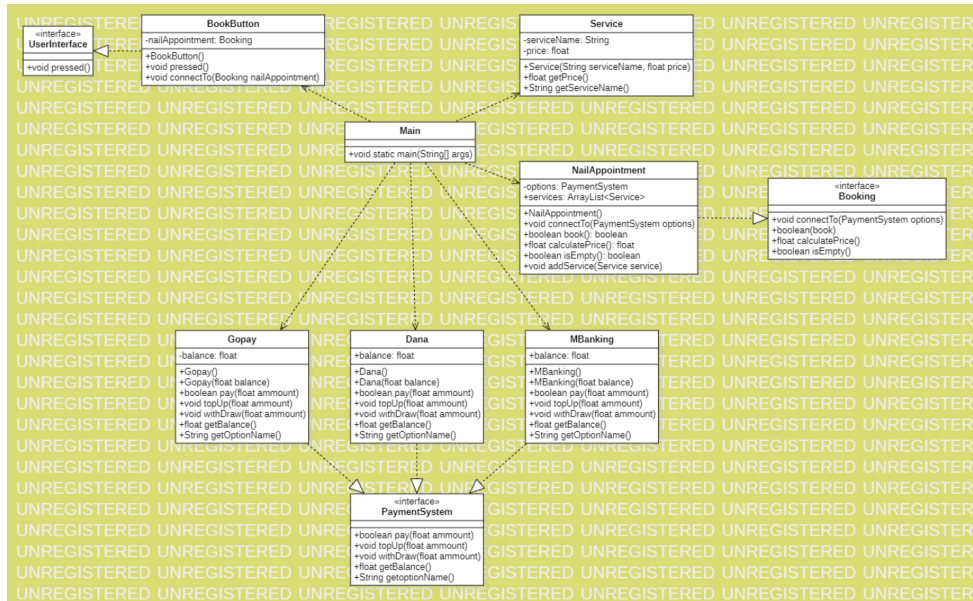


b. Use Case Description



4. Class Diagram

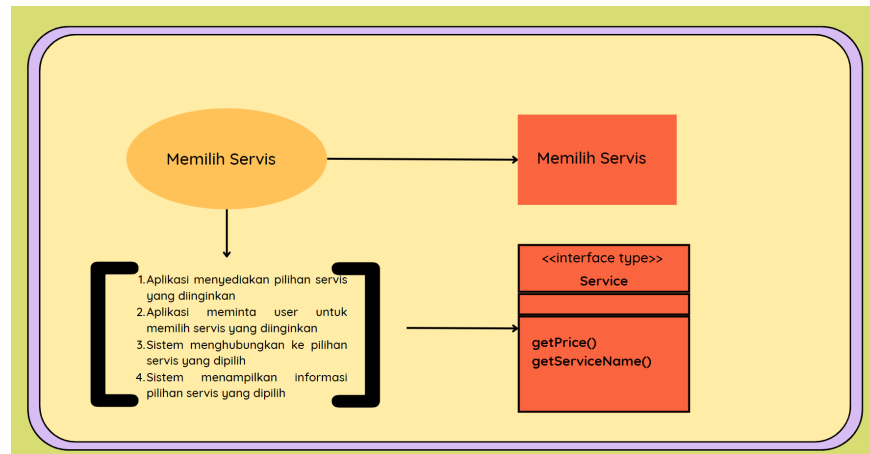
Berikut adalah Class Diagram dari sistem NailAppointment:



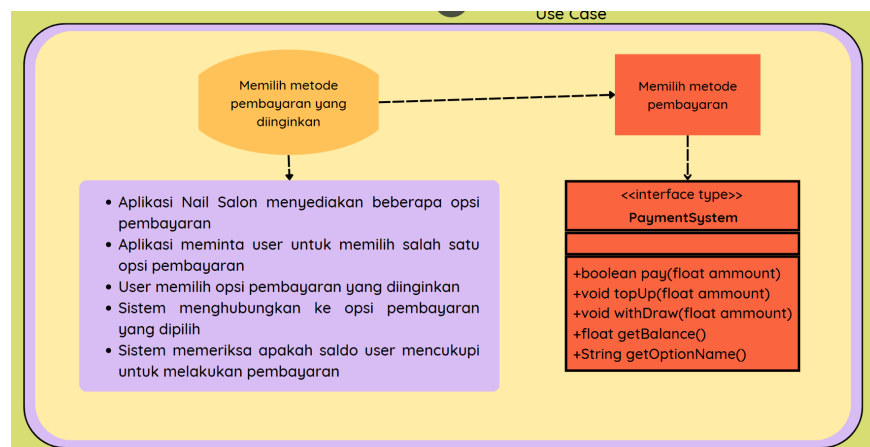
COMPONENT IDENTIFICATION

1. Interface Identification

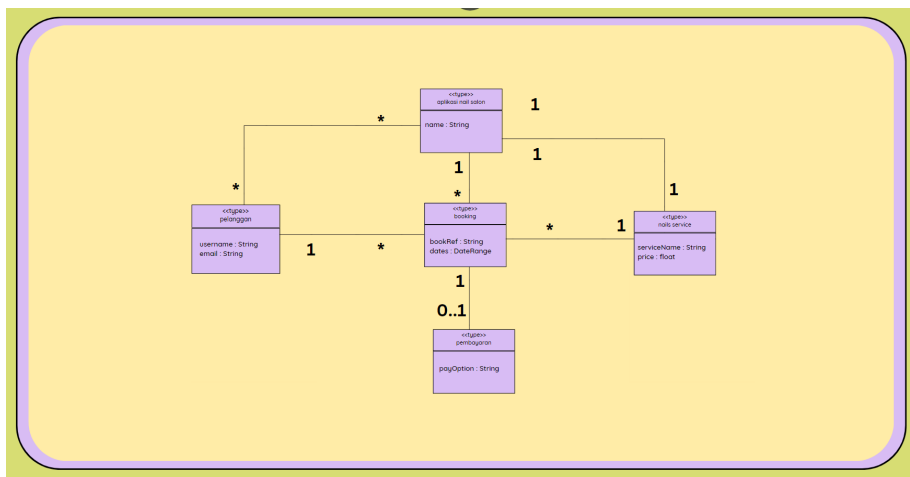
- **Interface Identification : Memilih Servis**



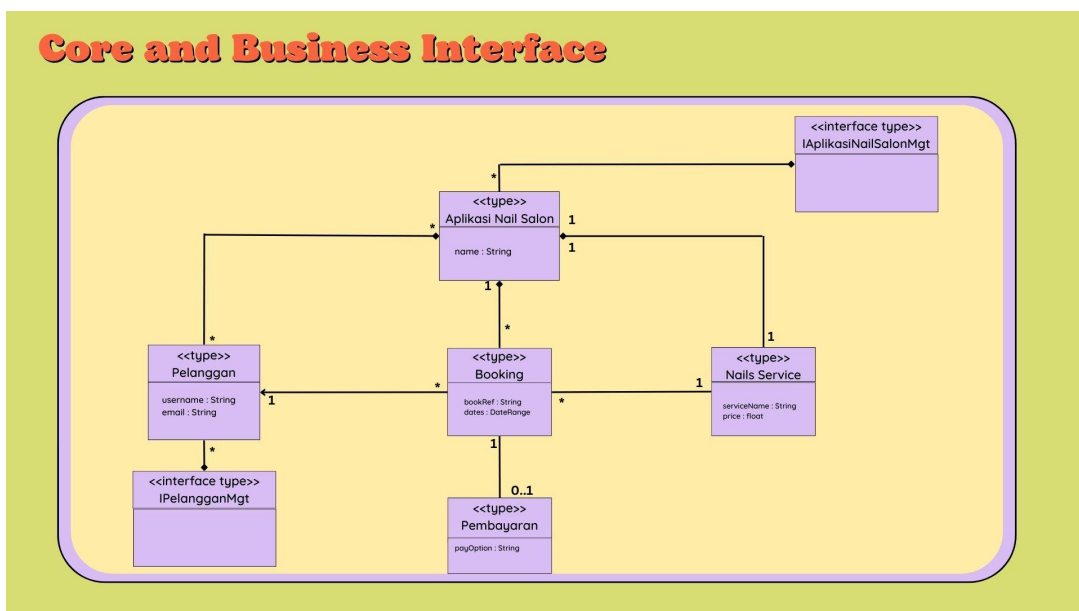
- **Interface Identification : Memilih Metode Pembayaran yang Diinginkan**



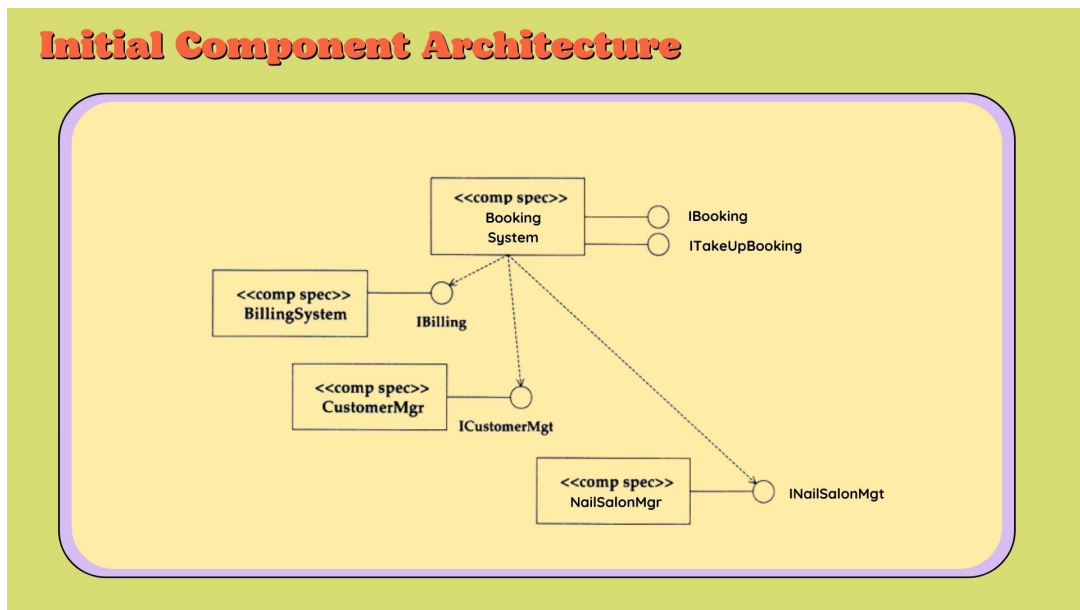
2. Business Type Model



3. Core and Business Interface

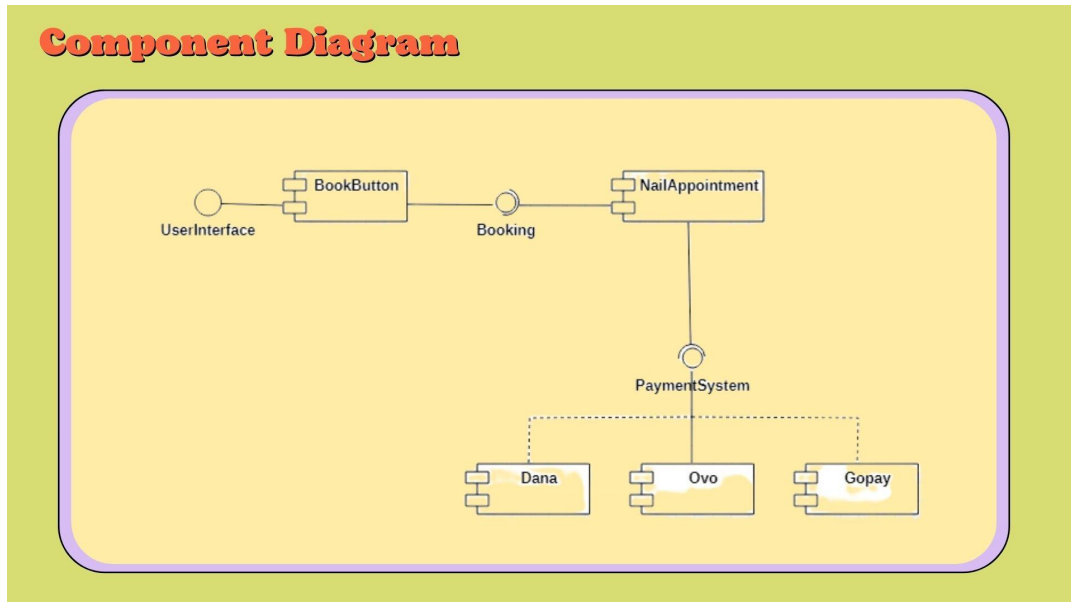


4. Initial Component Architecture

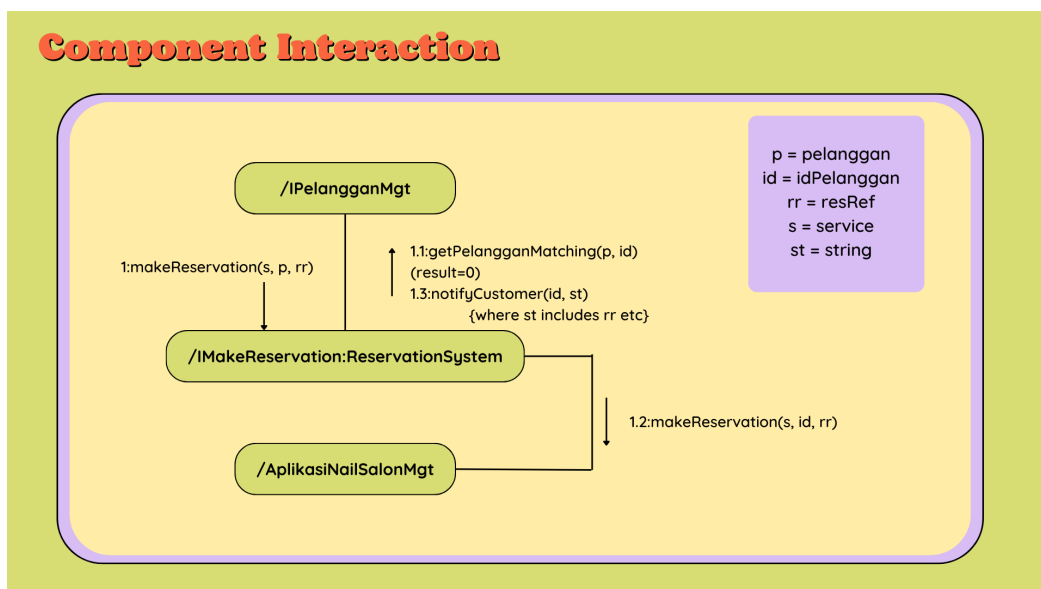


COMPONENT INTERACTION

1. Component Diagram

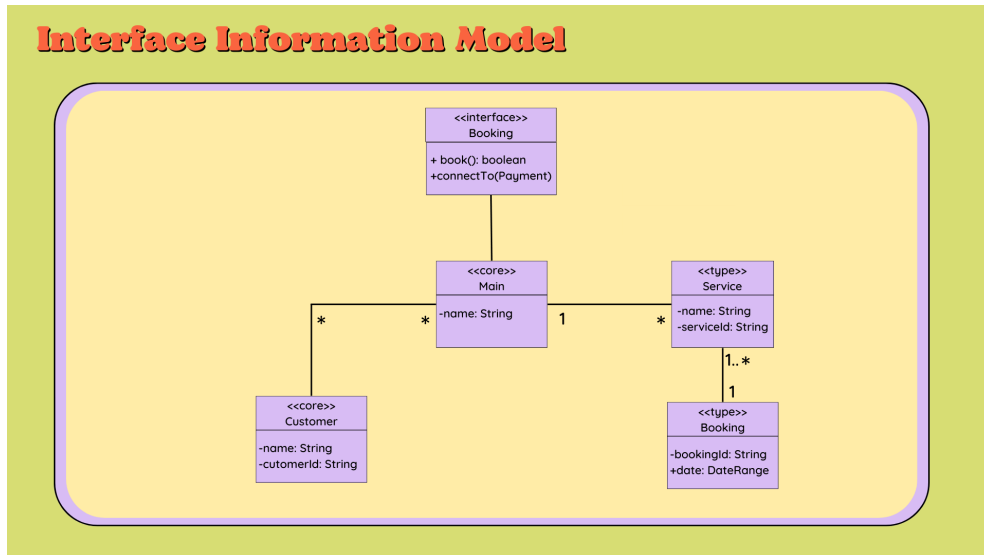


2. Component Interaction

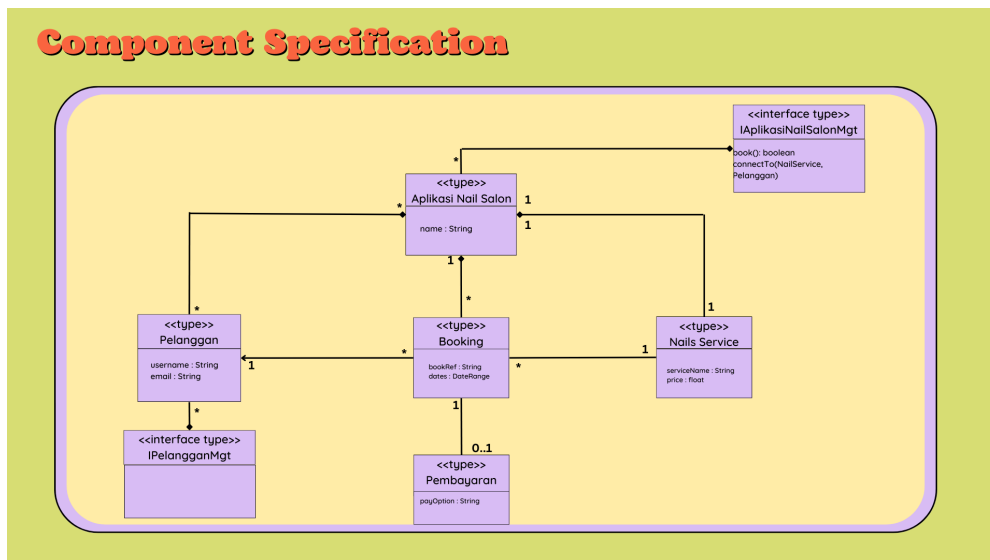


COMPONENT SPECIFICATION

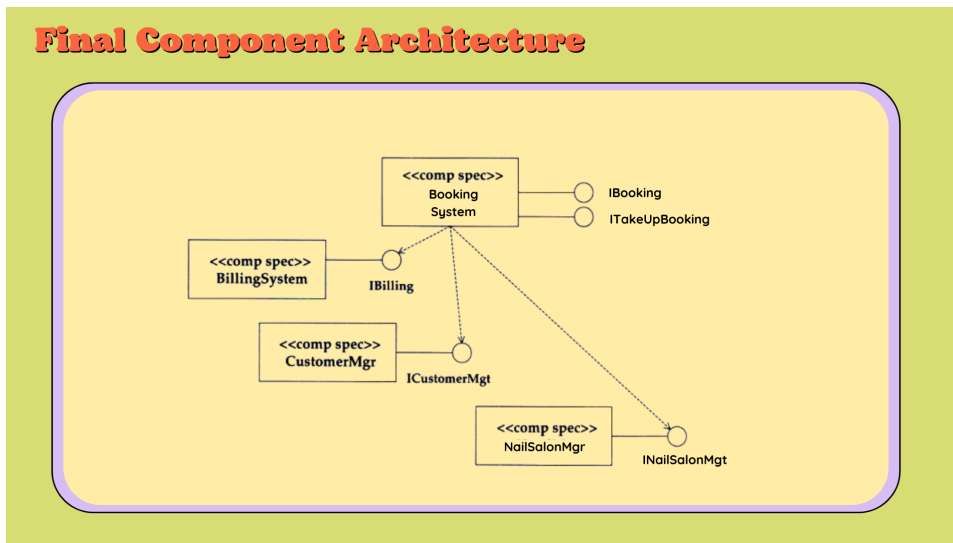
1. Interface Information Model



2. Component Specification




3. Final Component Architecture







OCL

1. BookButton.ocl

```
D: > coolyeah > sem6 > PLBK > NailAppointment > ocl >  BookButton.ocl
1  context BookButton::BookButton()
2  |   psot : self.nailAppointment = null
3
4
5  context BookButton::connectTo (nailAppointment : Book)
6  |   post : self.nailAppointment = nailAppointment
7
8
9  context BookButton::pressed()
10 |   pre : self.nailAppointment <> null
11 |   post : if nailAppointment.book() = true
12 |       |   then print "Book done succesfully!"
13 |       |   else
14 |       |   print "Book failed!"
15
```

2. Dana.ocl

```
D: > coolyeah > sem6 > PLBK > NailAppointment > ocl >  Dana.ocl
1  context Dana::Dana()
2  |   post : self.balance = 0
3
4
5  context Dana::Dana(ammount: Real)
6  |   post : self.balance = ammount
7
8
9  context Dana::pay(ammount: Real) : Boolean
10 |   init : false
11 |   inv : self.balance >= 0
12 |   pre : self.getBalance() >= ammount
13 |   post : self.withdraw(ammount)
14 |       |   and self@pre.getBalance()  self.getBalance() = ammount
15 |       |   and result = true
16
17
18 context Dana::topUp(ammount: Real)
19 |   inv : self.balance >= 0
20 |   post : self.balance + ammount
21 |       |   and self.balance = self@pre.getBalance() + ammount
22
23
24 context Dana::withdraw(ammount: Real)
25 |   inv : self.balance >= 0
26 |   post : self.balance - ammount
27 |       |   and self.balance = self@pre.getBalance() - ammount
28
29
30 context Dana::getBalance() : Real
31 |   post : result = self.balance
32
33
34 context Dana::getOptionName() : String
35 |   post : result = "Dana"
36
```

3. Gopay.ocl

```
D: > coolyeah > sem6 > PLBK > NailAppointment > ocl > ≡ Gopay.ocl
1  context Gopay::Gopay()
2  |   post : self.balance = 0
3
4
5  context Gopay::Gopay(ammount: Real)
6  |   post : self.balance = ammount
7
8
9  context Gopay::pay(ammount: Real) : Boolean
10 |   init : false
11 |   inv  : self.balance >= 0
12 |   pre  : self.getBalance() >= ammount
13 |   post : self.withdraw(ammount)
14 |       and self@pre.getBalance() ≤ self.getBalance() = ammount
15 |       and result = true
16
17
18 context Gopay::topUp(ammount: Real)
19 |   inv  : self.balance >= 0
20 |   post : self.balance + ammount
21 |       and self.balance = self@pre.getBalance() + ammount
22
23
24 context Gopay::withdraw(ammount: Real)
25 |   inv  : self.balance >= 0
26 |   post : self.balance - ammount
27 |       and self.balance = self@pre.getBalance() - ammount
28
29
30 context Gopay::getBalance() : Real
31 |   post : result = self.balance
32
33
34 context Gopay::getOptionName() : String
35 |   post : result = "Gopay"
36
```

4. NailAppointment.ocl

```
D: > coolyeah > sem6 > PLBK > NailAppointment > ocl > ≡ NailAppointment.ocl
1  context CheckoutButton::CheckoutButton()
2  |   post : self.options = null
3
4
5  context NailAppointment::connectTo (options : PaymentSystem)
6  |   post : self.options = options
7
8
9  context NailAppointment::checkout() : Boolean
10 |   init : false
11 |   inv  : self.services->size() >= 0
12 |   pre  : self.isEmpty() <> true
13 |   |   and self.options <> null
14 |   post : if self.options.pay(calculatePrice()) = true
15 |   |   then result = true
16
17
18 context NailAppointment::calculatePrice() : Real
19 |   init : 0
20 |   pre  : self.services->size() > 0
21 |   post : self.services->iterate(p: Service, sum: Real=0
22 |   |   | sum + p.getPrice()
23 |   |   and result = sum)
24
25
26 context NailAppointment::addService(product: Service)
27 |   inv  : self.services->size() >= 0
28 |   pre  : product <> null
29 |   post : services->size() ≤ services@pre->size() = 1
30 |   |   and services->exists(p: Service
31 |   |   | p.getServiceId() = product.getServiceId())
32
33
34 context NailAppointment::isEmpty() : Boolean
35 |   init : false
36 |   post : result = self.services->size() == 0
37
```

5. Ovo.ocl

```
D: > coolyeah > sem6 > PLBK > NailAppointment > ocl > ≡ Ovo.ocl
1  context Ovo::Ovo()
2  |   post : self.balance = 0
3
4
5  context Ovo::Ovo(ammount: Real)
6  |   post : self.balance = ammount
7
8
9  context Ovo::pay(ammount: Real) : Boolean
10 |   init : false
11 |   inv  : self.balance >= 0
12 |   pre  : self.getBalance() >= ammount
13 |   post : self.withdraw(ammount)
14 |       and self@pre.getBalance() ≤ self.getBalance() = ammount
15 |       and result = true
16
17
18 context Ovo::topUp(ammount: Real)
19 |   inv  : self.balance >= 0
20 |   post : self.balance + ammount
21 |       and self.balance = self@pre.getBalance() + ammount
22
23
24 context Ovo::withdraw(ammount: Real)
25 |   inv  : self.balance >= 0
26 |   post : self.balance - ammount
27 |       and self.balance = self@pre.getBalance() - ammount
28
29
30 context Ovo::getBalance() : Real
31 |   post : result = self.balance
32
33
34 context Ovo::getOptionName() : String
35 |   post : result = "Ovo"
36
```

6. Service.ocl

```
D: > coolyeah > sem6 > PLBK > NailAppointment > ocl > ≡ Service.ocl
1  context Service::Service(serviceId: String, price: Real)
2  |   post : self.serviceId = serviceId and self.price = price
3
4  context Service::getPrice() : Real
5  |   pre  : self.price <> null
6  |   post : result = self.price
7
8  context Service::getServiceId() : String
9  |   pre  : self.serviceId <> null
10 |   post : result = self.serviceId
11
```