Interaction overview diagram

2026-11-19 · `examples/types/uml-interaction-overview.json`

7 views · 21 cards · 29 lines

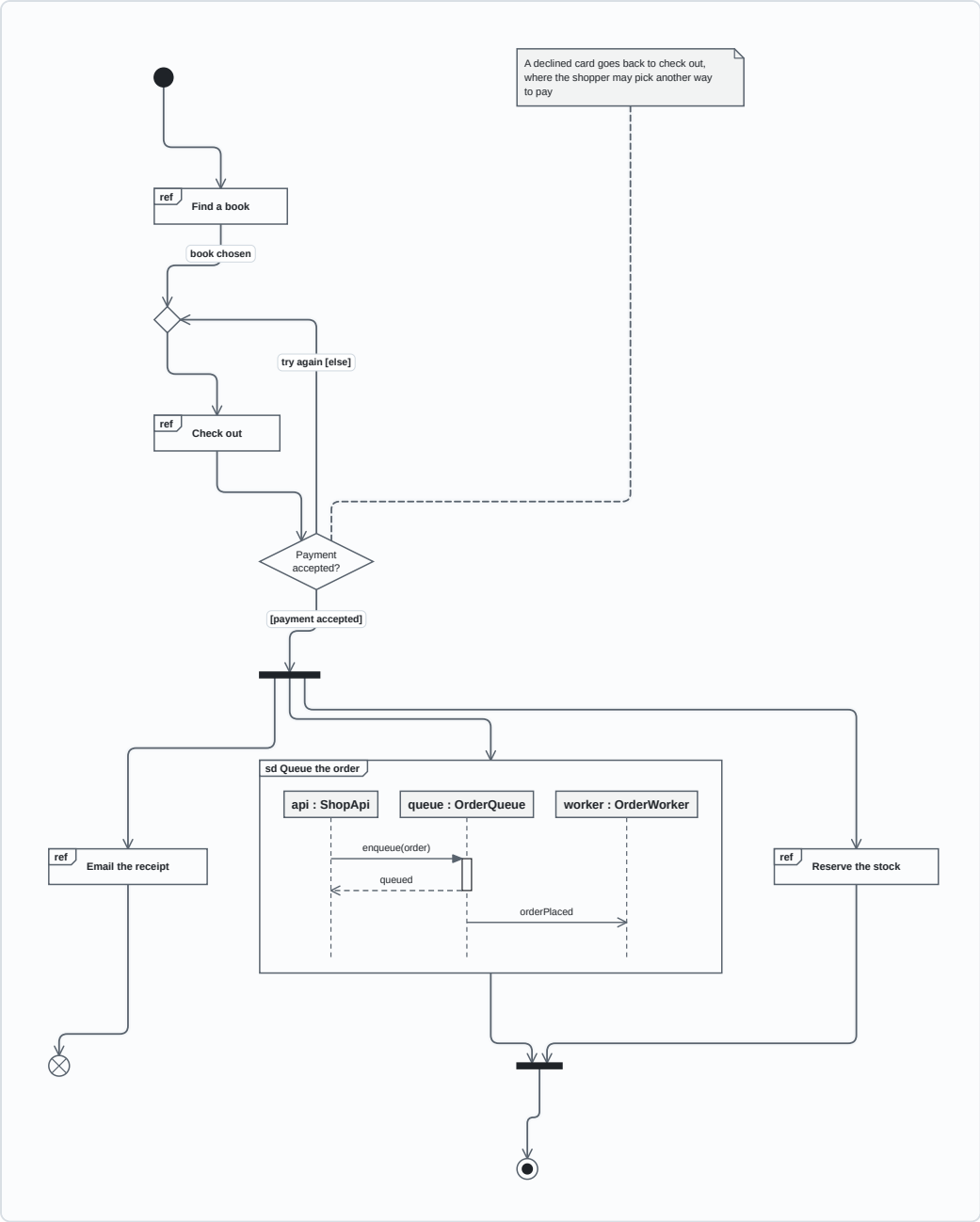
Contents

Buying a book 2

- Buying a book, laid out right 4
- Find a book 5
- Check out 6
- Reserve the stock 7
- Email the receipt 8
- Queue the order 9

Buying a book

13 cards · 14 lines



Legend → **Control flow** what runs next · ---- **Comment link** what a comment is about

● **Initial node** where the flow starts · ◉ **Activity final** ends the whole flow

⊗ **Flow final** ends one branch · ◇ **Decision** one way out, by its guards

◇ **Merge** where ways meet · ▌ **Fork** splits into branches that run together

▌ **Join** waits for all its branches

Elements 13

Card	Kind	Members
Initial node	Initial node	
Find a book	Interaction use	
Merge node	Merge node	
Check out	Interaction use	
Payment accepted?	Decision node	
Fork node	Fork node	
Email the receipt	Interaction use	
Reserve the stock	Interaction use	
Queue the order	Interaction	
Flow final node	Flow final node	
Join node	Join node	
Activity final node	Activity final node	
A declined card goes back to check out, where the shopper may pick another way to pay	Comment	

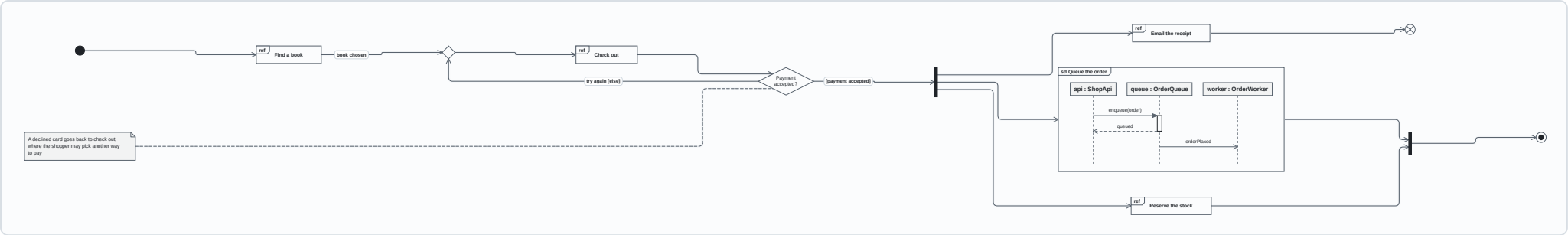
Lines 14

From → to	Label	Kind
Initial node → Find a book		Control flow
Find a book → Merge node	book chosen	Control flow
Merge node → Check out		Control flow
Check out → Payment accepted?		Control flow
Payment accepted? → Fork node		Control flow

Buying a book (continued)

Lines 14 (continued)

From → to	Label	Kind
Payment accepted? → Merge node	try again	Control flow
Fork node → Email the receipt		Control flow
Fork node → Queue the order		Control flow
Fork node → Reserve the stock		Control flow
Email the receipt → Flow final node		Control flow
Queue the order → Join node		Control flow
Reserve the stock → Join node		Control flow
Join node → Activity final node		Control flow
A declined card goes back to check out, where the shopper may pick another way to pay → Payment accepted?		Comment link



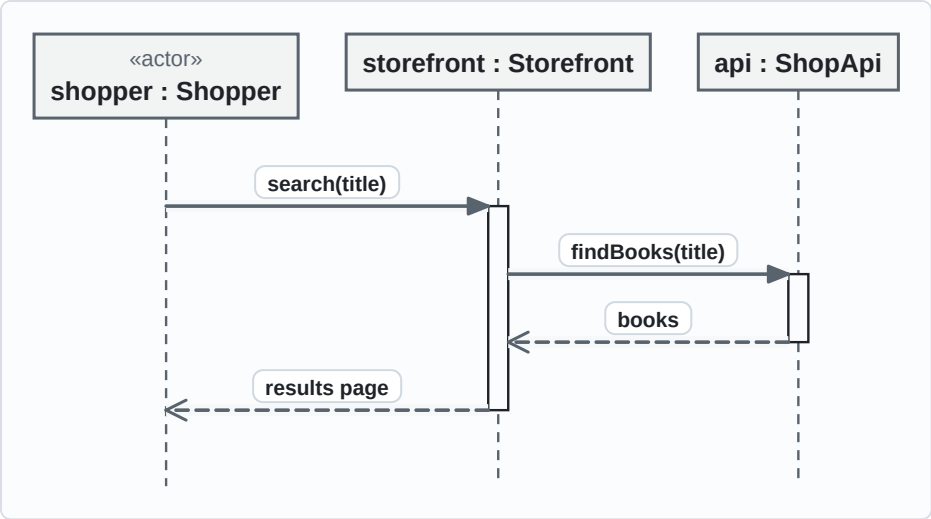
Legend **Control flow** what runs next · **Comment link** what a comment is about · **Initial node** where the flow starts · **Activity final** ends the whole flow · **Flow final** ends one branch · **Decision** one way out, by its guards · **Merge** where ways meet · **Fork** splits into branches that run together · **Join** waits for all its branches

Elements 13

Card	Kind	Members
Initial node	Initial node	
Find a book	Interaction use	
Merge node	Merge node	
Check out	Interaction use	
Payment accepted?	Decision node	
Fork node	Fork node	
Email the receipt	Interaction use	
Reserve the stock	Interaction use	
Queue the order	Interaction	
Flow final node	Flow final node	
Join node	Join node	
Activity final node	Activity final node	
A declined card goes back to check out, where the shopper may pick another way to pay	Comment	

Lines 14

From → to	Label	Kind
Initial node → Find a book		Control flow
Find a book → Merge node	book chosen	Control flow
Merge node → Check out		Control flow
Check out → Payment accepted?		Control flow
Payment accepted? → Fork node		Control flow
Payment accepted? → Merge node	try again	Control flow
Fork node → Email the receipt		Control flow
Fork node → Queue the order		Control flow
Fork node → Reserve the stock		Control flow
Email the receipt → Flow final node		Control flow
Queue the order → Join node		Control flow
Reserve the stock → Join node		Control flow
Join node → Activity final node		Control flow
A declined card goes back to check out, where the shopper may pick another way to pay → Payment accepted?		Comment link



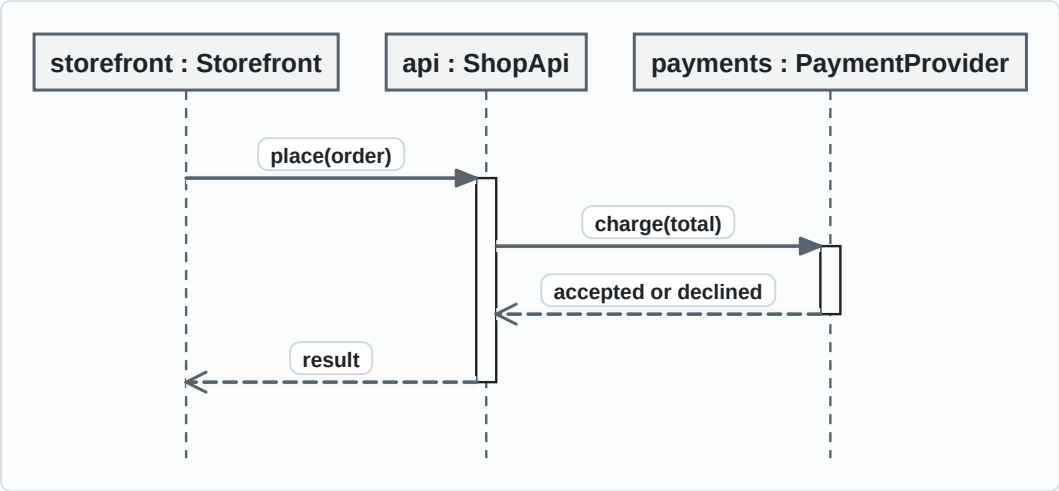
Legend → **Call** the sender waits for it --> **Reply** the answer to a call

Elements 3

Card	Kind	Members
shopper : Shopper	Lifeline	
storefront : Storefront	Lifeline	
api : ShopApi	Lifeline	

Lines 4

From → to	Label	Kind
shopper : Shopper → storefront : Storefront	search(title)	Message
storefront : Storefront → api : ShopApi	findBooks(title)	Message
api : ShopApi → storefront : Storefront	books	Message
storefront : Storefront → shopper : Shopper	results page	Message



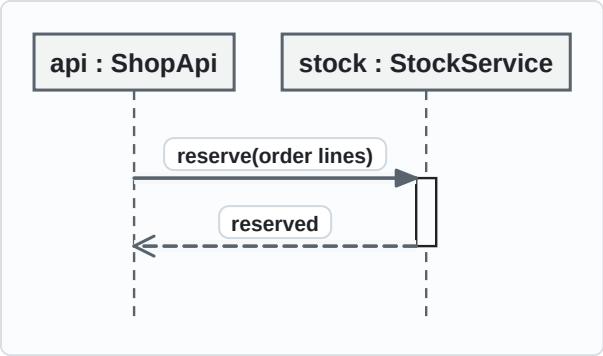
Legend → **Call** the sender waits for it --> **Reply** the answer to a call

Elements 3

Card	Kind	Members
storefront : Storefront	Lifeline	
api : ShopApi	Lifeline	
payments : PaymentProvider	Lifeline	

Lines 4

From → to	Label	Kind
storefront : Storefront → api : ShopApi	place(order)	Message
api : ShopApi → payments : PaymentProvider	charge(total)	Message
payments : PaymentProvider → api : ShopApi	accepted or declined	Message
api : ShopApi → storefront : Storefront	result	Message



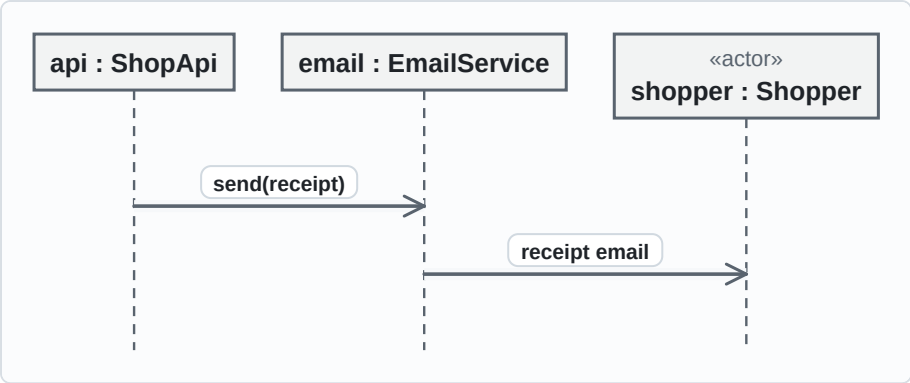
Legend → **Call** the sender waits for it --> **Reply** the answer to a call

Elements 2

Card	Kind	Members
api : ShopApi	Lifeline	
stock : StockService	Lifeline	

Lines 2

From → to	Label	Kind
api : ShopApi → stock : StockService	reserve(order lines)	Message
stock : StockService → api : ShopApi	reserved	Message



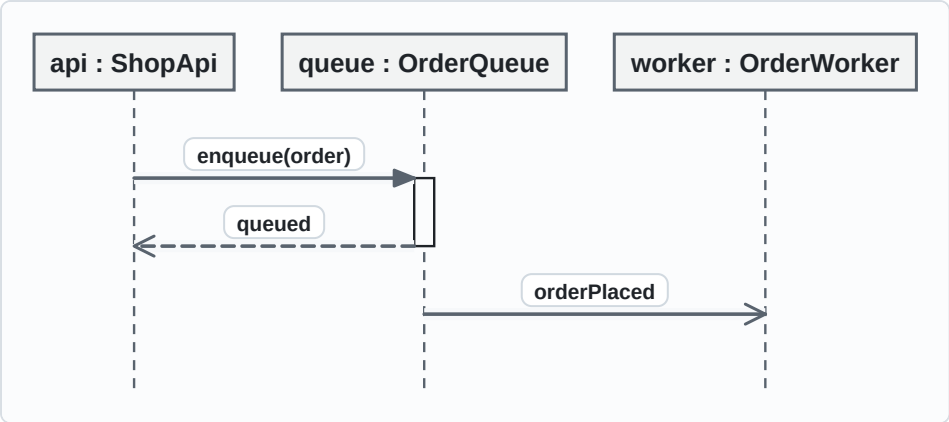
Legend —> **Async** the sender goes on

Elements 3

Card	Kind	Members
api : ShopApi	Lifeline	
email : EmailService	Lifeline	
shopper : Shopper	Lifeline	

Lines 2

From → to	Label	Kind
api : ShopApi → email : EmailService	send(receipt)	Message
email : EmailService → shopper : Shopper	receipt email	Message



Legend → **Call** the sender waits for it → **Async** the sender goes on
--> **Reply** the answer to a call

Elements 3

Card	Kind	Members
api : ShopApi	Lifeline	
queue : OrderQueue	Lifeline	
worker : OrderWorker	Lifeline	

Lines 3

From → to	Label	Kind
api : ShopApi → queue : OrderQueue	enqueue(order)	Message
queue : OrderQueue → api : ShopApi	queued	Message
queue : OrderQueue → worker : OrderWorker	orderPlaced	Message