SOLID Principles

Mobile Geeks - September 2020

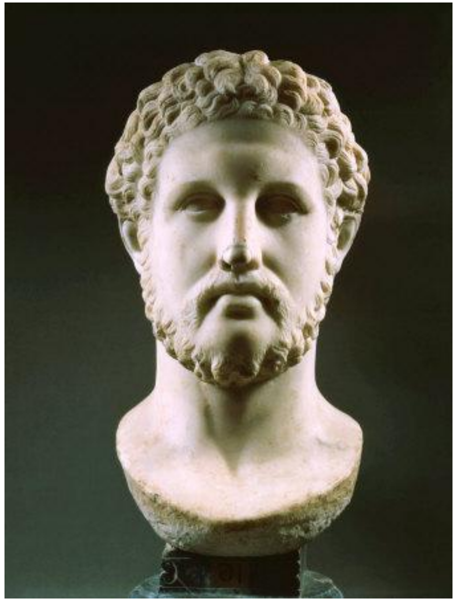
What is SOLID

Introduced in 2000 paper *Design Principles and Design Patterns* by this guy, Robert Martin (a.k.a. Uncle Bob):

What is SOLID

SOLID is a complicated way of saying “Divide et Impera”

Purported actual author: Philip II of Macedon



What is SOLID

Single Responsibility Principle
Open/Close Principle
Liskov Substitution Principle
Interface Segregation Principle
Dependency Inversion Principle

Essentially:
Take a monolith and break it down into constituent independent modules.

SOLID & Mobile Development

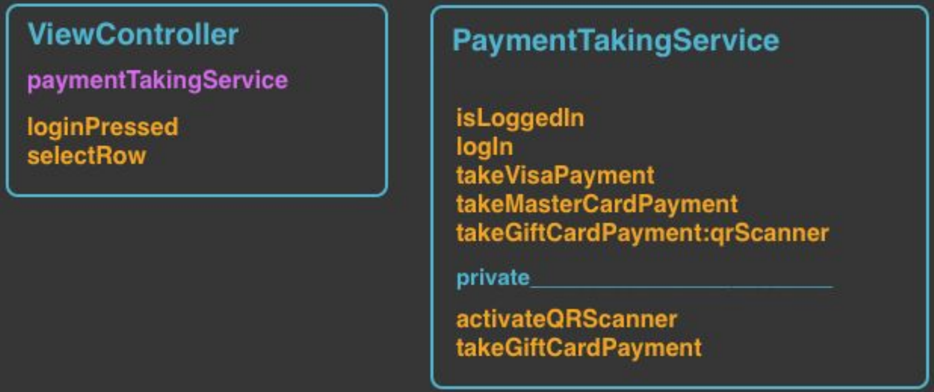
All Software has to be easy to:

- Maintain
- Extend
- Read
- (implicitly) Test

Software written for Mobile is often:

- None of these things

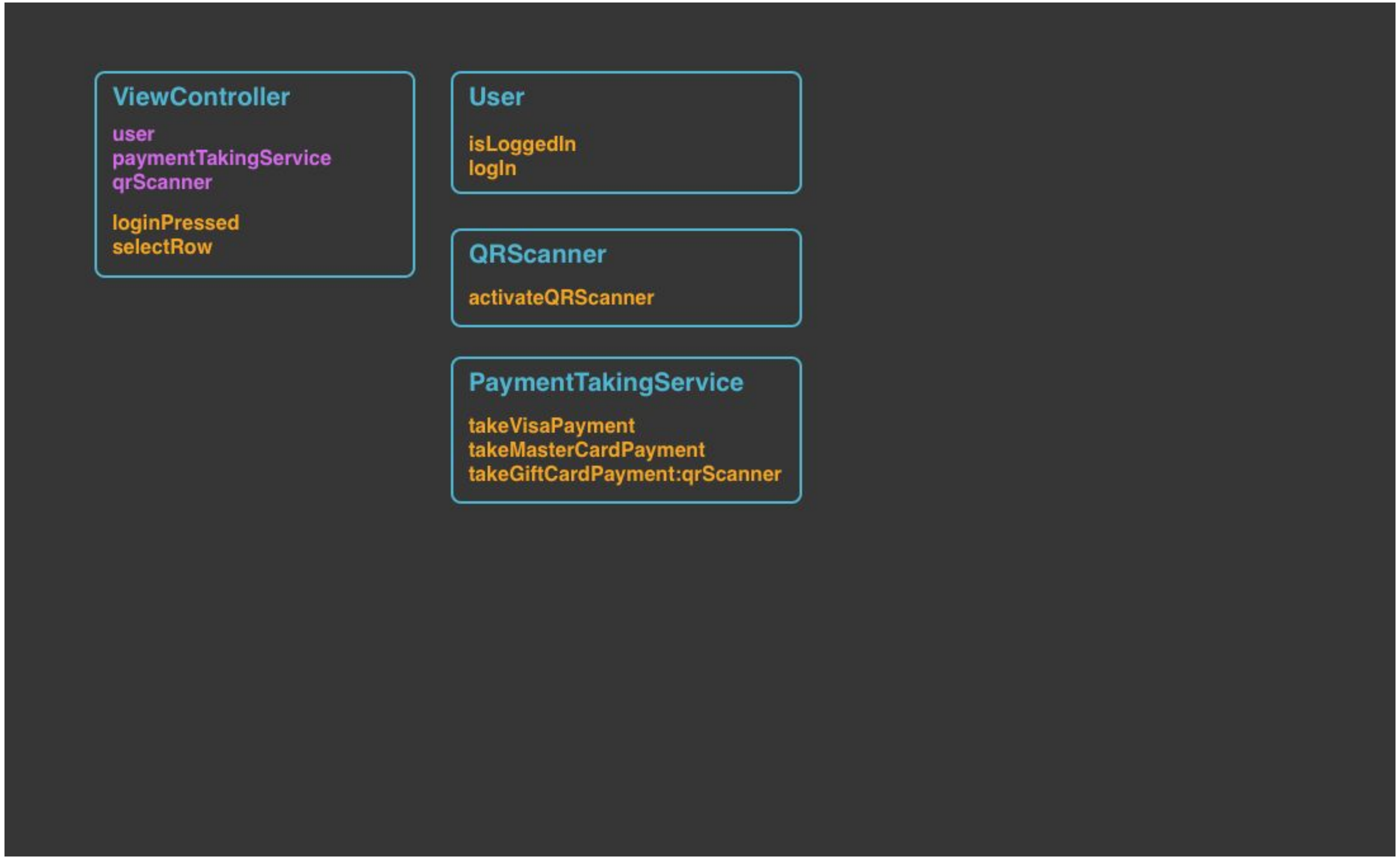
Demo Time!


ViewController
paymentTakingService
loginPressed
selectRow
PaymentTakingService
isLoggedIn
logIn
takeVisaPayment
takeMasterCardPayment
takeGiftCardPayment:qrScanner
private
activateQRScanner
takeGiftCardPayment

Single Responsibility Principle

Unsurprisingly: A class should only have a single responsibility

Single Responsibility Principle



Open-Closed Principle

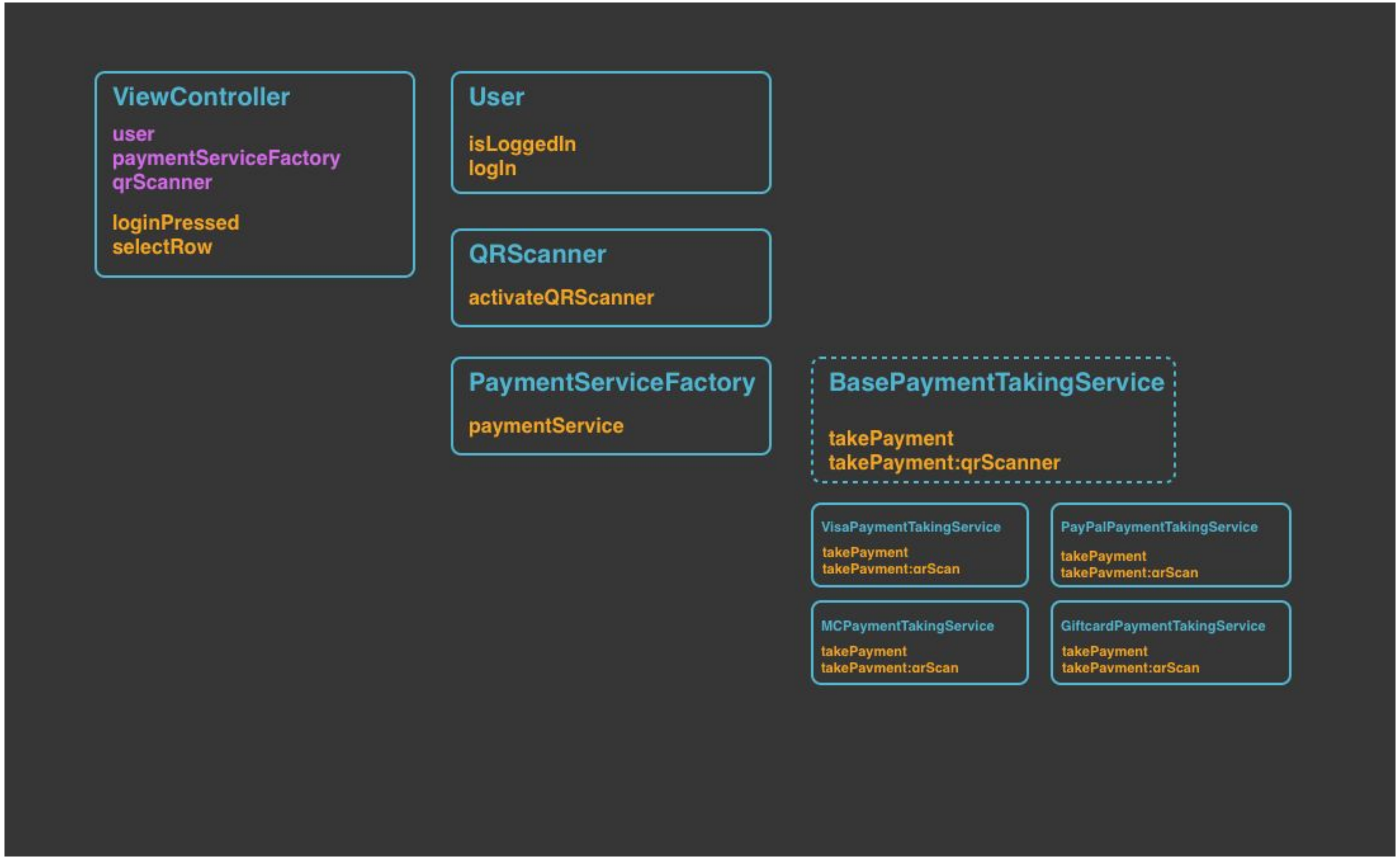
Software should be open for extension, but closed for modification.

Liskov Substitution Principle

Objects should be replaceable with instances of their subtypes without altering the program.

Replace one lens with another, the microscope still works

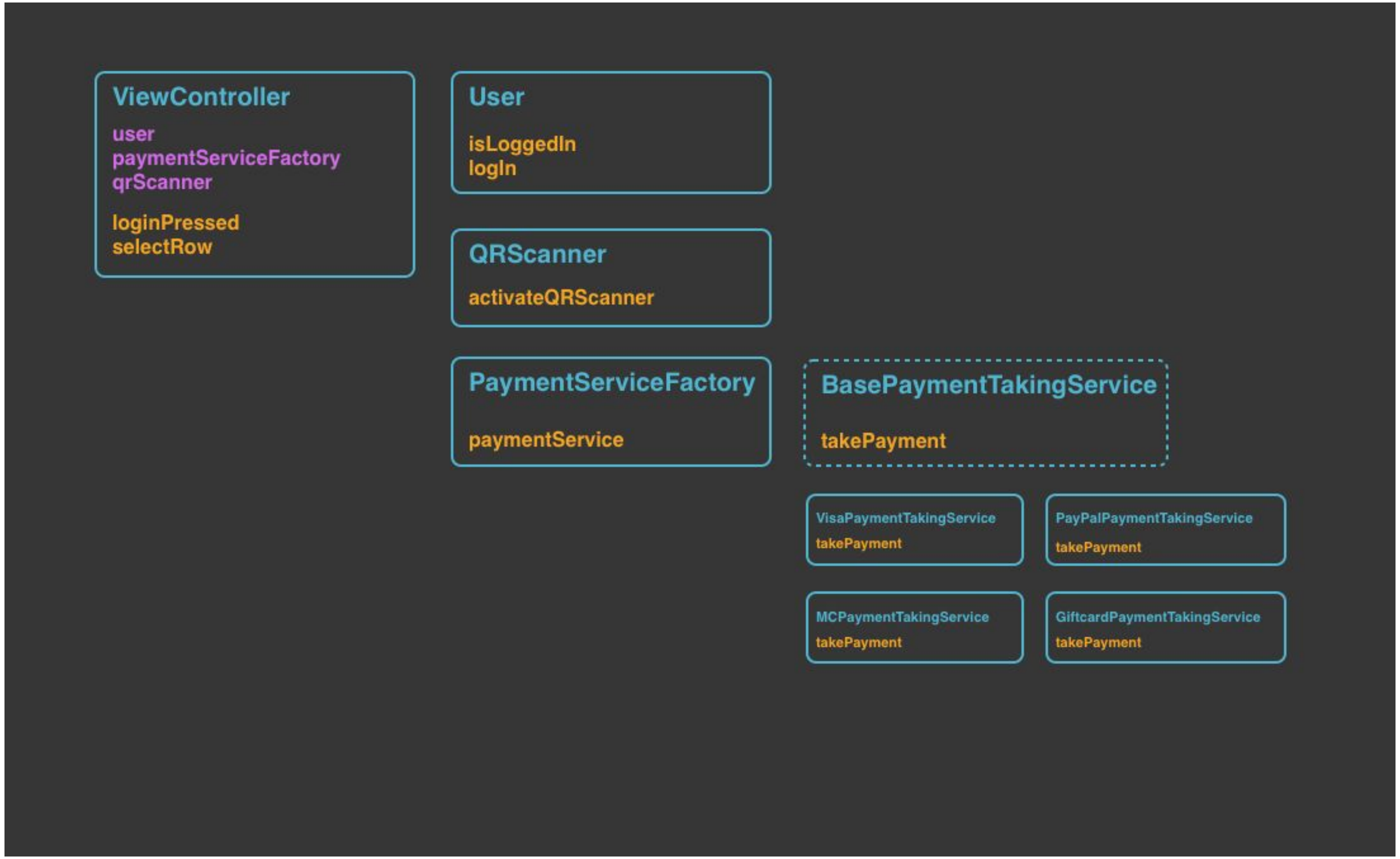
Open/Close Principle & Liskov Substitution Principle



Interface Segregation Principle

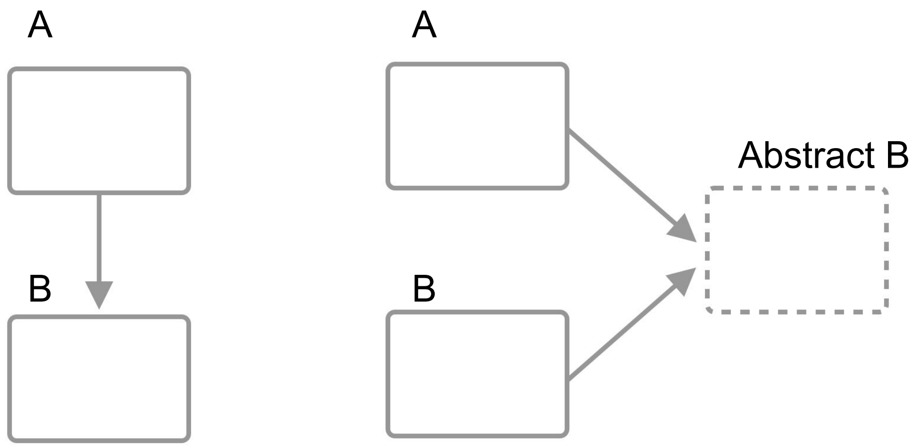
No client should be forced to depend on methods it does not use

Interface Segregation Principle

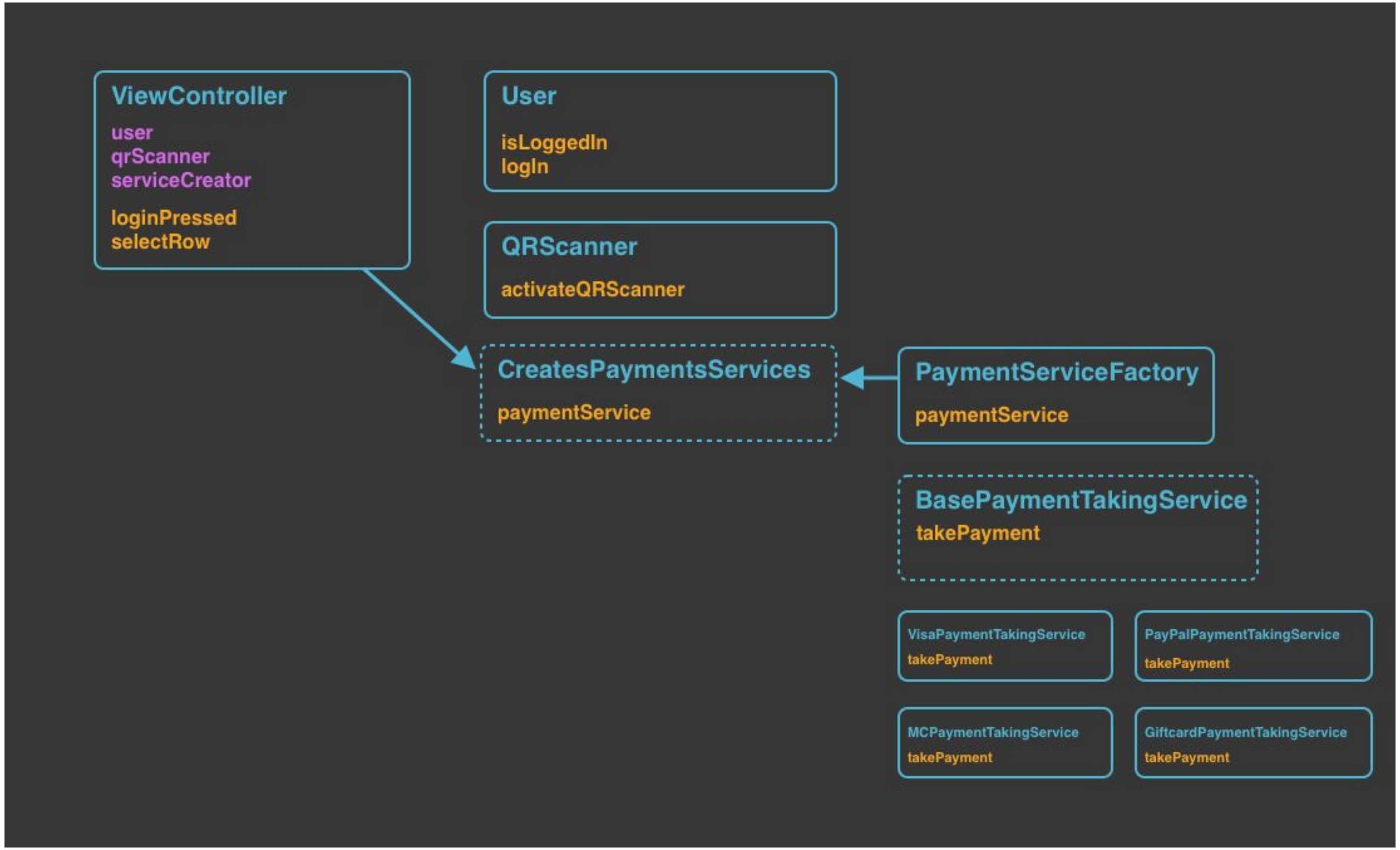


Dependency Inversion Principle

Depend on abstractions, not concrete objects



Dependency Inversion Principle



Takeaways and conclusion

- Remember that SOLID is a guide, not a doctrine.
- The goal is to break down code into constituent parts.
- Less code > More Code. More Classes > Fewer Classes.
- The thing to avoid: fragmentation and over-engineering.

Thanks!

Many thanks to Dragos Ionel and the Mobile Geeks audience.

Happy coding