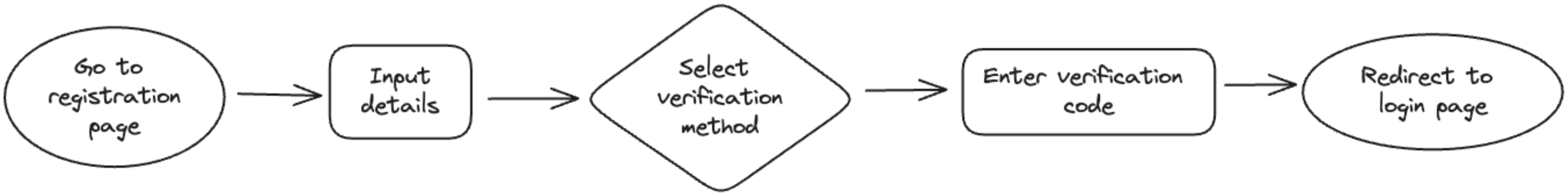
DWARVES
FOUNDATION

Template method

Behavioral design pattern

Problem

Problem



Go to registration page

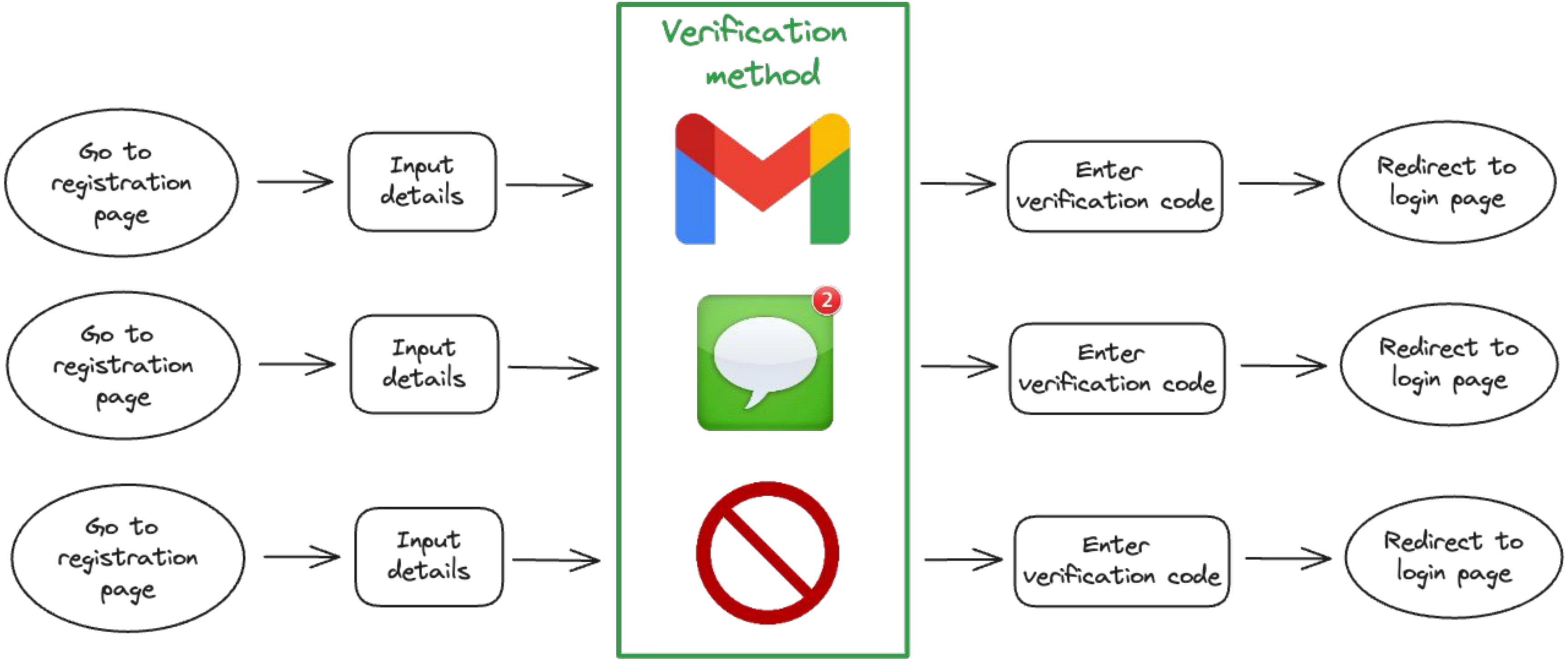
Input details

Select verification method

Enter verification code

Redirect to login page

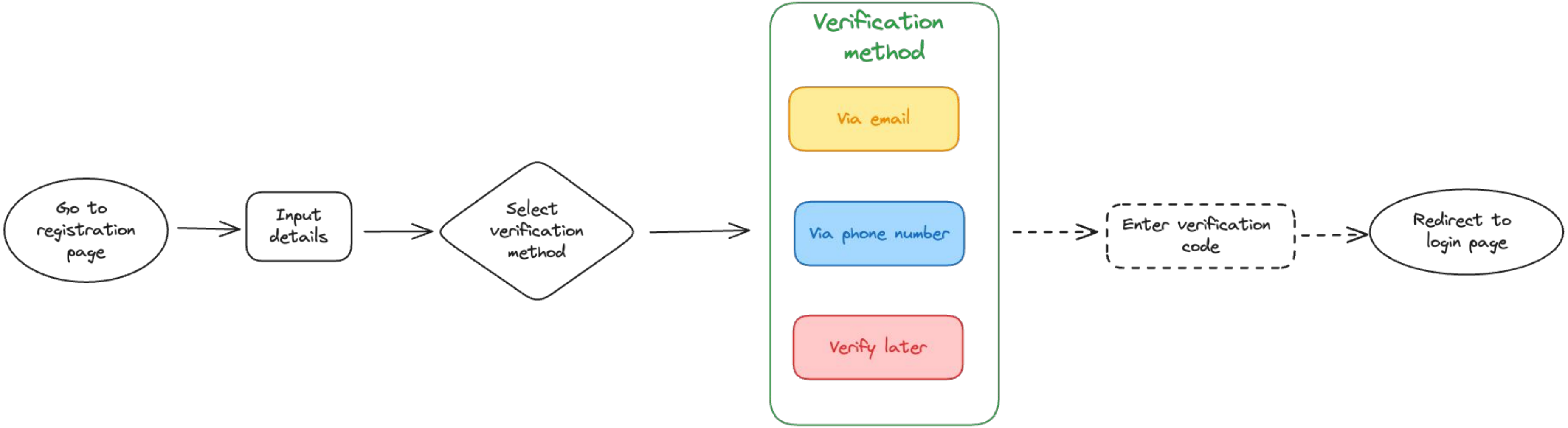
Problem



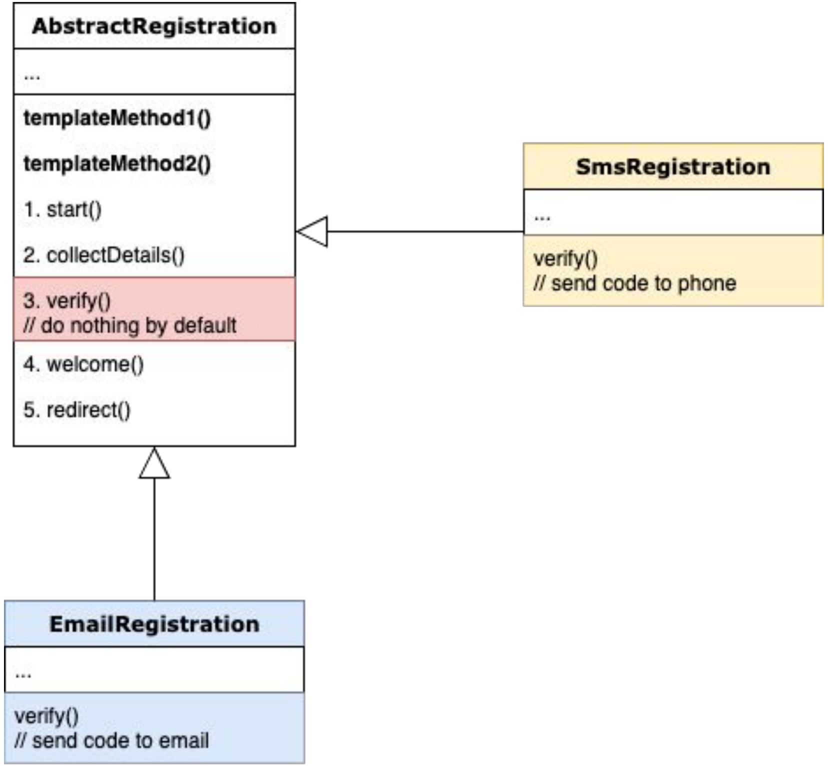
DWARVES FOUNDATION

Solution

Solution


Go to registration page
Input details
Select verification method
Verification method
Via email
Via phone number
Verify later
Enter verification code
Redirect to login page

Structure



AbstractRegistration

...

templateMethod1()

templateMethod2()

1. start()

2. collectDetails()

3. verify()
// do nothing by default

4. welcome()

5. redirect()

SmsRegistration

...

verify()
// send code to phone

EmailRegistration

...

verify()
// send code to email

DWARVES
FOUNDATION

Pros & Cons

Pseudocode

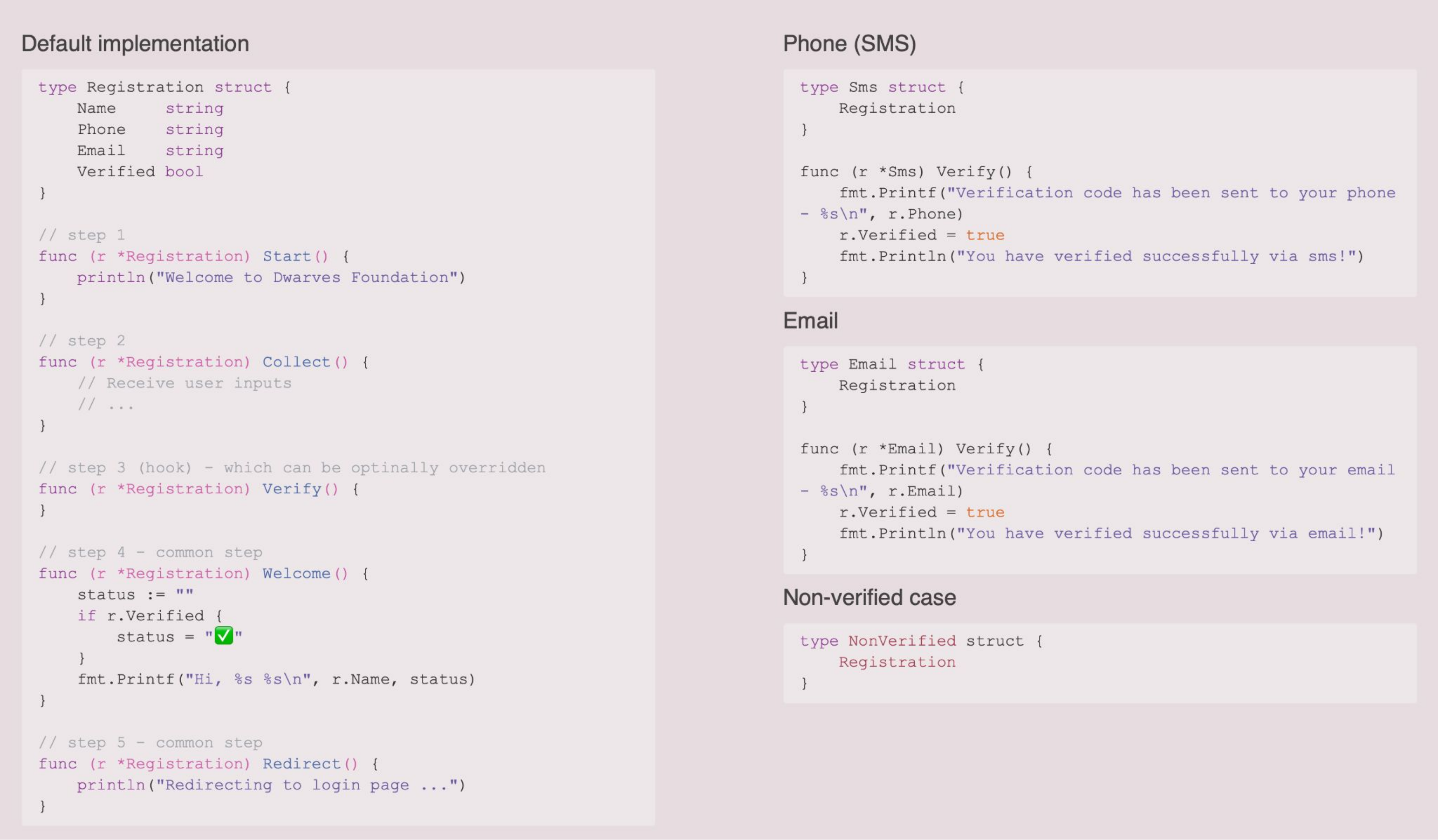
Golang example

interface

```go
type IRegistration interface {
    Start()
    Collect()
    Verify()
    Welcome()
    Redirect()
}
```

template method

```go
func Register(r IRegistration) {
    r.Start()
    r.Collect()
    r.Verify()
    r.Welcome()
    r.Redirect()
}
```



Default implementation

```go
type Registration struct {
    Name     string
    Phone    string
    Email    string
    Verified bool
}

// step 1
func (r *Registration) Start() {
    println("Welcome to Dwarves Foundation")
}

// step 2
func (r *Registration) Collect() {
    // Receive user inputs
    // ...
}

// step 3 (hook) - which can be optinally overridden
func (r *Registration) Verify() {
}

// step 4 - common step
func (r *Registration) Welcome() {
    status := ""
    if r.Verified {
        status = "✅"
    }
    fmt.Printf("Hi, %s %s\n", r.Name, status)
}

// step 5 - common step
func (r *Registration) Redirect() {
    println("Redirecting to login page ...")
}
```

Phone (SMS)

```go
type Sms struct {
    Registration
}

func (r *Sms) Verify() {
    fmt.Printf("Verification code has been sent to your phone - %s\n", r.Phone)
    r.Verified = true
    fmt.Println("You have verified successfully via sms!")
}
```

Email

```go
type Email struct {
    Registration
}

func (r *Email) Verify() {
    fmt.Printf("Verification code has been sent to your email - %s\n", r.Email)
    r.Verified = true
    fmt.Println("You have verified successfully via email!")
}
```

Non-verified case

```go
type NonVerified struct {
    Registration
}
```

Client

```
func main() {
    var r registration.IRegistration
    switch device {
    case DESKTOP:
        r = &registration.Email{}
    case PHONE:
        r = &registration.Sms{}
    default:
        r = &registration.NonVerified{}
    }

    registration.Register(r)
}
```

Thank you

DWARVES
FOUNDATION