



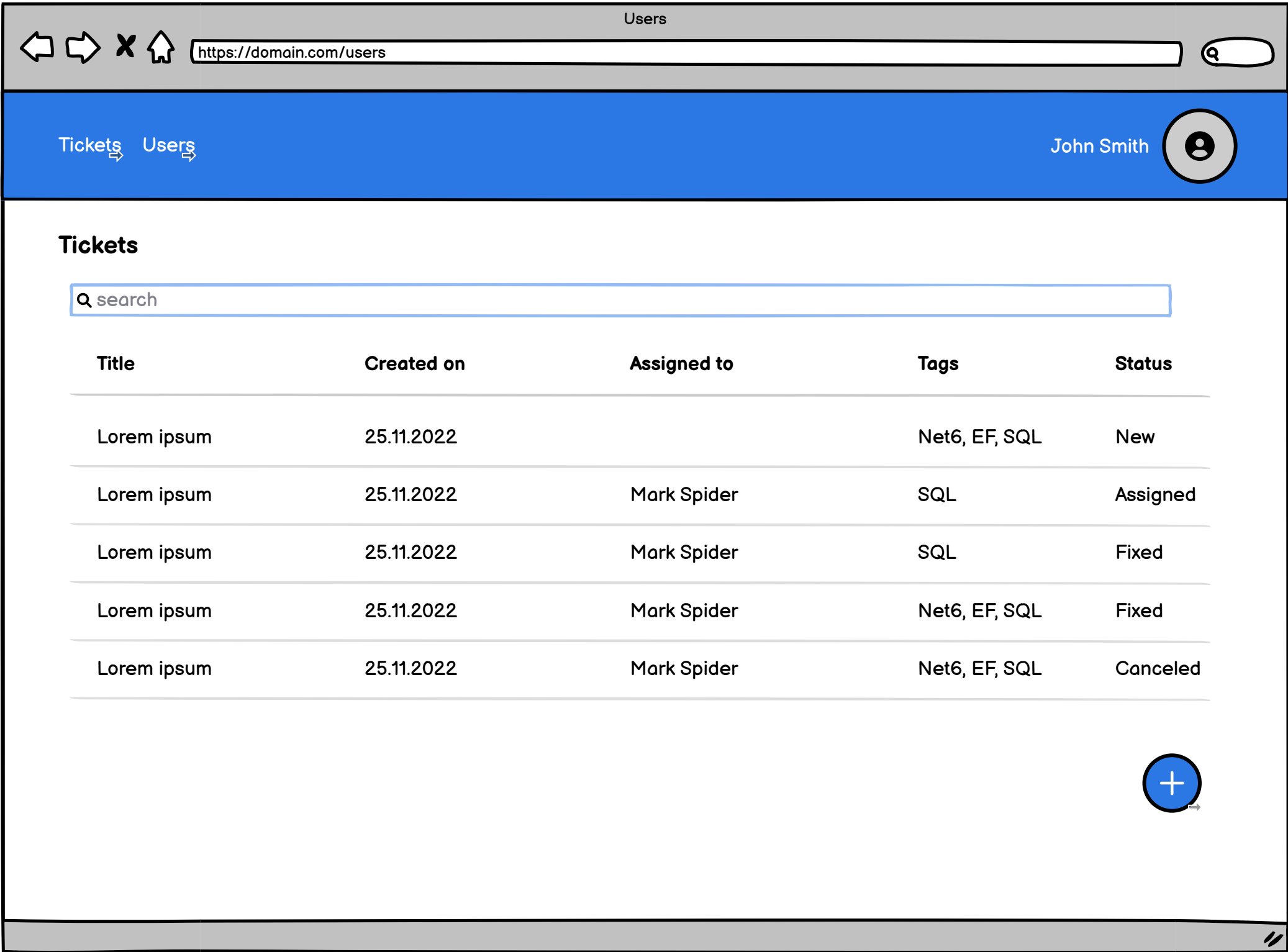
https://domain.com/users



john



Login





[Tickets](#) [Users](#)

John Smith



Add Ticket

Title

John

Description

Assigned to

Select user

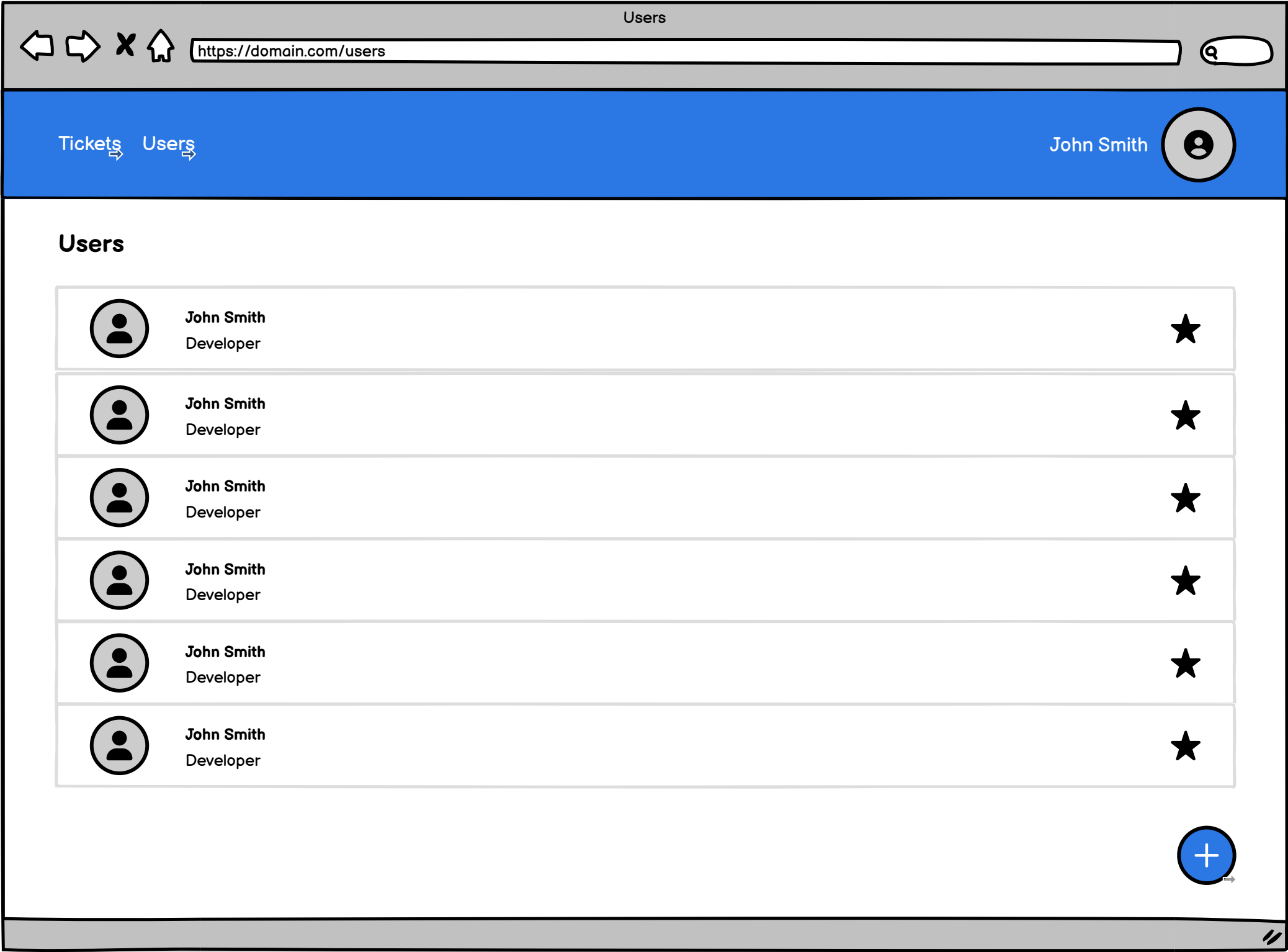


Tags

NET6, SQL, EF

Save





[Tickets](#) [Users](#)

John Smith



Add User

First Name

John

Last Name

Smith

Password

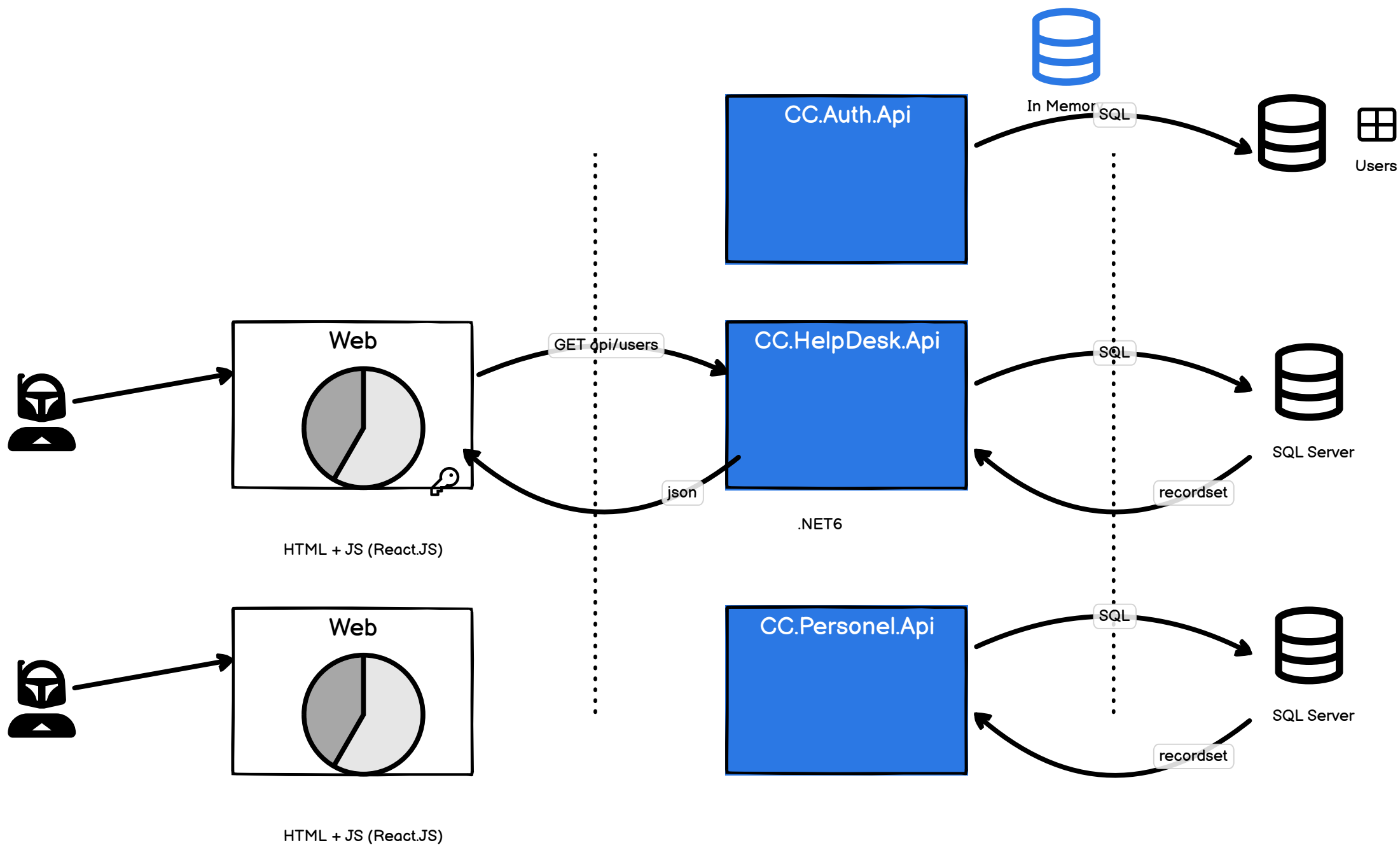
Email

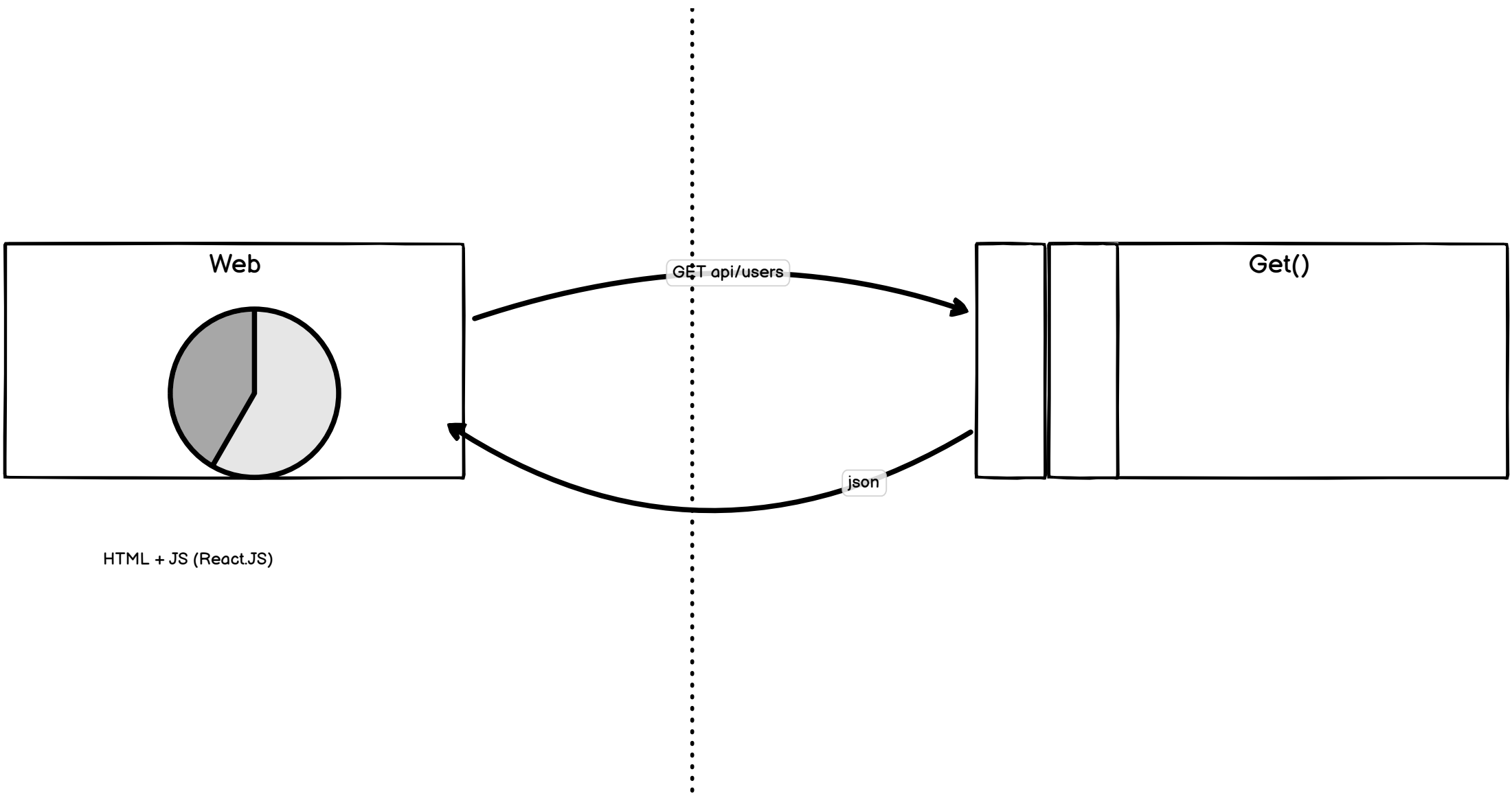
john.smith@domain.com

Role

☒ Developer☐ Lead☐ Manager☐ Send notifications

Save





CC.HelpDesk.App

CC.HelpDesk.Api

CC.HelpDesk.IRepositories

CC.HelpDesk.DbRepositories

CC.HelpDesk.InMemoryRepositor
ies

CC.HelpDesk.Domain

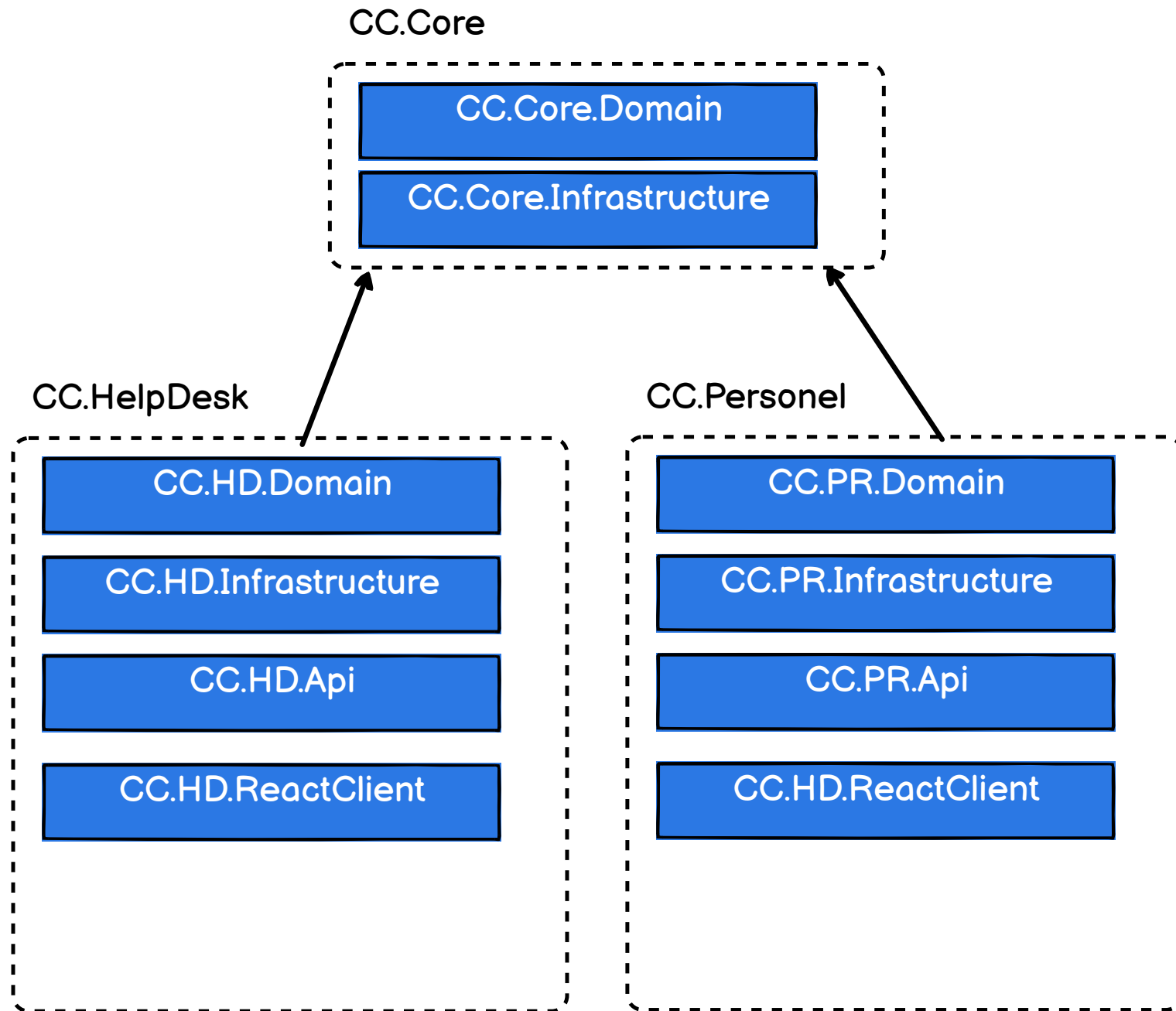
CC.HelpDesk.Domain

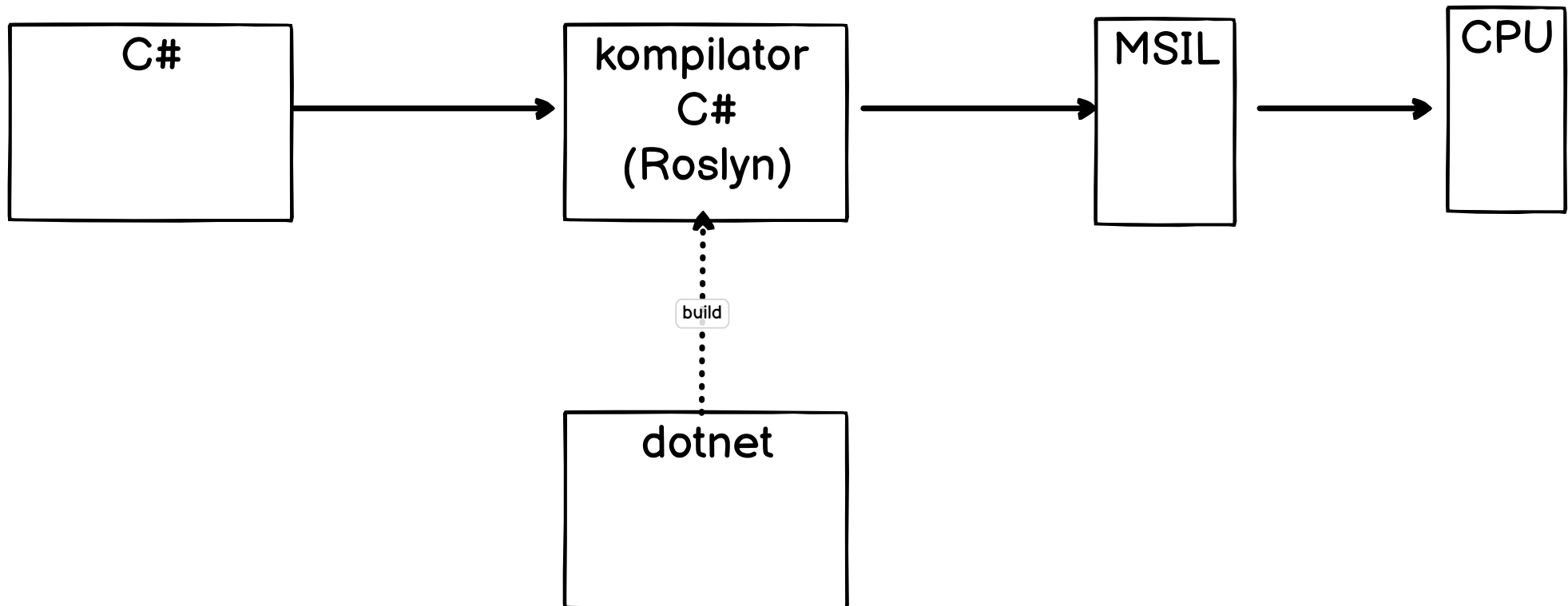
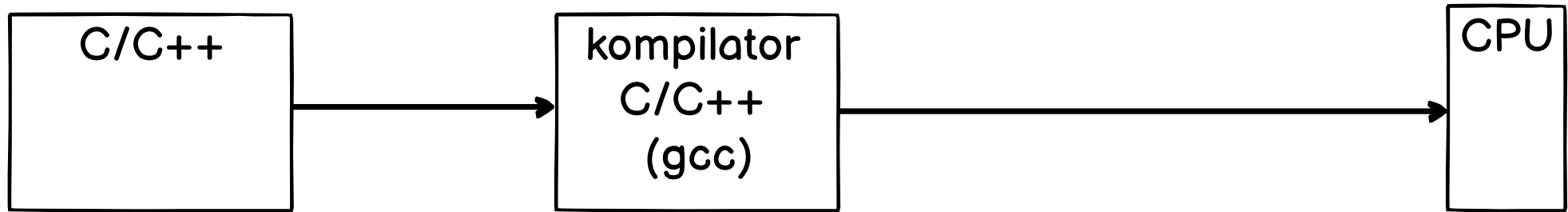
CC.HelpDesk.IRepositories

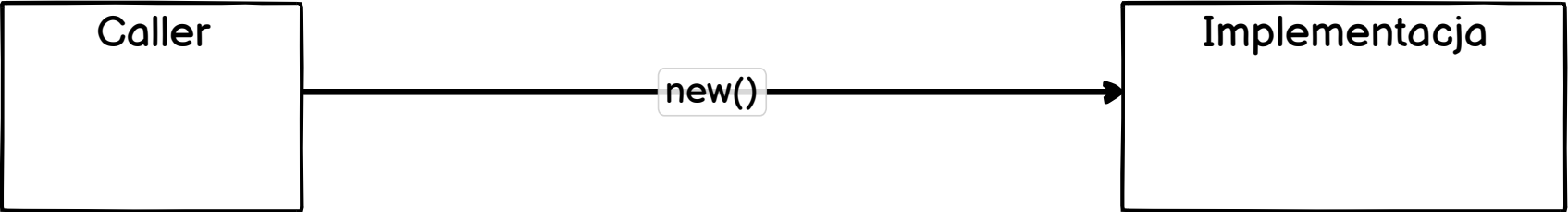
CC.HelpDesk.InMemoryRepositories

CC.HelpDesk.Api

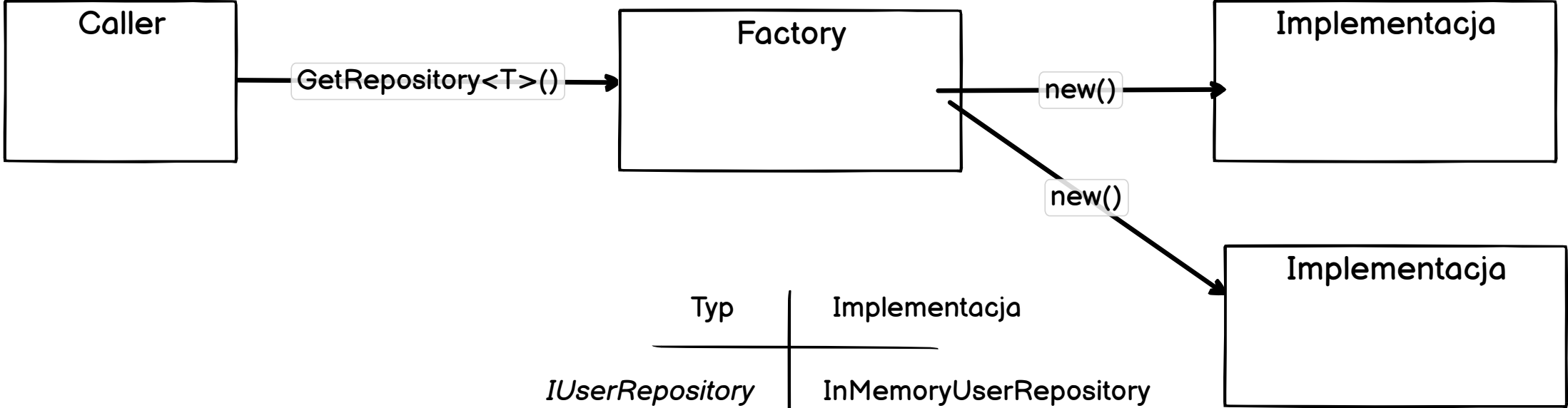




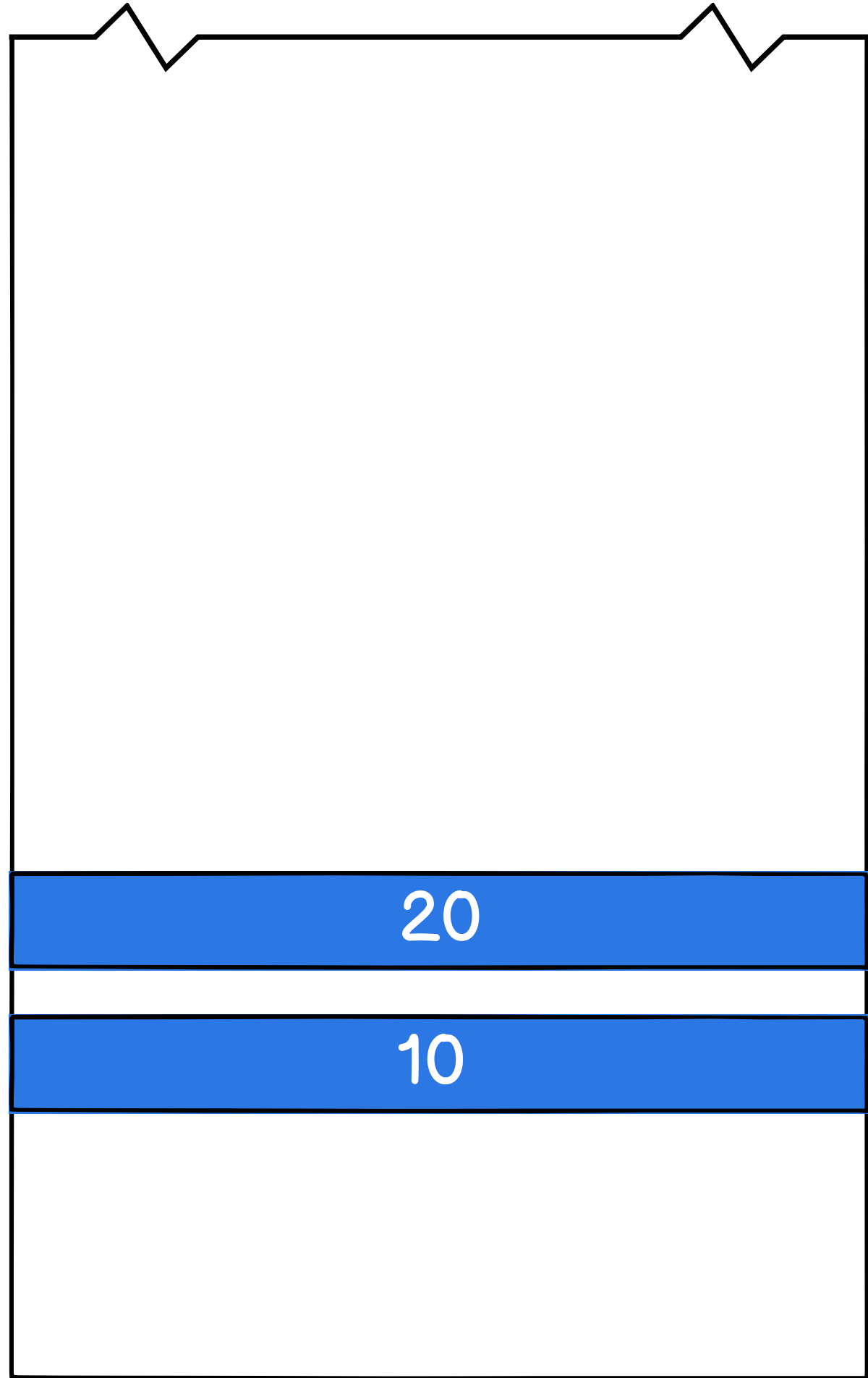




.....



Typ	Implementacja
<i>IUserRepository</i>	InMemoryUserRepository
<i>IUserRepository</i>	DbUserRepository



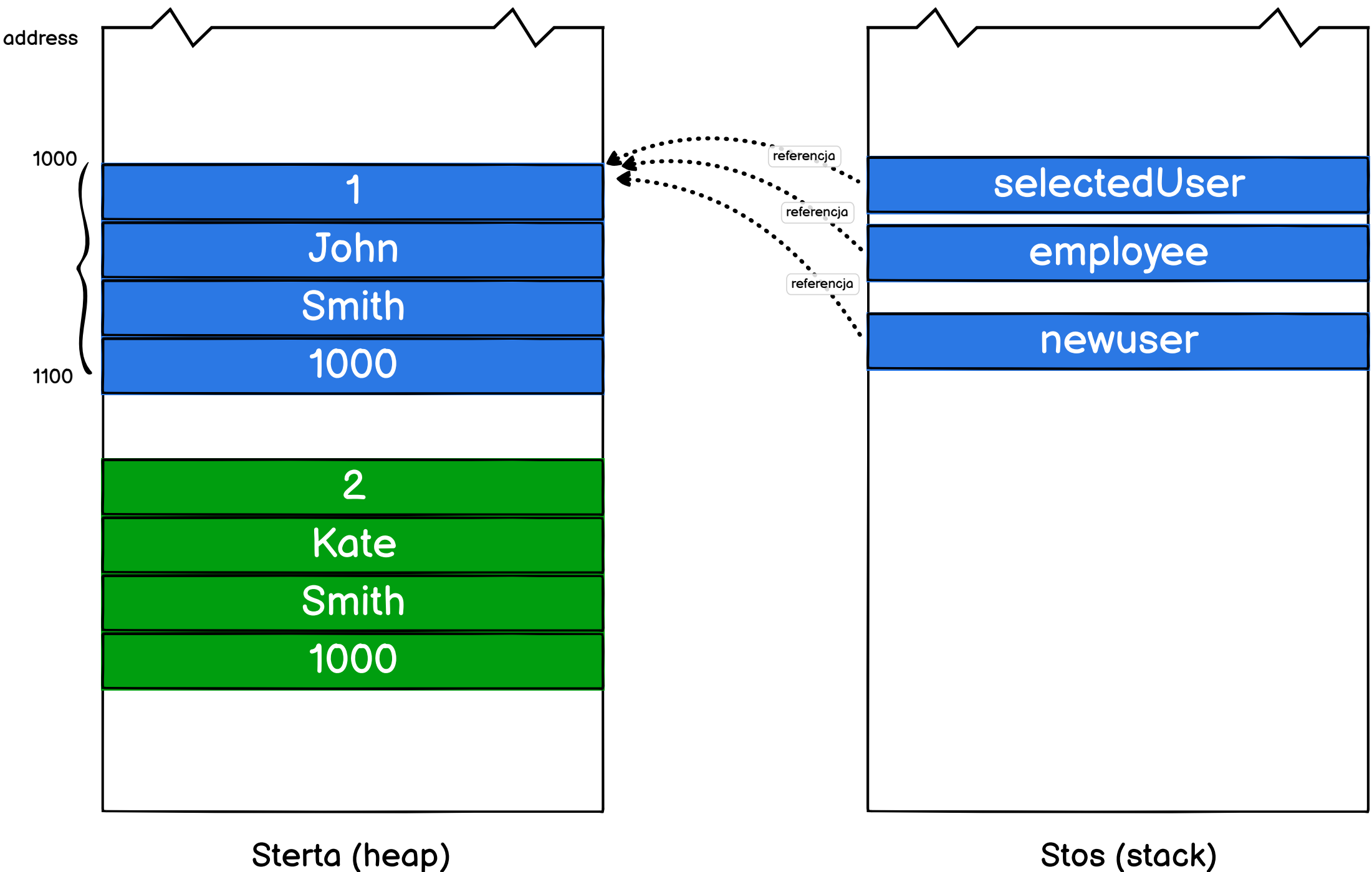
20

amount'

10

amount

Stos (stack)



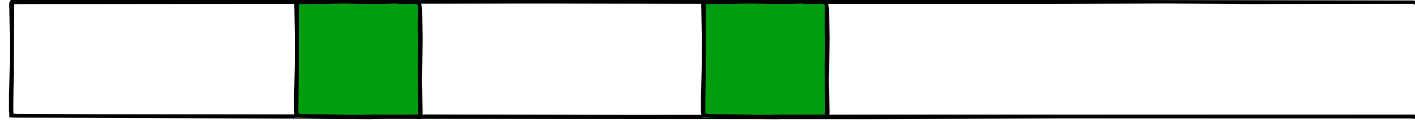
Url

Table

CurrencyCode

SeqUrl

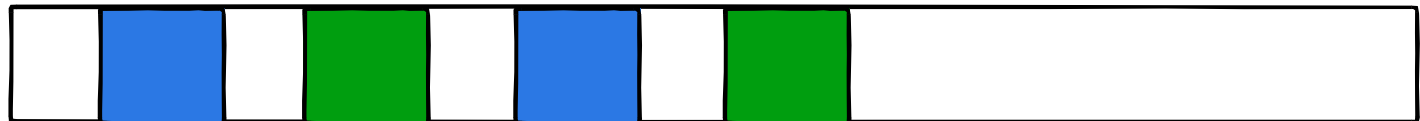
appsettings.development.json

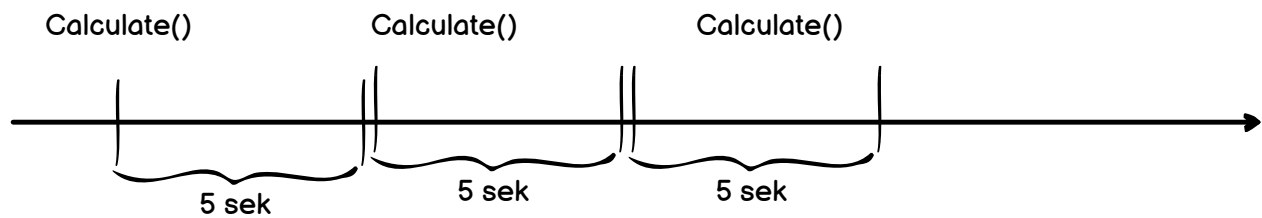


appsettings.json

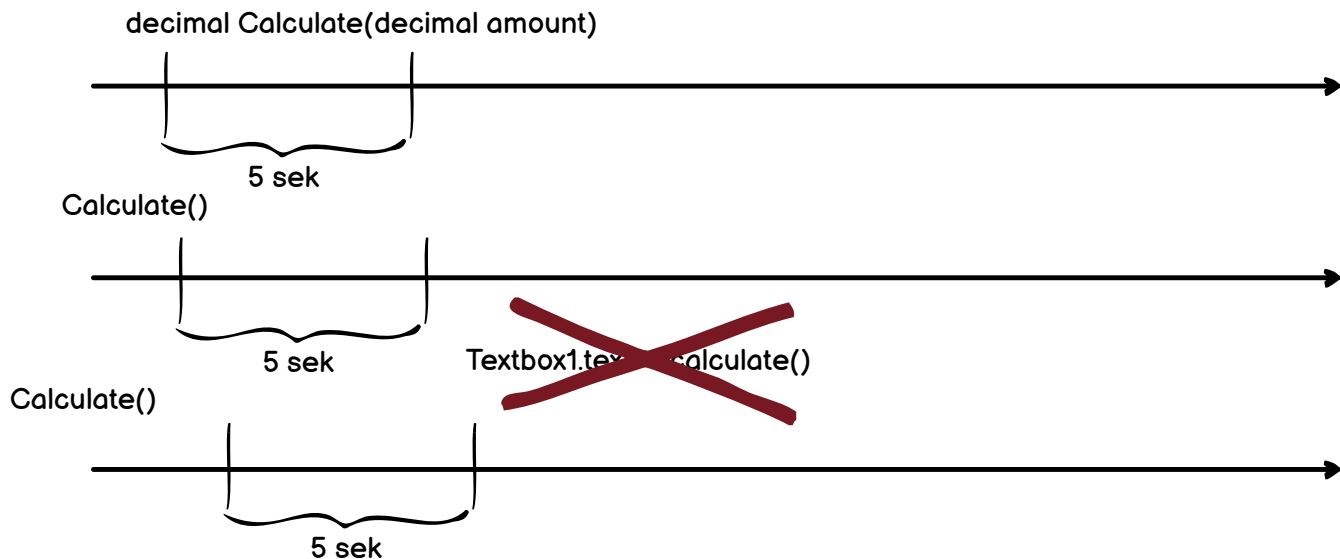


builder.Configuration[key]





.....



.....

```
Task(()=>Calculate(1)).ContinueWith(()=>Send("Twoje saldo wynosi ...."));
Task(()=>Calculate(2)).ContinueWith(()=>Send("Twoje saldo wynosi ...."));
```

```
Decimal result = Calculate(3);
Send($"Twoje saldo wynosi {result}");
```

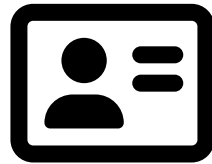
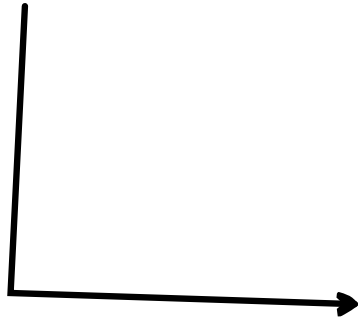
```
Task(()=>Calculate(3))
    .ContinueWith(t=>Send($"Twoje saldo wynosi {result}"));
```

```
Decimal result = await Calculate(3);
Send($"Twoje saldo wynosi {result}");
```



IPrincipal

`IIdentity Identity { get; }`



IIdentity

`String Name { get; }`

`AuthenticationType { get; }`

`bool IsAuthenticated { get; }`

`ClaimType`

`ClaimValue`

Kat

B

Kat

A

Phone

555-000-111

Role

Developer

Role

Consultant