

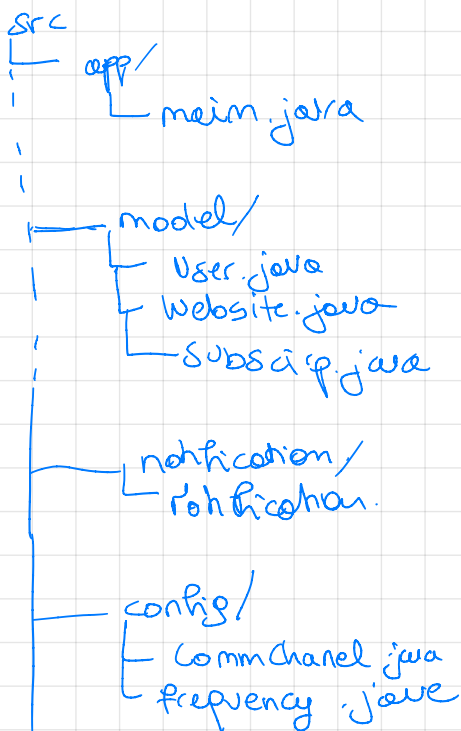
Aufgabe 2.

	EC (outgoing)	AC (Incoming)	I
User	Subscriptions	Sub, Not	0,33
→ Website	-	Subs	0
Subscriptions	User, Website, Frequency, Commman.	User,	0,8
Communication Chanel	-	Subs, Notfic	1
→ Frequency	-		0
→ Notification	User, Communication Chanel	-	0

Website, User are stable, well designed data or configuration object

Subs, Notification have more dependencie which makes them sensitive to change

Aufgabe 3



→ main logic / entry point

→ Data objects. (entities)

→ Sending Notification

→ Enumerations or static configuration

Aufgabe 4

- ⇒ use interfaces - to allow flexibility and easier substitution of different implementations
- ⇒ group by Responsibility ~~not~~ by type ⇒ this avoids packages that depend on many others just to assemble functionality
- ⇒ dependency injection : inject dependencies