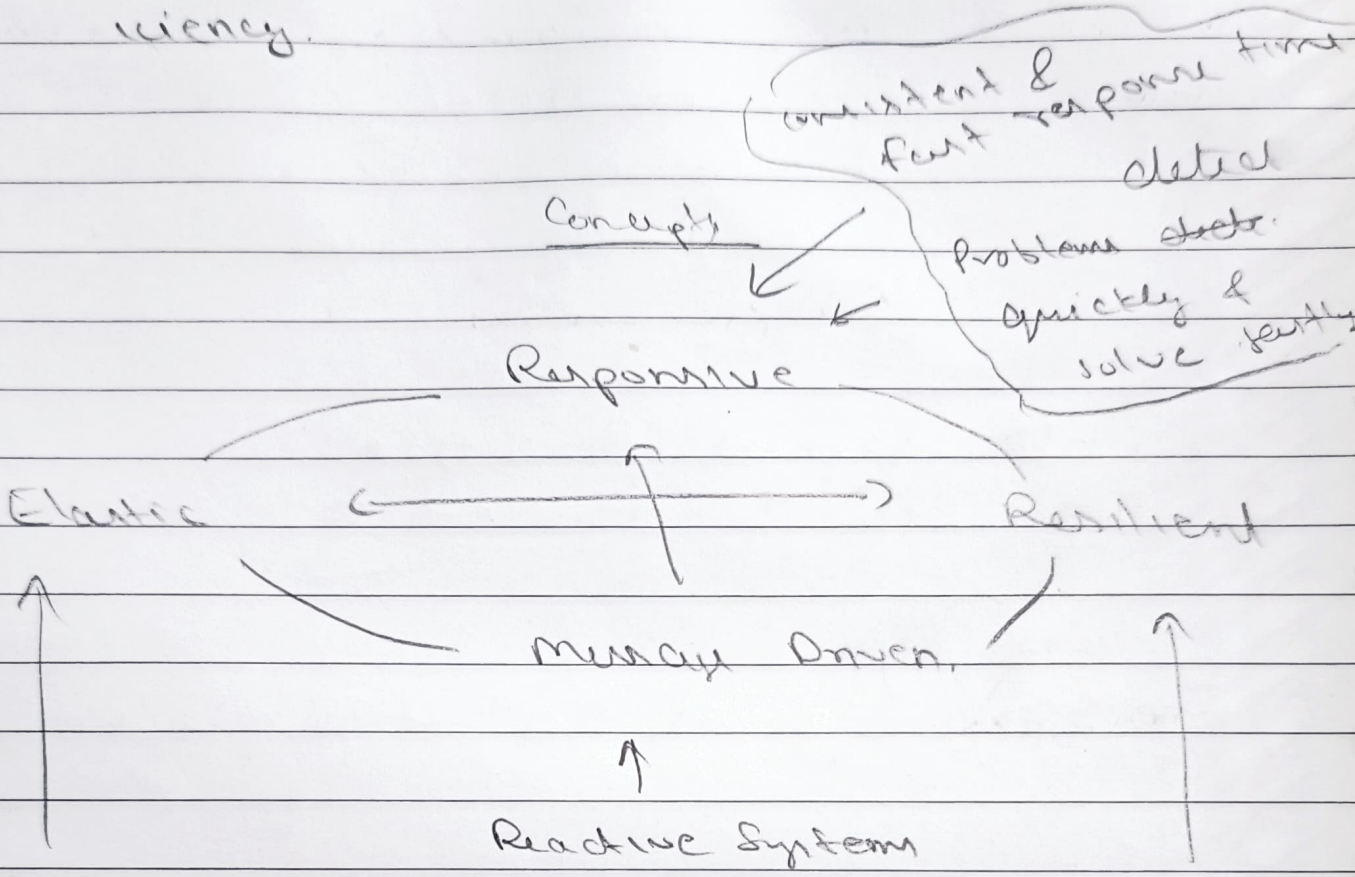
Dt. _____ Pg. _____

→ Reactive can improve computing efficiency.



Dt. _____ Pg. _____

Reactive Programming

→ is a way of implementation technique.

→ It focus on non-blocking, Asynchronous execution

It is Asynchronous Programming Paradigm focused on streams of data

Features of Reactive Programming

Data Streams -

Asynchronous

Non-blocking

Back pressure

Failure as Messages

Events are captured Asynchronously

Dt. _____ Pg. _____

→ A function is defined to execute when an event is emitted

→ Another function is defined if an error is emitted.

→ Another function is defined when complete is emitted.

Back pressure

→ The ability of the subscribe to throttle data. (handled by the libraries we use)

Failures as messages

→ Exceptions are not thrown in a traditional sense.
- Would break process of stream
- Exceptions are processed by a handler function. (a function to handle the failure)

→ We have a method where we assign to a function to handle any type of errors or exceptions that occur

Dt. _____ Pg. _____

5 Things To Know

- IDE / eclipse.
-
- Basics of language
- DATA STRUCTURES (Lists & Maps)
- Testing

Reactive Streams API

↓

It is set of 4 Interfaces that are going to define the API

- Publisher interface
  - → will publish the events
  - ↳ it defines a simple method i.e subscribe

← will subscribe the events via a subscription so we can see how the is working

→ Subscriber Interface

↳ the subscription is going to manage the back pressure

Dt. _____ Pg. _____

It gives us

- onSubscribe()
- onNext()
- onError()
- onComplete()

3 → Subscription Interface — The subscription is going to manage the back pressure

- request()
- cancel()

— the subscription

4 → Processor <T, R> extends Subscriber<T>, Publisher<T>

* A *

- publishers publish the events
- the subscribers gets us events via the subscription

@Builder ← annotation

Dt. _____ Pg. _____

Spring Reactive Types

↓

new reactive types are introduced with Spring Framework 5.

- Mono
  - ↳ is a publisher with zero or one elements in data stream
- Flux
  - ↳ is a publisher with zero or MANY elements in the data stream.

- Both types implements the Reactive Streams Publisher interface

Dt. _____ Pg _____

Mono.just(object)

↑
adding data

Flux.just(michel, sara, sam);

↑
adding multiple data

⇒ why we block ✗ not preffered. to get data.
to get the object from mono. stream.

→ More prefer subscribe()

&

" " map()

↑
map do transformation
here we have to apply
back pressure to
trigger mapping op"
so we use subscribe()
it map it down to the