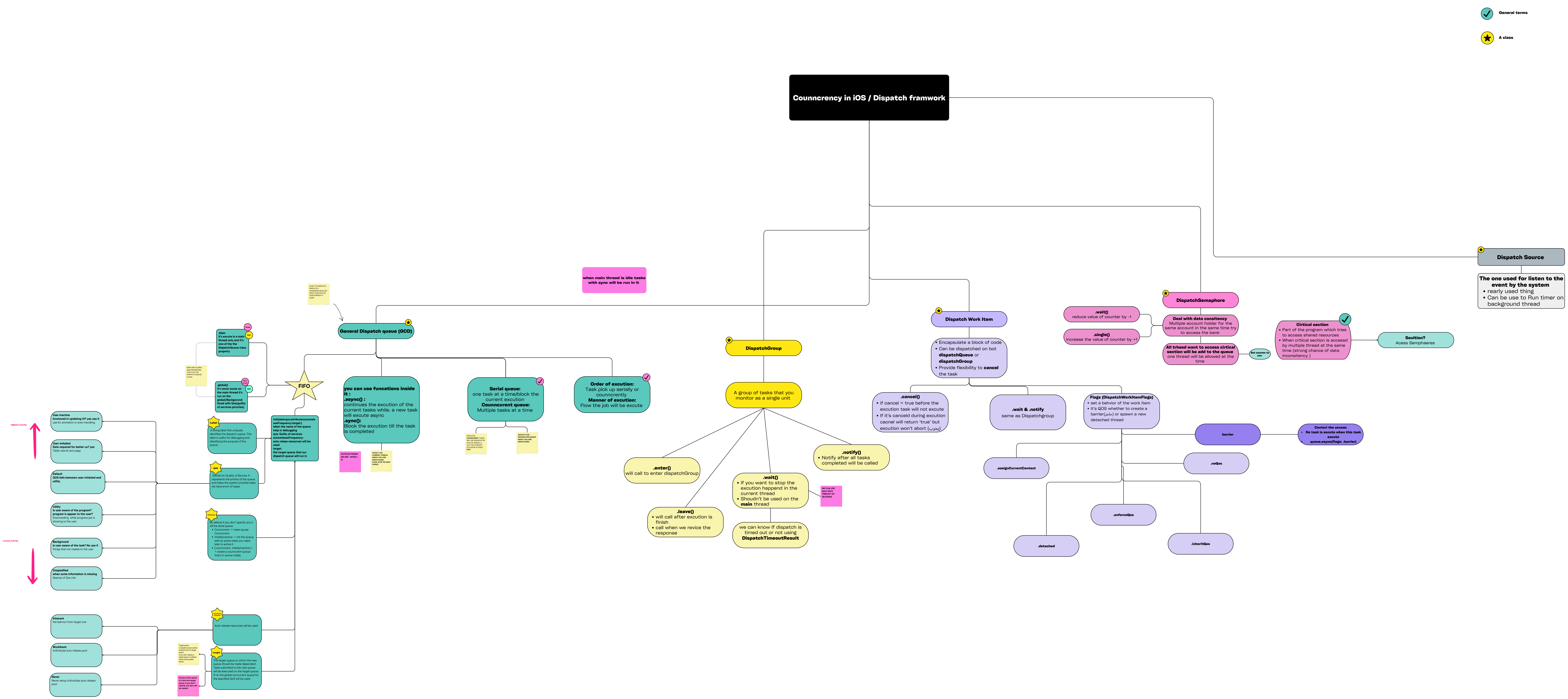

General terms
A class
Councrency in iOS / Dispatch framwork
ALSO IT'S DISPATCH WHICH IS A FRAMWORK BUILD BY APPLE FOR EXCUTE CONCURRENTLY CODE
General Dispatch queue (GCD)
when main thread is idle tasks with sync will be run in it
Serial
Main
.Main it's excute in a main thread only and it's one of the the DispatchQueue class property
MAIN AND GLOBAL ARE PROPERTIES THAT'S IN THE DISPATCH QUEUE CLASS
Councrent
QOS
.global() it's never exute on the main thread it's run on the global/Backgorund thrad with Qos(quility of services pirorties)
FIFO
you can use funcations inside it :
.async() :
continues the excution of the current tasks while. a new task will excute async
.sync():
Block the excution till the task is completed
ASYNCAFTER(DEADLINE: .NOW() + 3)
AFFECT THE CURRENT THREAD WHICH YOU ARE DISPATCHING [TASK STOP OR KEEP GOING]
Serial queue:
one task at a time/block the current excution
Councurent queue:
Multiple tasks at a time
EVEN FOR COUNNCRENT TASKS WILL BE DEQUEUE OR EXCUTE SERIALLY CUZ THE CONCEPT OF QUEUE IS FIXED FIFO
AFFECTS THE DESIGNATION QUEUE WHICH YOU ARE DISPATCHING
Order of excution:
Task pick up serially or councrently
Manner of excution:
Flow the job will be excute
init(label:qos:attributes:autoreleaseFrequency:target:)
label: the name of the queue help in debugging
qos: Quilty of services
autoreleaseFrequency:
auto relase resources will be used
target:
the target queue that our dispatch queue will run in
Label
A string label that uniquely identifies the dispatch queue. This label is useful for debugging and identifying the purpose of the queue.
qos
Stands for Quality of Service. It represents the priority of the queue and helps the system prioritize tasks we have enum of types
Attributes
By default if you don't specify any it will be serial queue
Councrent -> make quuee Councrent
initiallyInactive -> init the queue with an active state you need later to active it
[.councrent, .initiallyInactive ] -> create a councrent queue that's in-active initally
Highest priority
User inactive
Involoved in updating UI? yes use it
use for animation or even handling
User-initailed
Data required for better ux? yes
Table view & next page
Default
QOS falls between user-initatied and utility
Utility
Is user aware of the program? program is appear to the user?
Downloading while progress par is showing to the user
Lowest priority
Background
is user aware of the task? No use it
things that not visable to the user
Unspesified
when some information is missing
Abense of Qos info
autorelease Frequency
Auto release resources will be used
Inherent
the behvior from target one
WorkItemt
Individulas auto relases pool
Never
Never setup Individulas auto relases pool
Target queue
A dispatch queue priority inherent from it's target queue
If you don't specify a target queue it's default queue priority globe queue
Pirorty of the queue it's form the target queue if you don't specify any Qos will be deafult
target
The target queue on which the new queue should be made dependent. Tasks submitted to the new queue will be executed on the target queue. If nil, the global concurrent queue for the specified QoS will be used.
DispatchGroup
A group of tasks that you monitor as a single unit
.enter()
will call to enter dispatchGroup
.leave()
will call after excution is finish
call when we revice the response
.wait()
If you want to stop the excution happend in the current thread
Shoudn't be used on the main thread
WE CAN USE WAIT WITH TIMEOUT OF SECONDS
we can know If dispatch is timed out or not using DispatchTimeoutResult
.notify()
Notify after all tasks completed will be called
Dispatch Work Item
Encapsulate a block of code
Can be dispatched on bot dispatchQueue or dispatchGroup
Provide flexibility to cancel the task
.cancel()
if cancel = true before the excution task will not excute
If it's canceld during excution cacnel will return 'true' but excution won't abort (يخرب)
.wait & .notify
same as Dispatchgroup
Flags (DispatchWorkItemFlags)
set a behvior of the work item
It's QOS whether to create a barrier(حاجز) or spawn a new detached thread
barrier
Contorl the access
No task is excute when this task excute
queue.async(flags: .barrier)
.assignCurrentContext
.noQos
.enforceQos
.detached
.inheritQos
DispatchSemaphore
.wait()
reduce value of counter by -1
.single()
increase the value of counter by +1
Deal with data consitency
Multiple account holder for the same account in the same time try to access the bank
All trhead want to access cirtical section will be add to the queue
one thread will be allowed at the time
Set counter to one
Cirtical section
Part of the program which tries to access shared resources
When crtiical section is accessd by multiple thread at the same time (strong chance of data inconsitency )
Soultion?
Acess Semphaeres
Dispatch Source
The one used for listen to the event by the system
rearly used thing
Can be use to Run timer on background thread