COMP 4958: Lab 5

Submit a zip file named card.zip containing everything (except the `_build` directory) in the `card` directory created using `mix`. Maximum score: 12

For this lab, you are asked to implement an application that uses a (supervised) partition supervisor that creates dynamic supervisers used to dynamically create "card workers" from your previous labs. Each card worker created by a dynamic supervisor needs to retain its state after a crash. Use an ETS table named `Card.Store` to store the state. Put the code for the dynamic supervisor in a module named `Card.WorkerSupervisor`.

A dynamic supervisor provides a `start_worker` function that takes the name of a card worker to start. Each card worker is registered with a process registry. (Name this registry `Card.Registry`.) `Card.Worker` needs to provide the usual functions in its client API: `new(name)`, `shuffle(name)`, `count(name)` and `deal(name, n \\ 1)` (besides `start_link/1` which is invoked by a dynamic supervisor).

Make sure that `deal(name, n)` causes a crash if `n` is not integer, so that we can use something like `Card.Worker.deal("worker1", :die)` to cause a "restart" of a worker to check that it retains its state. Furthermore, a card worker must print a message that includes its PID when it starts/restarts.

The application supervisor (in `Counter.Application`) needs to create an ETS table, start the process registry as well as the partition supervisor. For this exercise, use 4 partitions. This means that the partition supervisor will start 4 dynamic supervisors when the application starts. Make sure that each dynamic supervisor prints a message that includes its PID when it starts.