**Author: Mohamed Shafee**

Design Document

**Abstract**

I solved this assignment by creating 2 simple functions. One starts the application and the other holds the code that each thread executes.

**API**

execute(pid\_manager):

Holds the code that each thread executes. Each thread requests a pid number, sleeps for a random period of time, and releases the pid allowing another thread to pick it up. I decided to pause each thread for a random period of time before a pid is requested so that another thread has the chance to pick it up after its release. Without this, all pid numbers would be allocated in order (as the threads run concurrently) then deallocated randomly based on how long each thread sleeps.

def start():

Start the application. Creates an instance of the PidManager object, allocates a list, and creates a thread pool with a maximum of 50 workers. Each thread evokes the execute function concurrently.