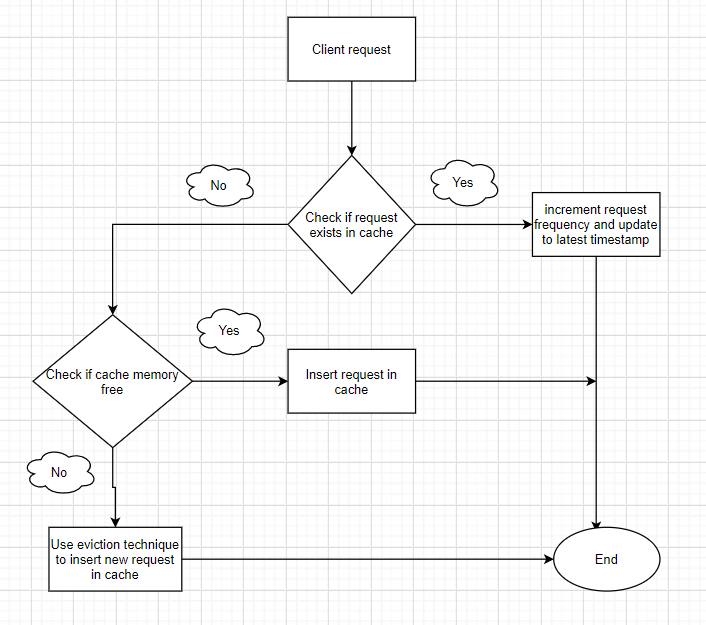
**Cache library:**

**Introduction:**

Cache is a temporary storage area that stores frequently accessed items/records. This is designed to overcome latency problems when accessing data from backend layers each and every time for same request.

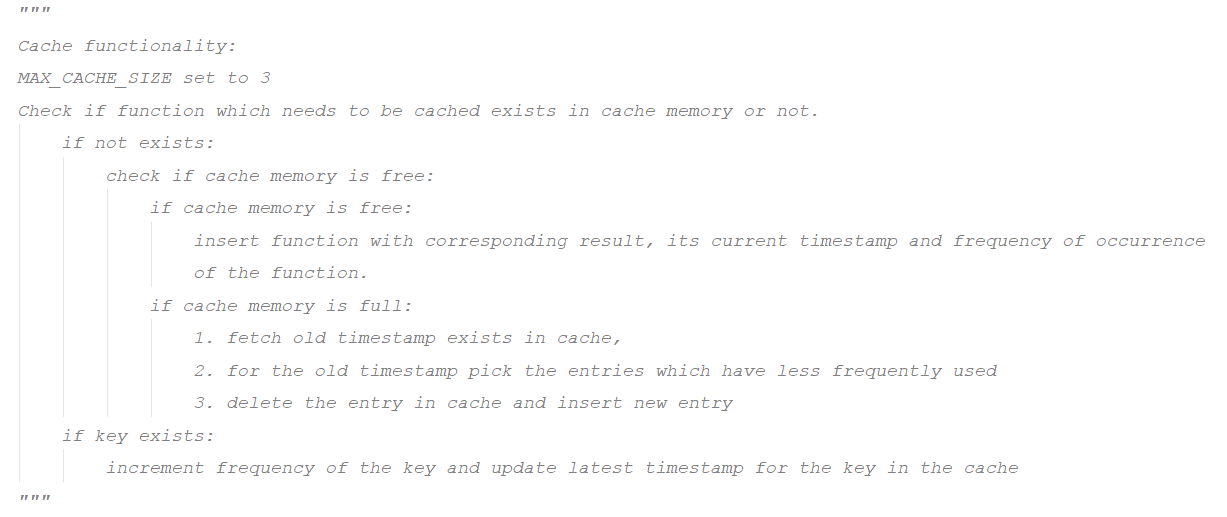
**Flow chart of cache library:**



Cache area

**Cache Implementation Details:**

1. Implemented design using list and dictionary.
2. It is an in-process cache where application instantiates, manages and runs its own cache
3. Cache eviction is based on entry/function/record age and its frequency.
4. Before adding entry into cache, below is the design flow process.



1. Cache operations include add entry, update entry value, delete entry, clear entries and get entry value.
2. write-through implementation where write to database happens through cache.
3. More the frequent records, more the cache hit rate

**Trade-off:**

1. This cache library gives priority to new tasks during cache eviction.
2. Every time cache will have updated data, so there is no problem of cache invalidation but there is write overhead as it will write to cache and also database every time.
3. Less frequent records have more chances of cache miss which affects the latency.