**Coding Exercise No 1.**

Consider this product requirement statement:

We wish to build a multi-user prioritized task list service.  This service should allow for a relatively large number of independent lists (scale to millions) of tasks (thousands per list).  Each list should support having thousands of items and allow for a user prioritized ordering of items where an item can be just a user presentation string for this exercise.  The prioritized ordering is common between all users that have access to the list.

The service needs to provide an HTTP based API to create and mutate these lists.  This API needs to support the notion of multiple thick clients (e.g. iOS and Android mobile phones and single-page web applications) where the applications themselves will desire to perform local cached copies of the lists and maintain near-real-time view of the lists when the devices are online as well as ability to present and locally mutate the lists while they are offline.  The client applications will use background synchronization via this designed API and the API must support multiple users attempting to mutate the lists at overlapping points in time with some method of consistency consensus that attempts to avoid users in offline states to lose their changes and start over.

Statement of project - Client:

Design the API, design a data model, and write the client in a language of choice. Using AWS cloud native techniques, build a deployment model for the service to support the load above including some support for scaling cost to load. Explicitly design the sequencing of events. Focus specifically on how synchronization is achieved in certain exceptions, including loss of internet connectivity (typically due to cellular connectivity issues). Show how eventual consistency is achieved.

Please provide the result in a public github repo, running on AWS instances if needed.