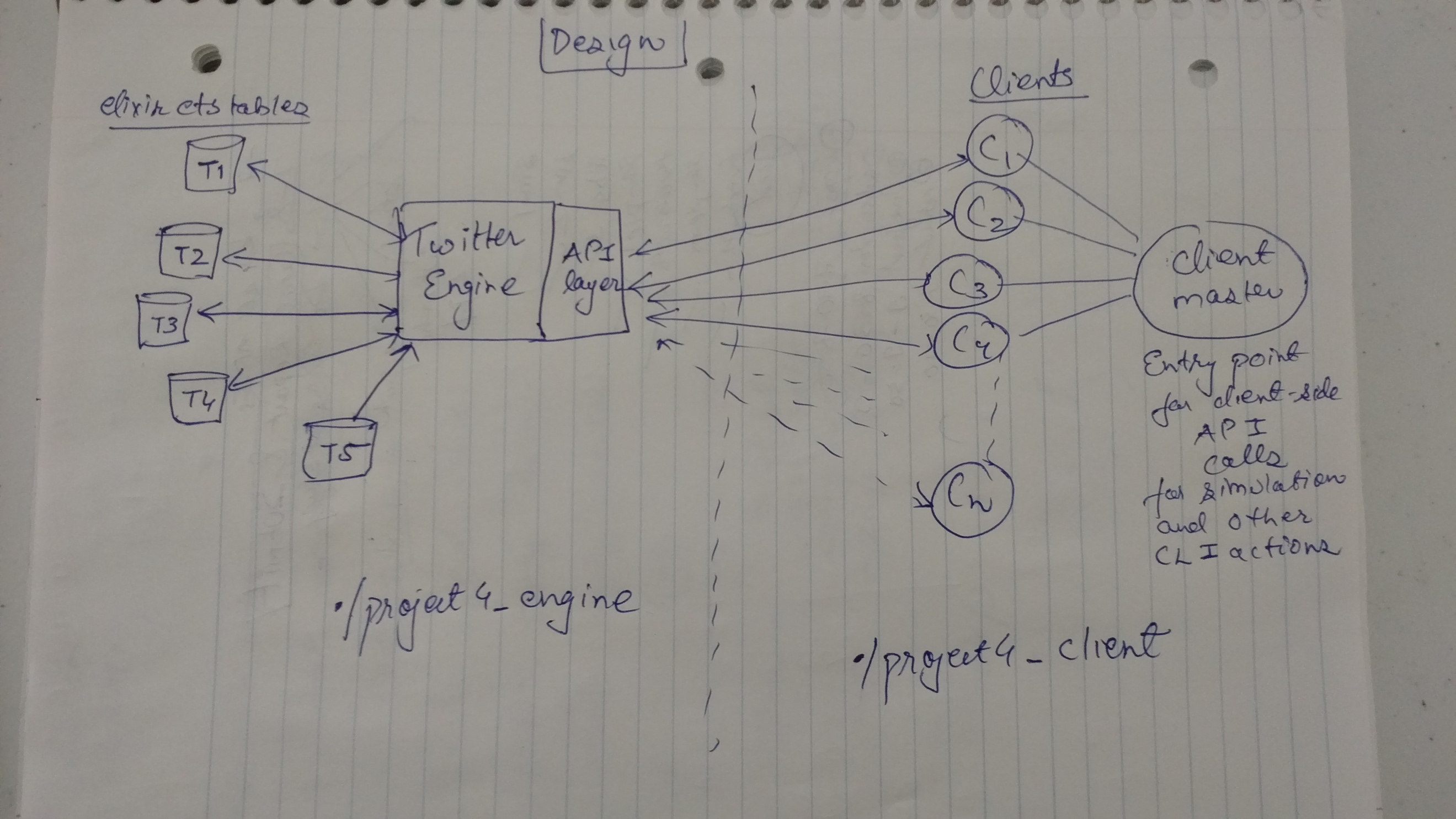
**README has directions as to how to run the project with the best practices for testing**

**In this Report, I discuss the design and performance of the solution**

# **Design**

A picture is worth a thousand words:



## Engine

The Twitter Engine is started by running ./project4\_engine. It supports the following APIs:

1. Register
   1. Register a new user
   2. Register the client-master for reporting stats
2. Feed – Get Feed for a user based on the users it is subscribed to
3. Hashtag
   1. Tweets containing a particular hashtag
   2. Sample of hashtags from the database for CLI actions
4. Mention
   1. Tweets containing a particular mention
   2. Sample of mentions from the database for CLI actions
5. Subscribe – User A subscribes to User B
6. Tweet – A user tweets
7. Retweet – A user retweets something that was in it’s feed

## Client

The clients are spawned by running ./project4\_client *with the appropriate parameters*. More information of various Command line actions and the order in which they need to be performed is in the README file

## Tables

I have used 5 tables (ets tables) for various kinds of querying with the following schemas. The primary key is in **bold**:

1. **UserId (int)** | SubscribedToUserIds(array of ints) | SubscribersUserIds(array of ints)
2. **UserId (int)** | TweetIds (array of ints)
3. **Hashtag (string)** | TweetIds (array of ints)
4. **Mention (string)** | TweetIds (array of ints)
5. **TweetId (int)** | Tweet (string) | timestamp (System.monotonic\_time in microseconds)

The tables help me in efficient retrieval of all the basic queries and API calls

# **Performance and Parameters**