**For a pictorial overview of the application**

1. Please open ksubaka-applicationOverview.xml in https://www.draw.io/

**High level overview:**

1. LoadInputQuery.java is the main class that accepts api and movie paramters
2. The input is validated using LoadInputQueryValidatorimpl
3. InputQueryHandlerFactory is used to get the corresponding handler based on the type of api.
4. The handlers then use ApliCallServiceImpl to call the corresponding api and obtain results in JSON format
5. InputQueryMovieHandlerImpl uses the HelperApiHandler to get all the movie related information.
6. The response from ApliCallServiceImpl is then printed to console.

**Improvements:**

1. I used [www.jsonschema2pojo.org](http://www.jsonschema2pojo.org/) to generate models from JSON. So they need to be refactored to move all the common fields to an Abstract model.
2. By accomplishing #1 the printThResult method in both the handlers can be moved to AbstractQueryhandler there by having only one copy of printThResult
3. More research can be done to find the movie api that gives all the information there by reducing the call to HelperApiHandler.
4. Should we do integration testing to test API calls as I don’t see mocking ClientResponse makes any difference.