Challanger Server

1) **BootStrap Class** : Will Bootstrap the application
2) **ServerTemplate** : Is based on Template design pattern and decides application Life cycle currently it manages server startup and metadata config flow
3) **ClientHandler**: Thread class receives the request from individual user and pass the same To the input analyzer to analyze.every client handler is observer by default So that in future we can have inter thread data communication for chatting Between user
3) **Input Analyzer** : It will analyze the incoming data and map it to corresponding Class to process.
4) **StageProcessorFactory:**Implements the factory pattern and provides the proper stage b processor class to the input analyzer.
5) **StageProcessor:** For every stage of workflow Application have separate processor class Which two methods :
1) getStageMessage - provides the stage level welcome message.
2) processInput - process input and pass control to the next stage.

6) Validator : every stage processor has validator which consist of two methods
1) Function validation
2) Technical (cosmetic and exception specific)

7) RouterMap : Consist Detail of Navigation of work flow .

8) Messanger : data transfer object which the standard format across both client Server and internal classes and methods.

9) Config : on bootstrap the classes with @Config (custom) annotation will be loaded And their loadconfig method will be called via reflection which eventually Metadata in the system which are stored in file with .ser format