**Documentation of Slackbot reminder system**

**Submitted by:** Laiba Aftab

**Submitted to:** Nabeel

**Classes:**

Rcrs

Createquartsjob

Schedularjava

SlackAPI

Reminder

TimerClock

User

UserDAO

Userslack

**Attributes and methods of classes:**

***Rcrs*—**Rcrs has a post method used as an end point take request payload from user and send response according. A path is set in the rcrs class to all the post method is postman

***User*—**User has an attribute of message, slack-handle and minutes from now. It is used to store the information of request payload and do the further processing.

***Reminder*—**Reminder class has no attributes but it is used for the validation of user. It checks and if user attributes are completely fine then it store that reminder in the database and according to time of reminder it returns a string of reminder timer.

***UserSlack*—**UserSlack is used to extract things from database and store them inside the attributes. It has a function where transaction from database is done having condition of current time and current date. It also gives all the information.

***SchedulerJava*—**It is importing a quartz library and used for scheduling a job and creating a trigger setting a time and after that there is a certain class defining a job which will execute on the time.

***CreateQuartzJob*—**when the time arrived this class will execute the required function which will retrieve information and process it and send notification to slack and after that delete that record from the database.

***SlackAPI*—**SlackAPI has a method of sendMessage having attributes of slack handle and message. First of all there is a configuration on SlackAPI where you make an app and get link of each direct messages and then in code compare the name and send notification accordingly.

***UserDAO*—**UserDAO is having all the functions to do retrieval and data entry according to different conditions. It is using SQL database and connect it with the drivers, set the user and password and then run the query accordingly.

***TimerClock*—**TimerClock is used for all the processing of timings. Its constructor will generate a current time and then the minutes will be added accordingly. It also has a function to return the date and time in a certain way which will further be used for setting records in database.

**Work Flow**

The **rcrs** file in resources folder is used to create end points. There is a post method implemented in rcrs class which is used to get a payload from user with the help of postman and return the response according to the request generated. The request payload should be in json form and have a key of messge, minutes\_from\_now and slack\_handle and none of its value should be empty. The post method receives the payload with the help of GSON it will extract the data members of a user and send it to reminder class for validation. If object is valid then a function from reminder class will enter the USER object into the database. If it stored properly then the program takes minutes taken from the user and call TimerClock class which will add the minutes in current time and get the updated time. It makes the scheduler object then set that time in the scheduler object. Schedular has an attribute of unique name of job and group of job. For this purpose UUID is used and is converted into string and send to scheduler object. Scheduler will create and trigger on updated time and schedule the job. When the specific time is reached then Createquartsjob will run the line of codes inside execute function. It will get record from database of current time and store it is Userslack object which will give attribute to the sendMessage of SlackAPI class and when message is send it will delete that record from database.