Logger

log()

print()

getTimeStamp()

format()

LoggerFile

FILENAME

print()

+clear()

EpitrelloJDBCDAO

URL, USER, PASSWORD

getConnection()

resetTable()

insertUser()

getAllUsers()

Configuration

FILENAME

getValueFromKey()

Use

/config
url=jdbc:h2:~/test2
user=test
password=test

ListService

lists

addList(title)

printAllLists()

printList()

getList(title)

listExists()

TaskService

tasks

addTask()

editTask()

assignTask()

deleteTask()

completeTask()

printTask()

moveTask()

taskExists()

moveTask()

deleteList(title)

printAllTasks()

printUnassignedTasksByPriority()

printUnfinishedTasksByPriority()

UserService

users

addUser()

userExists()

getUser()

printUserTasks()

printAllUsers()

printUsersByWorkload()

printUsersByPerformance()

List

title
tasks

printAllTasks()

Task

listAssigned

userAssigned

name

duration

priority

description

completed

hasUserAssigned()

User

name

tasks

assignTask()

unassignTask()

getTotalWorkload()

getRemainingWorkload()

getPerformance()

NOTE:
- **LinkedHashMap** has been used for the list of **tasks** added to the system in order to maintain their **insertion order**.
- **HashMap** has been used to store the list of **users** added to the system.
- **TreeMap** has been used to store the list of **lists** created so that they maintain an **ascending order.**
The usage of a Map in the three cases allows setting the task.name, user.name, and list.name as the key for
the respective Map. This let us quickly access a task, list, or user by their name. Besides, according to the project definition;
tasks, users, and lists must be unique.

NOTE:
getTotalWorkload() -> total estimated time **(duration of all tasks assigned to the user**)
getRemainingWorkload() -> total remaining time **(duration of all tasks assigned to the user and not completed**)
getPerformance() -> total time passed **(duration of all tasks assigned to the user and completed)**