|  |  |  |
| --- | --- | --- |
| Schedule Management Framework | December 22  2014 | |
| This document contains the problem, solution design, application build and run steps of schedule management process. | | Readme file |

Table of Contents

[Problem: 2](#_Toc407023204)

[Solution Design: 2](#_Toc407023205)

[Algorithm: 2](#_Toc407023206)

[Assumptions: 2](#_Toc407023207)

[Sequence Diagram of Conference Management process: 3](#_Toc407023208)

[File description: 4](#_Toc407023209)

[Pre-requisites: 5](#_Toc407023210)

[Build Application: 5](#_Toc407023211)

[Run Application: 6](#_Toc407023212)

# Problem:

Problem Two: Conference Track Management

You are planning a big programming conference and have received many proposals which have passed the initial screen process but you're having trouble fitting them into the time constraints of the day there are so many possibilities! So you write a program to do it for you.

* The conference has multiple tracks each of which has a morning and afternoon session.
* Each session contains multiple talks.
* Morning sessions begin at 9am and must finish by 12 noon, for lunch.
* Afternoon sessions begin at 1pm and must finish in time for the networking event.
* The networking event can start no earlier than 4:00 and no later than 5:00.
* No talk title has numbers in it.
* All talk lengths are either in minutes (not hours) or lightning (5 minutes).
* Presenters will be very punctual; there needs to be no gap between sessions.

# Solution Design:

## Algorithm:

1. Read the input file and create list of tasks in String format.
2. Validate each task in String format and create Task objects using task title and time duration.
3. Calculate the number of possible tracks using the total time duration.
4. Find the possible combinations of tasks for morning session (session duration is exactly 180 mins) and for evening session (session duration is greater than or equal to 180 mins and less than or equal to 240 mins). Mark each tasks assigned to each session as scheduled.
5. If there are tasks remaining un-scheduled; schedule those in evening session till time permits.
6. If all tasks scheduled create schedule list for each track using possible morning and evening combinations.

## Assumptions:

1. Morning session cannot be less or more than 180 minutes.
2. Given task cannot split into two sessions or more.
3. Maximum time duration for a given task is 240 minutes (equal to maximum session duration) and if there are tasks more than 240 minutes then the task list is invalid.

## Sequence Diagram of Conference Management process:



## File description:

|  |  |
| --- | --- |
| File Name | Description |
| ScheduleManagement\src\main\java\com\mycompany\schedulemanagement\conference\controler\ ConferenceControler.java | Controller class of Conference Management; calls schedule manager to create the list of schedules and print them out. |
| ScheduleManagement\src\main\java\com\mycompany\schedulemanagement\conference\model\ ConferenceManagerImpl.java | Model class of Conference Management; implements ScheduleManager interface and provides logics specific for scheduling conferences. |
| ScheduleManagement\src\main\java\com\mycompany\schedulemanagement\exception\ InvalidTaskException.java | Exception class in schedule management framework; wraps exceptions and validation failures during scheduling process. |
| ScheduleManagement\src\main\java\com\mycompany\schedulemanagement\model\ ScheduleConstant.java | Constant interface in schedule management framework; holds constant values used in frame work. |
| ScheduleManagement\src\main\java\com\mycompany\schedulemanagement\model\ ScheduleManager.java | Schedule management interface; provides definitions of common methods in schedule management framework. |
| ScheduleManagement\src\main\java\com\mycompany\schedulemanagement\model\dto\ Schedule.java | Schedule data transfers object; holds list of tasks and track id. |
| ScheduleManagement\src\main\java\com\mycompany\schedulemanagement\model\dto\ Schedule.java | Task data transfers object; holds attributes related to task. |
| ScheduleManagement\src\main\java\com\mycompany\schedulemanagement\model\util\ ScheduleUtil.java | Utility class in schedule management framework; provides common utility methods implementations. |
| ScheduleManagement\src\main\resources\ schedule-properties.properties | Holds configurable messages and properties. |
| \ScheduleManagement\src\main\resources\ TaskDetails.txt | Test input file. |
| ScheduleManagement\src\test\java\com\mycompany\schedulemanagement\conference\model\ ConferenceManagerImplTest.java | Unit test class of ConferenceManagerImpl.java |
| ScheduleManagement\src\test\java\com\mycompany\schedulemanagement\model\util\ ScheduleUtilTest.java | Unit test class of ScheduleUtil.java |

# Pre-requisites:

1. jdk1.8.0\_25
2. apache-maven-3.2.3
3. JUnit 4.8.2

# Build Application:

Open command prompt and move to the location of pom.xml

**(<Your directory path>/ScheduleManagement/pom.xml)**

Execute command “**mvn package**”. This will build the application and run the unit tests.

schedule-management-1.0.jar will be created inside following directory. **<Your directory path>/ScheduleManagement/target**



# Run Application:

Execute **“java -cp target/schedule-management-1.0.jar com.mycompany.schedulemanagement.conference.controler.ConferenceControler”.**

Give the path of file that contains task list and press enter.

