

# Activity Scheduler Application

activity-scheduler is an java based Activity Scheduler Application for planning events.

This application has the following features:

- The catalog of activities for the event will be read from an input file. ([Input file format](#))
- The employees will be divided into various teams and each team will be performing various activities in a day
- Output will be printed on console and to a file generate in the same directory as the application
- Application schedules activities using Greedy Algorithm to fit in maximum activities in a given day (A flavor of Dynamic Programming Knapsack problem)

Current implementation supports scheduling of activities for the following events:

- Deloitte Digital Away Day ([Requirement Specification](#))

## Tools and Technologies

Activity Scheduler Application uses the following tools and technologies:

| Technology | Version |
|------------|---------|
| JDK        | 1.0     |

|                      |        |
|----------------------|--------|
| BeanIO               | 2.1.0  |
| commons-io           | 2.5    |
| Log4j                | 2.7    |
| Junit                | 4.12   |
| Apache Common Lang 3 | 3.7    |
| Maven                | 4.0.0  |
| Git                  | 2.20.1 |
| Github               | N/A    |
| Eclipse              | Oxygen |

## Deliverables

The deliverables for the application can be found at the below location

[Activity Scheduler Application](#)

| Deliverable                    | Summary   |
|--------------------------------|---|
| CodeCoverageReport.PNG         | Contains the latest junit code coverage report of the project   |
| Requirement Specification.pdf  | Requirement Specification for “Deloitte Digital Away Day Event” |
| activities.txt                 | Sample input text file  |
| activity-scheduler-javadoc.zip | Javadoc for the project   |
| activity-scheduler.jar         | Executable “Activity Scheduler                                  |

## Assumptions

- All the activities in the file needs to be scheduled
- Staff Motivation Presentation is currently configured to be of 15mins
- A team will be created even if it doesn't consist of activities that can stretch whole day.

This is to accomodate the assumption that all activities needs to be scheduled and the input file may not contain sufficient number of activities that can be schedule for a team on a day

- User has necessary read-write permissions to the folder containing the application which are required for the application to generate output file

## Setup

- Ensure Jdk 1.8 or above is installed on your system. If not, you can download the latest version from the below link.

Please follow the instructions in the link for the setup

[JDK Installation](#)

- Click on the below link to download the Activity Scheduler Application

[Activity Scheduler Application](#)

- unzip the application to the folder of your choice.
- After you unzip the application, inside the activity-

scheduler/deliverables folder you will find application executable with the name “activity-scheduler.jar”

- A sample input file for the application is also provided in the activity-scheduler/deliverables folder.

Below is the allowed format of activities in the file

| Type           | Format                                  | Example                   |
|----------------|---|---------------------------|
| Timed Activity | [activity name]<br>[time_in_minutes]min | Duck Herding<br>60min     |
| Sprint         | [activity name] sprint                  | Salsa & Pickles<br>sprint |

### Sample Input file content

```
Duck Herding 60min
Archery 45min
Learning Magic Tricks 40min
Laser Clay Shooting 60min
Human Table Football 30min
Buggy Driving 30min
Salsa & Pickles sprint
2-wheeled Segways 45min
Viking Axe Throwing 60min
Giant Puzzle Dinosaurs 30min
Giant Digital Graffiti 60min
Cricket 2020 60min
Wine Tasting sprint
Arduino Bonanza 30min
```

Digital Tresure Hunt 60min

Enigma Challenge 45min

Monti Carlo or Bust 60min

New Zealand Haka 30min

Time Tracker sprint

Indiano Drizzle 45min

## Execution

- Open a terminal and navigate to the “activity-scheduler\deliverables” folder

```
$ cd [PATH TO APPLICATION FOLDER]\deliverables
```

- run the following command to execute the jar in the folder

```
$ java -jar activity-scheduler.jar [INPUT FILE PATH]
```

## Example

```
C:\>cd C:\Users\rbutti\git\ActivityScheduler2\deliverables
```

```
C:\Users\rbutti\git\ActivityScheduler2\deliverables>java -  
jar activity-scheduler.jar activities.txt
```

Team 1:

09:00 AM : Human Table Football

09:30 AM : Buggy Driving

10:00 AM : Salsa & Pickles sprint

10:15 AM : Giant Puzzle Dinosaurs  
10:45 AM : Wine Tasting sprint  
11:00 AM : Arduino Bonanza  
11:30 AM : New Zealand Haka  
12:00 PM : Lunch Break  
01:00 PM : Archery  
01:45 PM : Learning Magic Tricks  
02:25 PM : 2-wheeled Segways  
03:10 PM : Enigma Challenge  
03:55 PM : Time Tracker sprint  
04:10 PM : Indiano Drizzle  
05:00 PM : Staff Motivation Presentation

Team 2:

09:00 AM : Duck Herding  
10:00 AM : Laser Clay Shooting  
11:00 AM : Viking Axe Throwing  
12:00 PM : Lunch Break  
01:00 PM : Giant Digital Graffiti  
02:00 PM : Cricket 2020  
03:00 PM : Digital Treasure Hunt  
04:00 PM : Monte Carlo or Bust  
05:00 PM : Staff Motivation Presentation

C:\Users\rbutti\git\ActivityScheduler2\deliverables>

## Technical Design

# Project Structure



## Package Structure

| Package                             | Summary                                 |
|-------------------------------------|---|
| com.activityscheduler.application   | contains application entrypoint classes |
| com.activityscheduler.constant      | contains constants                      |
| com.activityscheduler.domain        | contains all the domain classes         |
| com.activityscheduler.exception     | contains custom application exceptions  |
| com.activityscheduler.facade        | intefaces for facade classes            |
| com.activityscheduler.facade.impl   | implementation of facade classes        |
| com.activityscheduler.service       | intefaces for service classes           |
| com.activityscheduler.service.impl  | implementation for service classes      |
| com.activityscheduler.strategy      | intefaces for strategy classes          |
| com.activityscheduler.strategy.impl | implementation for strategy classes     |

## Domains

| Package                   | Summary  |
|---------------------------|--|
| AbstractDomainObject.java | Abstact class implemented by all Domain objects in the application |
|                           |  |

|                               |   |
|-------------------------------|---|
| Activity.java                 | Represents the activity that needs to be scheduled for an event               |
| ActivityCatalog.java          | A catalog of all the activities that needs to be scheduled for an event       |
| ActivitySchedule.java         | Schedule of Activities  |
| EventInfo.java                | Holds information regarding a particular event                                |
| DeloitteAwayDayEventInfo.java | Extends EventInfo and holds information specific to “Deloitte Away Day Event” |
| Team                          | Represents a team created for an event  |

## Contact Information

Author : Ravikiran Butti,

Email Id : ([ravikiran763@gmail.com](mailto:ravikiran763@gmail.com))