Back-end developer assignment

Your employer has an existing system for employees to submit booking requests for meetings in the boardroom. Your employer has now asked you to implement a system for processing batches of booking requests. Current system is based on text messages, but you need to do it using JSON.

Please implement following REST web service using one of following development frameworks to succeed:

- Jersey (http://jersey.java.net/)
- Restlet (http://restlet.org/)
- Spring MVC (http://www.springsource.org/)
- Any other cool JVM-based framework

Your processing system must receive input (and sent output) message not as text, but in JSON format. Below you see the data as row text (how it's used in a current system), so please create a correspondent JSON structure yourself.

Input

The first line of the input text represents the company office hours, in 24 hour clock format, and the remainder of the input represents individual booking requests. Each booking request is in the following format.

[request submission time, in the format YYYY-MM-DD HH:MM:SS] [employee id]
[meeting start time, in the format YYYY-MM-DD HH:MM] [meeting duration in hours]

A sample input text:

```
0900 1730

2011-03-17 10:17:06 EMP001

2011-03-21 09:00 2

2011-03-16 12:34:56 EMP002

2011-03-21 09:00 2

2011-03-16 09:28:23 EMP003

2011-03-22 14:00 2

2011-03-17 11:23:45 EMP004

2011-03-22 16:00 1

2011-03-15 17:29:12 EMP005

2011-03-21 16:00 3
```

Output
Your system must provide a successful booking calendar as output, with bookings being grouped chronologically by day. For the sample input displayed above, your system must provide the following output:

```
2011-03-21

09:00 11:00 EMP002

2011-03-22

14:00 16:00 EMP003

16:00 17:00 EMP004
```

Constraints / Notes

- No part of a meeting may fall outside office hours.
- Meetings may not overlap.
- The booking submission system only allows one submission at a time, so submission times are guaranteed to be unique.
- Bookings must be processed in the chronological order in which they were submitted.
- The ordering of booking submissions in the supplied input is not guaranteed.
- The current requirements make no provision for alerting users of failed bookings; it is up to the user to confirm that their booking was successful.

Results

- The purpose of this test is not to get a "right" or "wrong" answer, but it's to see how you code. So, show us what you've got!

- Please create production-quality code. This means naming conventions, coding style, sensible design and meaningful commenting. If you feel you can give your work to a brand new colleague with a minimal of hand-over, you've probably got it right.

- We don't expect you to use every design pattern you've ever heard of. Please only apply patterns when it makes sense to do so.

- We're not expecting you to have optimized the solution for performance or memory size. Readability is more important than performance.

- We are big fans of Test Driven Development, so including unit tests is a good idea.

- Let us know how long you spent on the task.

- If you use additional libraries we'd expect to see a Maven or Gradle build script.

- Results should be either “pushed” to private Git or Mercurial repository on Github or Bitbucket or zipped together with a Mercurial repo (.hg folder) and sent to us.