

```
package com.example.TaskManager;

import org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.SpringBootApplication
;

@SpringBootApplication
public class TaskManagerApplication {

    public static void main(String[] args) {
        SpringApplication.run(TaskManagerApplication.class,
args);
    }

}
```

```

package com.example.TaskManager.controllers;

import java.util.Date;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.example.TaskManager.entities.Task;
import com.example.TaskManager.services.TaskService;

@Controller
public class TaskController {

    @Autowired
    private TaskService taskService;

    @GetMapping(value="/")
    public String home() {
        return "index";
    } //end of method 'home()'

    @GetMapping(value="/list")
    public String list(ModelMap model) {
        Optional<Task> task =
taskService.GetTaskById(1); //this was a failed attempt, but
had no time to rebound with fix
        model.addAttribute("tasks", task);
        return "list";
    } //end of method 'list()'

```

```

@GetMapping(value="/add")
public String add() {
    return "add";
} //end of method 'add()'

@PostMapping(value="/add")
public String addToTask(@RequestParam("fname") String
name, @RequestParam("email") String email,
    @RequestParam("sdate")
@DateTimeFormat(pattern="yyyy-mm-dd") Date sDate,
@RequestParam("edate")
    @DateTimeFormat(pattern="yyyy-mm-dd") Date eDate,
@RequestParam("desc") String desc, @RequestParam("sev")
    String sev, @RequestParam("user_id") int u_id ) {

    if(!name.isBlank() || !email.isBlank() ||
!sDate.toString().isBlank() ||
        !eDate.toString().isBlank() ||
!sev.isBlank() || u_id == 0) {
        Task task = new Task();
        task.setName(name);
        task.setEmail(email);
        task.setStartDate(sDate);
        task.setEndDate(eDate);
        task.setDescription(desc);
        task.setSeverity(sev);
        task.setUser(u_id);

        taskService.taskSaver(task);
    } //end of 'if statement'

    return "add";
} //end of method 'addToTask()'

@GetMapping(value="/update")
public String update() {
    return "update";
} //end of method 'home()'

```

```

        @PostMapping(value="/update")
        public String updateTask(@RequestParam("id") int t_ID,
        @RequestParam("fname") String name, @RequestParam("email")
        String email,
                @RequestParam("sdate")
        @DateTimeFormat(pattern="yyyy-mm-dd") Date sDate,
        @RequestParam("edate")
        @DateTimeFormat(pattern="yyyy-mm-dd") Date eDate,
        @RequestParam("desc") String desc, @RequestParam("sev")
        String sev,@RequestParam("user_id") int u_id ) {
            taskService.deleteTask(t_ID);

            Task task = new Task();
            task.setId(t_ID);
            task.setName(name);
            task.setEmail(email);
            task.setStartDate(sDate);
            task.setEndDate(eDate);
            task.setDescription(desc);
            task.setSeverity(sev);
            task.setUser(u_id);

            taskService.taskUpdater(task);

            return "update";
        } //end of method 'updateTask()'

        @GetMapping(value="/delete")
        public String delete() {
            return "delete";
        } //end of method 'home()'

        @PostMapping(value="/delete")
        public String delete(@RequestParam("id") String id) {
            taskService.deleteTask(Integer.parseInt(id));
            return "delete";
        } //end of method 'delete()'

```

}

```

package com.example.TaskManager.entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;

import lombok.Data;

import java.util.Date;

@Entity

public class Task {

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer id;
    private String name;
    private Date startDate;
    private Date endDate;
    private String severity;
    private String description;
    private String email;
    @Column(name="user_id")
    private int user;

    public Task() {}
    public Task(String name, Date startDate, Date endDate,
String severity, String description, String email,
        int user) {
        super();
        this.name = name;
        this.startDate = startDate;
        this.endDate = endDate;
        this.severity = severity;
    }

```

```
        this.description = description;
        this.email = email;
        this.user = user;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Date getStartDate() {
        return startDate;
    }

    public void setStartDate(Date startDate) {
        this.startDate = startDate;
    }

    public Date getEndDate() {
        return endDate;
    }

    public void setEndDate(Date endDate) {
        this.endDate = endDate;
    }

    public String getSeverity() {
        return severity;
    }
}
```

```
}

    public void setSeverity(String severity) {
        this.severity = severity;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public int getUser() {
        return user;
    }

    public void setUser(int user) {
        this.user = user;
    }

}
```



```
package com.example.TaskManager.entities;
```

```
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;
```

```
@Entity // This tells Hibernate to make a table out of this  
class
```

```
public class User {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private Integer id;  
  
    private String name;  
  
    private String email;  
  
    private String password;  
  
    public String getPassword() {  
        return password;  
    }  
  
    public void setPassword(String password) {  
        this.password = password;  
    }  
  
    public Integer getId() {  
        return id;  
    }  
  
    public void setId(Integer id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

```
}

public void setName(String name) {
    this.name = name;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}
}
```

```
package com.example.TaskManager.repositories;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.example.TaskManager.entities.Task;
import com.example.TaskManager.entities.User;

@Repository
public interface TaskRepository extends CrudRepository<Task,
Integer> {

    public Task findByName(String name);

    public Iterable<Task> findAllByUser(User user);

    public Iterable<Task> findById(int id);
}
```

```
package com.example.TaskManager.repositories;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.example.TaskManager.entities.User;

@Repository
public interface UserRepository extends CrudRepository<User,
Integer> {

    public User findByName(String name);
    public User findById(int id);
}
```

```

package com.example.TaskManager.services;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.TaskManager.entities.Task;
import com.example.TaskManager.entities.User;
import com.example.TaskManager.repositories.TaskRepository;

@Service
public class TaskService {

    @Autowired
    private TaskRepository taskRepository;

    public Iterable<Task> GetAllTasks() {

        return taskRepository.findAll();
    }

    public Optional<Task> GetTaskById(Integer taskId) {

        //TODO: create exception for task not found.

        return taskRepository.findById(taskId);
    }

    public Iterable<Task> GetTasksByUser(User user) {
        //TODO: what do we do if the user doesn't have any
        tasks or doesn't exist?

        return(taskRepository.findAllByUser(user));
    }
}

```

```
public void taskSaver(Task task) {  
    taskRepository.save(task);  
} //end of method 'taskSaver()'  
  
public void taskUpdater(Task task) {  
    taskRepository.save(task);  
} //end of method 'taskSaver()'  
  
public void deleteTask(int id) {  
    taskRepository.deleteById(id);  
} //end of method "deleteTask()  
}
```

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3
306/phase_end3
spring.datasource.username=root
spring.datasource.password=Mossie123?
```

```
logging.level.org.springframework.web: DEBUG
spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix=.jsp
server.port=8090
```

```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Add Task</title>
</head>
<body>
<h1>Add A Task</h1>
    <form action="add" method="post">
        <label for="fname">Task name:</label><br>
        <input type="text" id="fname" name="fname"><br>
        <label for="email">Email:</label><br>
        <input type="text" id="email" name="email"><br>
        <label for="sdate">Start Date:</label><br>
        <input type="date" id="sdate" name="sdate"><br>
        <label for="edate">End Date:</label><br>
        <input type="date" id="edate" name="edate"><br>
        <label for="desc">Description:</label><br>
        <textarea id="desc" name="desc" rows=10
cols=30></textarea><br>
        <label for="sev">Severity</label><br>
        <select id="sev" name="sev">
            <option value="high">High</option>
            <option
value="medium">Medium</option>
            <option value="low">Low</option>
        </select>
        <br>
        <label for="user_id">User ID:</label><br>
        <input type="text" id="user_id" name="user_id"><br>
        <input type="submit" value="Submit">

    </form>
</body>
</html>
```



```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Delete Task</title>
</head>
<body>
<h1>Delete A Task</h1>
    <form action="delete" method="post">
        <label for="id">Task ID:</label><br>
        <input type="text" id="id" name="id"><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

```
<html>
<head>
<title>Spring Application</title>
```

```
</head>
<body>
  <h1 align="center">Welcome please select an option</h1>

  <div align="center">
    <h4>
      <a href="list">List my Tasks</a>
    </h4>
  </div>

  <div align="center">
    <h4>
      <a href="add">Add Task</a>
    </h4>
  </div>

  <div align="center">
    <h4>
      <a href="update">Update a Task</a>
    </h4>
  </div>

  <div align="center">
    <h4>
      <a href="delete">Delete a Task</a>
    </h4>
  </div>
</body>
</html>
```

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>List Tasks</title>
</head>
<body>
<h1>Welcome to your list of tasks</h1>
<table>
    <tr><th>Task ID</th><th>Task
Name</th><th>Description</th><th>Email</th><th>Start
Date</th><th>End Date</th><th>Severity</th><th>User
ID</th></tr>
    <c:forEach items="${tasks}" var="task"
varStatus="count">
        <tr id="${count.index}">
            <td>${tasks.id}</td>
            <td>${tasks.name}</td>
            <td>${tasks.description}</td>
            <td>${tasks.email}</td>
            <td>${tasks.startDate}</td>
            <td>${tasks.endDate}</td>
            <td>${tasks.severity}</td>
            <td>${tasks.user}</td>
        </tr>
    </c:forEach>
</table>
</body>
</html>

```

```

<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Update Tasks</title>
</head>
<body>
<h1>Update A Task</h1>
<form action="update" method="post">
    <label for="id">Task ID:</label><br>
    <input type="text" id="id" name="id"><br>
    <label for="fname">Task name:</label><br>
    <input type="text" id="fname" name="fname"><br>
    <label for="email">Email:</label><br>
    <input type="text" id="email" name="email"><br>
    <label for="sdate">Start Date:</label><br>
    <input type="date" id="sdate" name="sdate"><br>
    <label for="edate">End Date:</label><br>
    <input type="date" id="edate" name="edate"><br>
    <label for="desc">Description:</label><br>
    <textarea id="desc" name="desc" rows=10
cols=30></textarea><br>
    <label for="sev">Severity</label><br>
    <select id="sev" name="sev">
        <option value="high">High</option>
        <option
value="medium">Medium</option>
        <option value="low">Low</option>
    </select>
    <br>
    <label for="user_id">User ID:</label><br>
    <input type="text" id="user_id" name="user_id"><br>
    <input type="submit" value="Submit">

```

```
    </form>
</body>
</html>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.4.4</version>
    <relativePath/> <!-- lookup parent from repository
-->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>TaskManager</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>TaskManager</name>
  <description>Phase 3 End Projet</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-
jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-
web</artifactId>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>

```

```

        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
    </dependency>
</dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
</dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>true</optional>
    </dependency>
</dependency>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>

<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-
test</artifactId>
    <scope>test</scope>
</dependency>

</dependencies>

<build>
    <plugins>
        <plugin>

```

```
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-
plugin</artifactId>
        <configuration>
            <excludes>
                <exclude>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>
                </exclude>
            </excludes>
        </configuration>
    </plugin>
</plugins>
</build>

</project>
```