HUBEI UNIVERSITY OF AUTOMOTIVE TECHNOLOGY

SCHOOL OF ELECTRICAL AND INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

Computer major Comprehensive Practical Training II

####

Student Report

Student Id: 202041201

Student Name: Ndombasi Joao Andre Diakusala

Teacher: Qi Xin

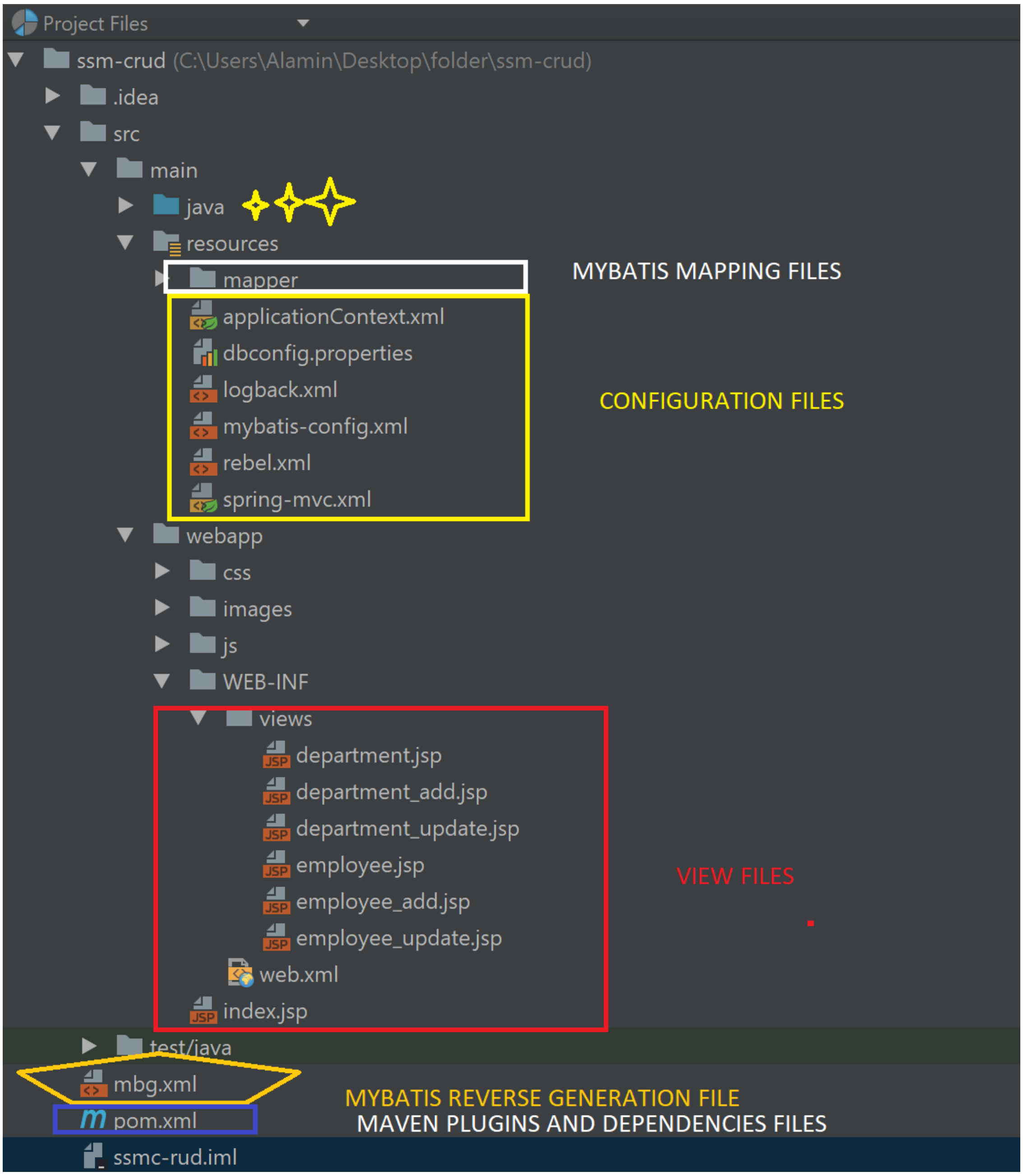
1. Objective

This report is the comprehensive discussion of SSM(SPRING + SPRING-MVC + MyBatis) framework. It uses a concreate example to demonstrate how everything is working together from database(DAO) layer to presentation layer(Front-End).

It emphasizes on the MVC project style, CRUD and REST API. It brings forth the best framework for Java EE web application development and modern adapted solutions for easy and quick implementation of tasks.

2. Project Organization

The demo project is named ssm-crud and project files are organized as follow:



```
Project Files
▼ ssm-crud (C:\Users\Alamin\Desktop\folder\ssm-crud)
    ▶ .idea
    ▼ src
        ▼ main
            ▶ java
            ▼ resources
                ▶ mapper
                  applicationContext.xml
                  dbconfig.properties
                  logback.xml
                  mybatis-config.xml
                  rebel.xml
                  spring-mvc.xml
            ▼ webapp
                ▶ css
                ▶ images
                ▶ js
                ▼ WEB-INF
                    ▼ views
                          department.jsp
                          department_add.jsp
                          department_update.jsp
                          employee.jsp
                          employee_add.jsp
                          employee_update.jsp
                      web.xml
                  index.jsp
        ▶ test/java
      mbg.xml
      pom.xml
      ssmc-rud.iml
```

MYBATIS MAPPING FILES

CONFIGURATION FILES

VIEW FILES

MYBATIS REVERSE GENERATION FILE

MAVEN PLUGINS AND DEPENDENCIES FILES

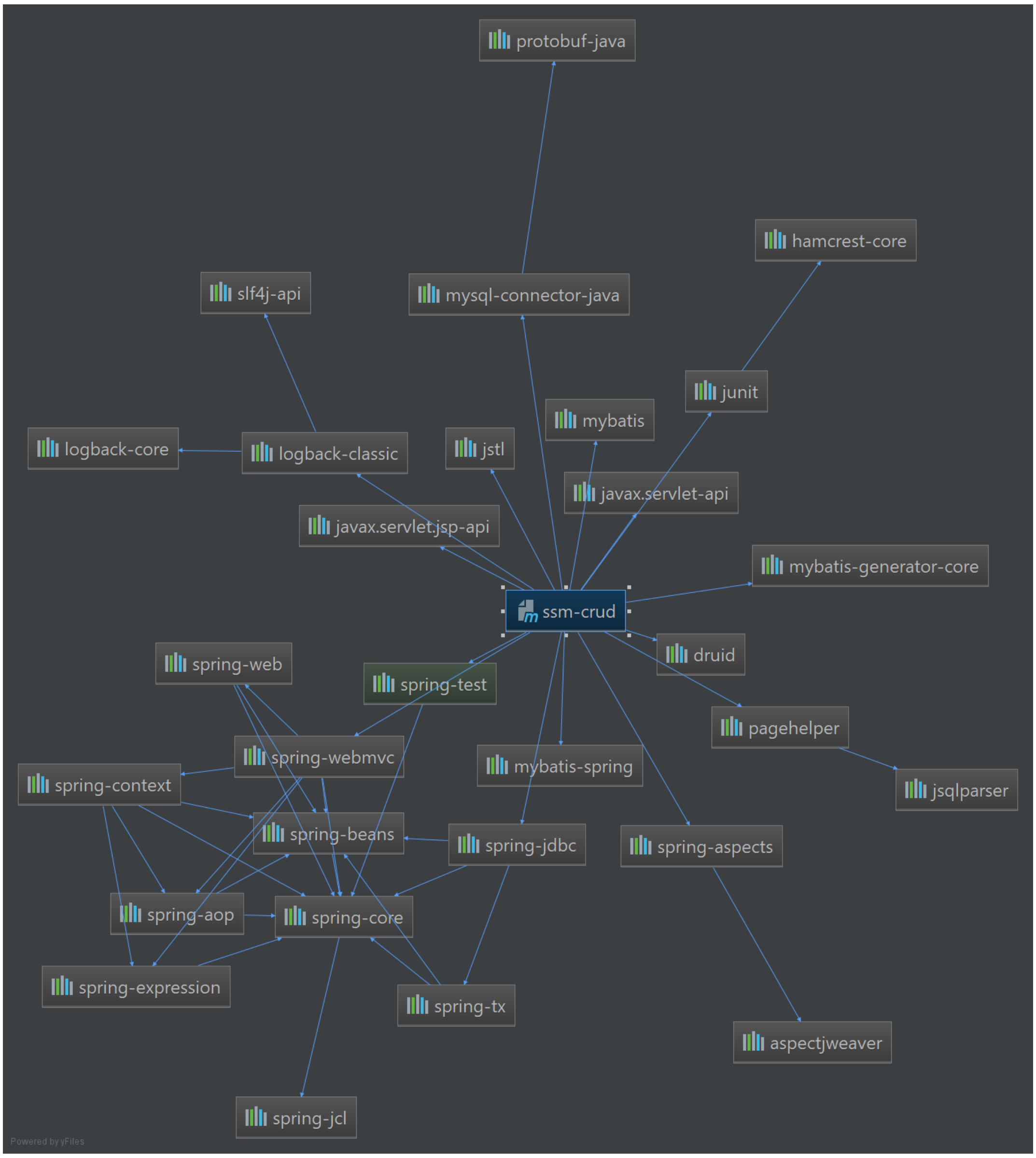
Files Description

Files, as shown on the image above, are mainly divided into two categories:

1. **CONFIGURATION FILES**

   These XML files are set to set all the logic behind the SSM logic.

   - POM: project maven dependencies



- ApplicationContext.xml : gives the business logic. configure transaction enhancements which are made by Spring **beans definition**, **context component scan** and **AOP configurations**.

- dbconfig.properties: it sets the JDBC properties (JDBC: JAVA DATABASE CONNECTIVITY)

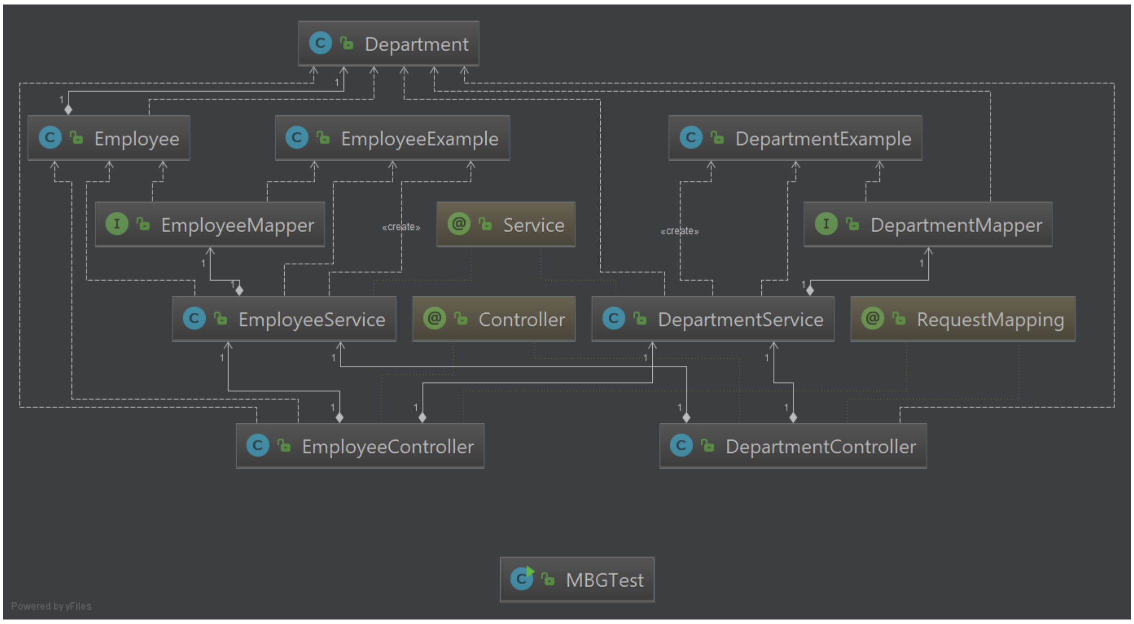
- logback.xml: It gives the detailed logging of debugging process when using the databse services.

- mybatis-config.xml: set the DBMS (database management system) used, in thiis case MySQL and enables the page helper plugin.

- rebel.xml: This is the JRebel configuration file. It maps the running application to your IDE workspace, enabling JRebel reloading for this project.
- spring-mvc.xml: sets the component-scan base package for spring and the default servlet handler.

2. JAVA FILES

This is the java code used to implement the system. I gives 5 layers for data processing in and out the server.



- Bean or VO: java classes to define the objects that are used to encapsulate the data in transactions.

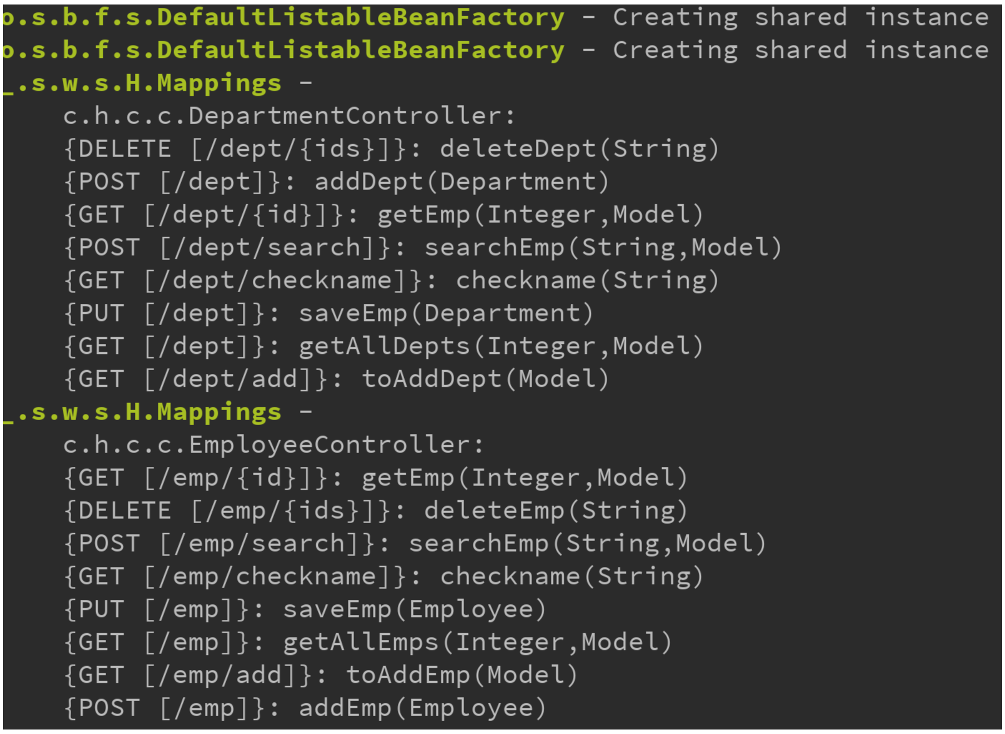
  In occurrence, Department.java and DepartmentExample.java

- DAO: Data Access Object, A java interface that is mapped with a class configuration mapper file to sets all the signatures of methods used by the referenced bean. In occurrence, EmployeeMapper.xml and DepartmentMapper.xml.

- Controller: It is the spring version of servlets, It gets requests and sends back responses. spring annotation @Controller

- Service: sets the way for data flow in and outside the database. spring annotation @Service. In occurrence, EmployeeService.java and DepartmentService.java.

- Util: MyBatisTest.java, the MyBatis framework reverse generation technology is used to generate all the beans, mapping interfaces and mapping xml files.

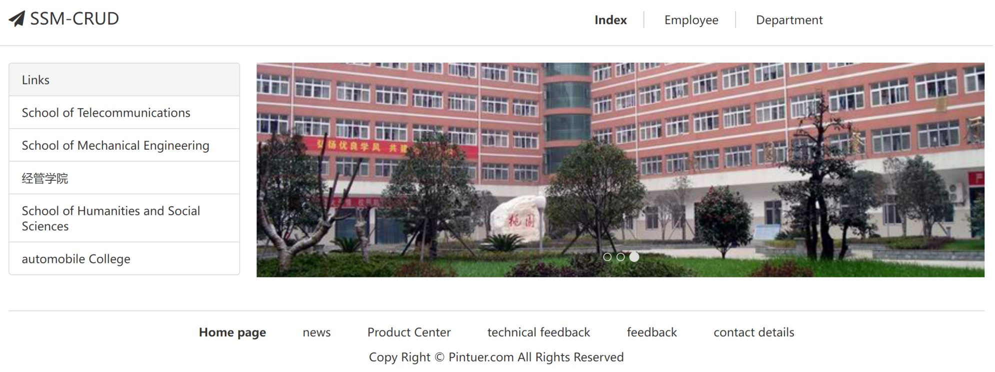


```
o.s.b.f.s.DefaultListableBeanFactory - Creating shared instance
o.s.b.f.s.DefaultListableBeanFactory - Creating shared instance
_.s.w.s.H.Mappings -
    c.h.c.c.DepartmentController:
    {DELETE [/dept/{ids}]}: deleteDept(String)
    {POST [/dept]}: addDept(Department)
    {GET [/dept/{id}]}: getEmp(Integer,Model)
    {POST [/dept/search]}: searchEmp(String,Model)
    {GET [/dept/checkname]}: checkname(String)
    {PUT [/dept]}: saveEmp(Department)
    {GET [/dept]}: getAllDepts(Integer,Model)
    {GET [/dept/add]}: toAddDept(Model)
_.s.w.s.H.Mappings -
    c.h.c.c.EmployeeController:
    {GET [/emp/{id}]}: getEmp(Integer,Model)
    {DELETE [/emp/{ids}]}: deleteEmp(String)
    {POST [/emp/search]}: searchEmp(String,Model)
    {GET [/emp/checkname]}: checkname(String)
    {PUT [/emp]}: saveEmp(Employee)
    {GET [/emp]}: getAllEmps(Integer,Model)
    {GET [/emp/add]}: toAddEmp(Model)
    {POST [/emp]}: addEmp(Employee)
```

> ## LOGIC
>
> the logic is fairly simple.
>
> A request is sent from the browser and the DispatcherServlet filters the controllers and passes the request object to the right referenced URL and method (GET, POST, UPDATE, DELETE) which defines CRUD API. and the controller calls the Service layer for implementation of the operation, the service calls for mapper file of DAO layer for database resources and the result of the operations is given by rebel.
>
> When the response is readym, data come from database and upto controller that will pass the objects to the Model(session) and the view mapping will help call the right view page.

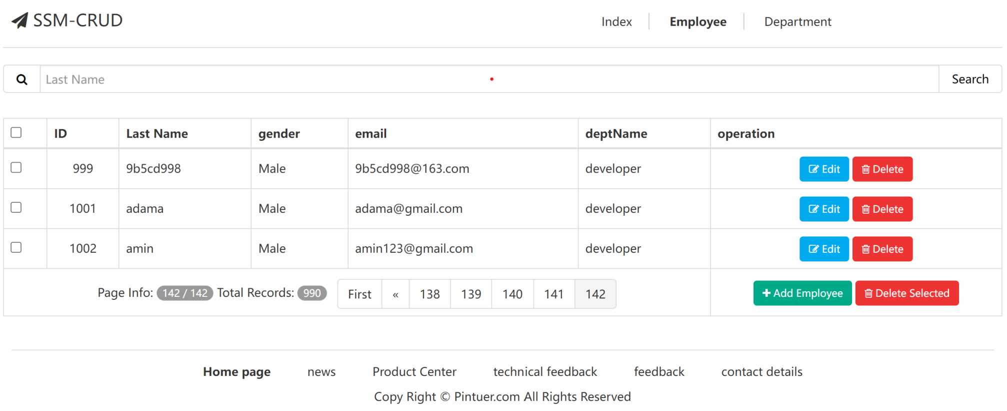
3. **VIEW FILES**

Home Page ---> index.jsp


SSM-CRUD
Index
Employee
Department
Links
School of Telecommunications
School of Mechanical Engineering
经管学院
School of Humanities and Social Sciences
automobile College
Home page
news
Product Center
technical feedback
feedback
contact details
Copy Right © Pintuer.com All Rights Reserved

EMPLOYEE

- Employee Home Page ---> employee.jsp. list of all employees with pagination



SSM-CRUD | Index | Employee | Department

Last Name | Search

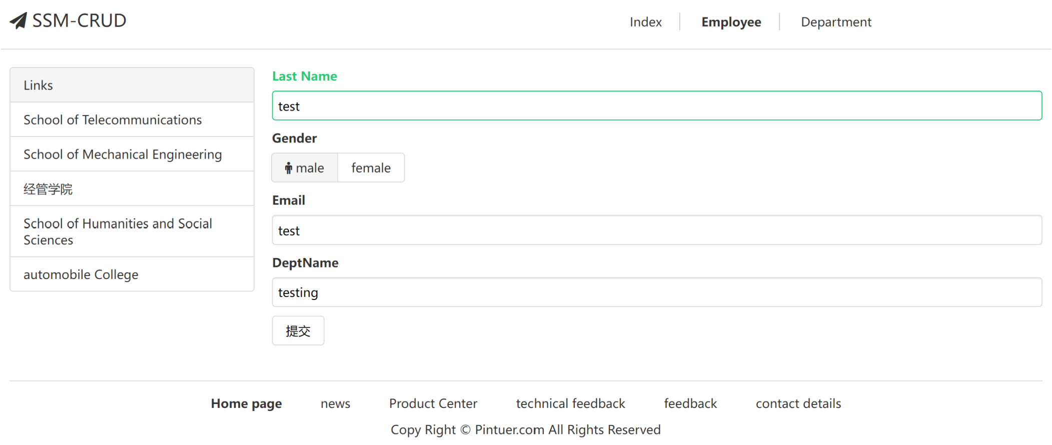
□	ID	Last Name	gender	email	deptName	operation
□	999	9b5cd998	Male	9b5cd998@163.com	developer	Edit Delete
□	1001	adama	Male	adama@gmail.com	developer	Edit Delete
□	1002	amin	Male	amin123@gmail.com	developer	Edit Delete

Page Info: 142 / 142 Total Records: 990 | First « 138 139 140 141 142 | + Add Employee | Delete Selected

Home page news Product Center technical feedback feedback contact details

Copy Right © Pintuer.com All Rights Reserved

- Add Employee Page: a form to register a new employee.



SSM-CRUD | Index | Employee | Department

Links
School of Telecommunications
School of Mechanical Engineering
经管学院
School of Humanities and Social Sciences
automobile College

Last Name: test

Gender: male | female

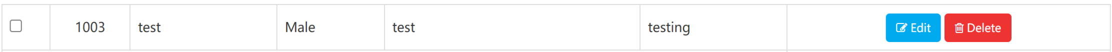
Email: test

DeptName: testing

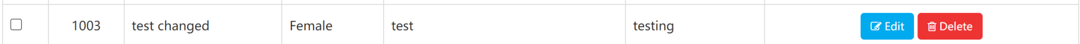
提交

Home page news Product Center technical feedback feedback contact details

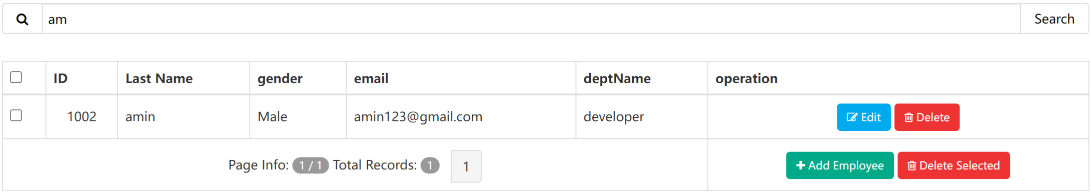
Copy Right © Pintuer.com All Rights Reserved

> RESULT id=1003
>
> 
>
> | | | | | | | |
> |---|---|---|---|---|---|---|
> | □ | 1003 | test | Male | test | testing | Edit Delete |

- Update Employee id: 1003 name changes to testing and gender to Female

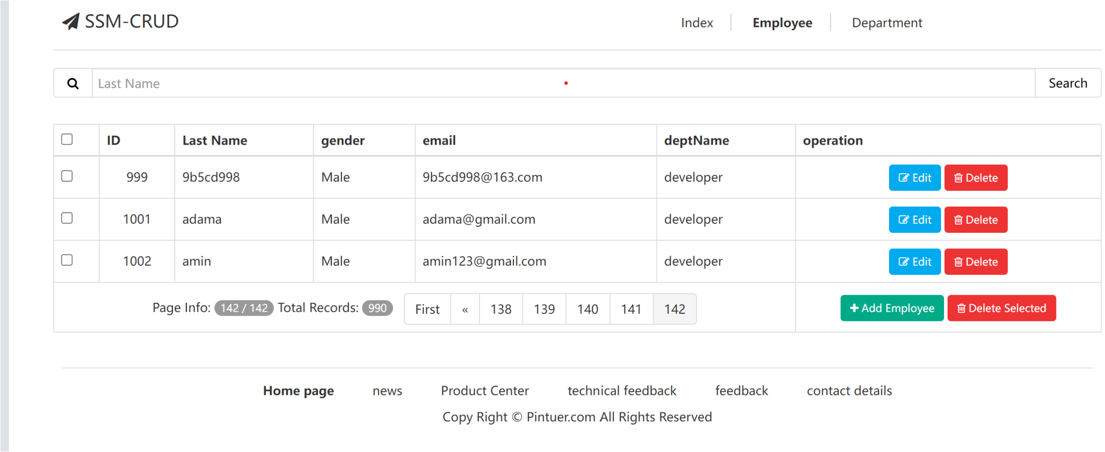
> RESULT
>
> 
>
> | | | | | | | |
> |---|---|---|---|---|---|---|
> | □ | 1003 | test changed | Female | test | testing | Edit Delete |

- Search Emplyee fuzzy search key=am

> RESULT id=1002 and name = amin
>
> 
>
> am | Search
>
> | □ | ID | Last Name | gender | email | deptName | operation |
> |---|---|---|---|---|---|---|
> | □ | 1002 | amin | Male | amin123@gmail.com | developer | Edit Delete |
>
> Page Info: 1 / 1 Total Records: 1 | 1 | + Add Employee | Delete Selected

- Delete Employee id: 1003 is removed from the table.

> RESULT



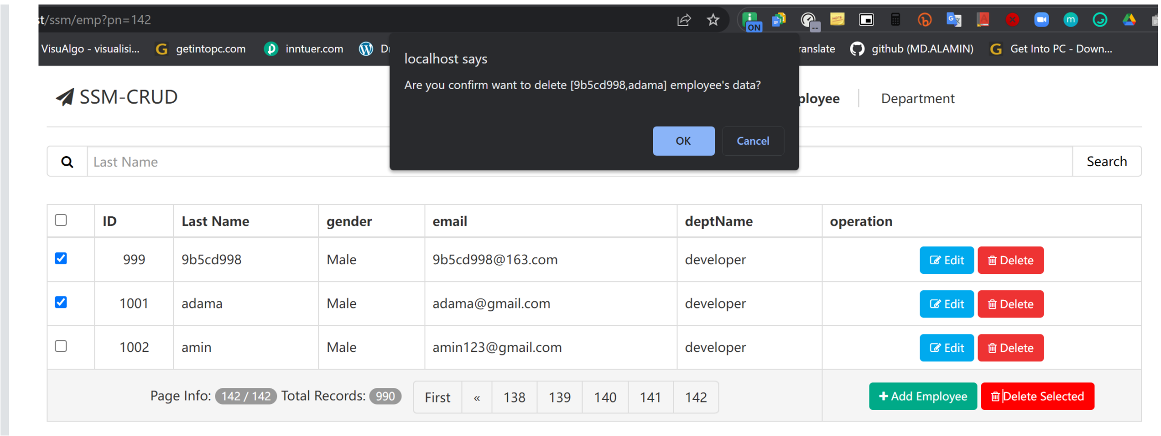
SSM-CRUD

□	ID	Last Name	gender	email	deptName	operation
□	999	9b5cd998	Male	9b5cd998@163.com	developer	Edit Delete
□	1001	adama	Male	adama@gmail.com	developer	Edit Delete
□	1002	amin	Male	amin123@gmail.com	developer	Edit Delete

Page Info: 142 / 142 Total Records: 990

Copy Right © Pintuer.com All Rights Reserved

- Batch Delete Employee id: 999 and 1001 will be removed from the table.



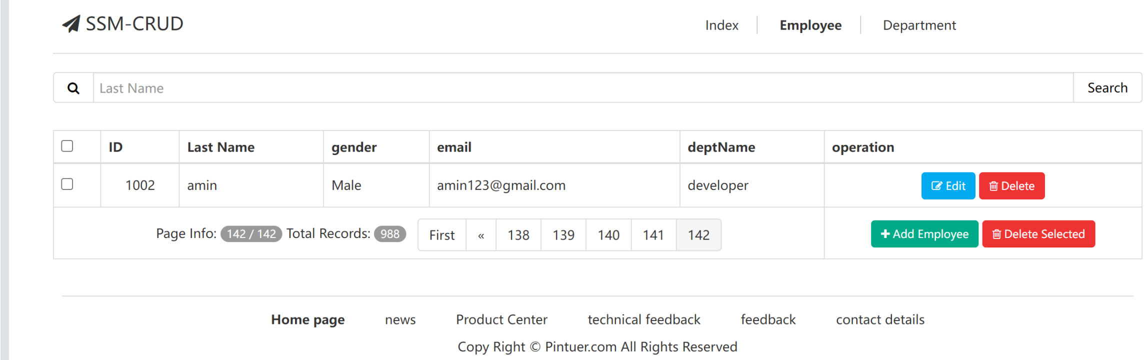
localhost says

Are you confirm want to delete [9b5cd998,adama] employee's data?

□	ID	Last Name	gender	email	deptName	operation
☑	999	9b5cd998	Male	9b5cd998@163.com	developer	Edit Delete
☑	1001	adama	Male	adama@gmail.com	developer	Edit Delete
□	1002	amin	Male	amin123@gmail.com	developer	Edit Delete

Page Info: 142 / 142 Total Records: 990

RESULT employees id=999 and id=1001 are removed from the table.



SSM-CRUD

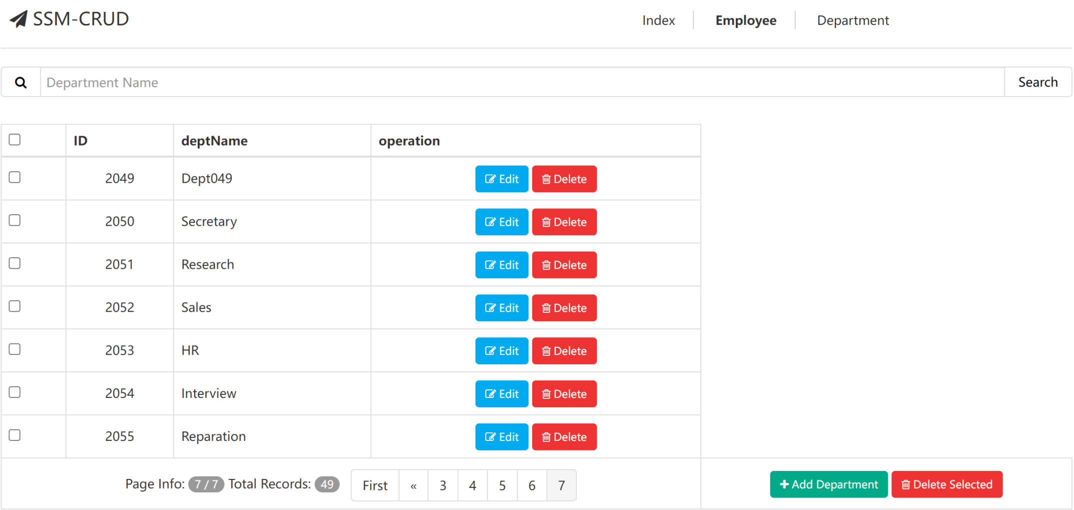
□	ID	Last Name	gender	email	deptName	operation
□	1002	amin	Male	amin123@gmail.com	developer	Edit Delete

Page Info: 142 / 142 Total Records: 988

Copy Right © Pintuer.com All Rights Reserved

DEPARTMENT

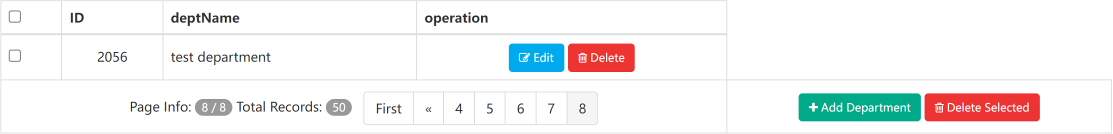
- Department Home Page ---> department.jsp. list of all departments with pagination.



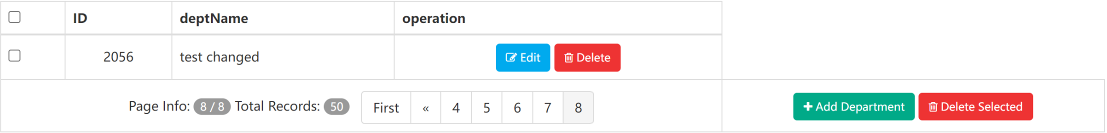
☐	ID	deptName	operation
☐	2049	Dept049	Edit Delete
☐	2050	Secretary	Edit Delete
☐	2051	Research	Edit Delete
☐	2052	Sales	Edit Delete
☐	2053	HR	Edit Delete
☐	2054	Interview	Edit Delete
☐	2055	Reparation	Edit Delete

Page Info: 7 / 7 Total Records: 49

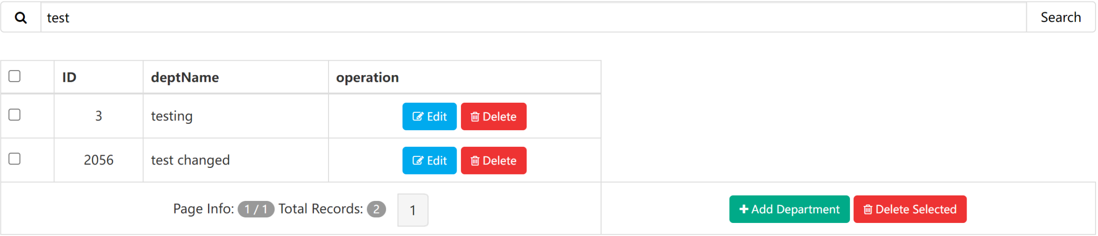
- Add Department Page: a form to register a new department.

> RESULT id=2056
>
> 
>
> | ☐ | ID | deptName | operation |
> |---|---|---|---|
> | ☐ | 2056 | test department | Edit Delete |
>
> Page Info: 8 / 8 Total Records: 50
>
>

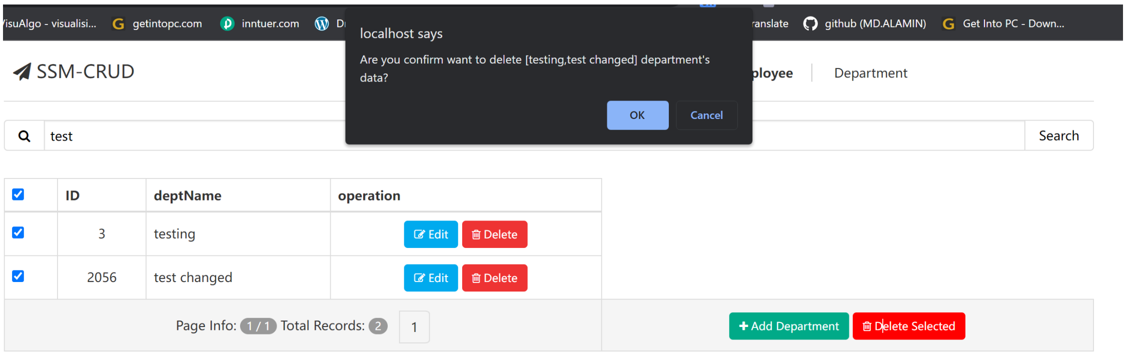
- Update Department id: 2056 name changes to test changed

> RESULT
>
> 
>
> | ☐ | ID | deptName | operation |
> |---|---|---|---|
> | ☐ | 2056 | test changed | Edit Delete |
>
> Page Info: 8 / 8 Total Records: 50
>
>

- Search Department fuzzy search key=test

> RESULT id=3 and id=2056
>
> 
>
> test
>
>
>
> | ☐ | ID | deptName | operation |
> |---|---|---|---|
> | ☐ | 3 | testing | Edit Delete |
> | ☐ | 2056 | test changed | Edit Delete |
>
> Page Info: 1 / 1 Total Records: 2
>
>

- Batch Delete Departments id: 3 and 2056 will be removed from the table.

> 
>
>
>
> localhost says
>
> Are you confirm want to delete [testing,test changed] department's data?
>
>
>
> test
>
>
>
> | ☑ | ID | deptName | operation |
> |---|---|---|---|
> | ☑ | 3 | testing | Edit Delete |
> | ☑ | 2056 | test changed | Edit Delete |
>
> Page Info: 1 / 1 Total Records: 2
>
>

PERSONAL EXPERIENCE

Before this course, I could develop simple web apps with mvc nd maven dependency tool but it was not up to date as the java EE is on the market now. But this course has given me a very rich view to:

- AOP(Aspect Oriented Programming), something that I didn't know before. but the teacher gave us more examples to demonstrate the logic.
- Spring framework: with IoC (Inversion of Control) and DI(dependency injection). to realize the loose coupling between beans which augments stability in the program, since no class is dependent directly on the other.
- MVC: The user sends a request to the server through the View layer, and the request is received by the Controller in the server. The Controller invokes the corresponding Model layer to process the request, and returns the result to the Controller, and the Controller finds the corresponding view according to the request processing result. The data is rendered and the final response is sent back to the browser
- Spring MVC: is a complete set of solutions provided by Spring for the development of presentation layer. After the evolution of presentation layer framework, such as Strust, WebWork, Strust2 and so on, SpringMVC is generally chosen as the first choice for presentation layer development in Java EE projects
- RESTFUL API: a design style and development method of network applications (**Re**presentational **S**tate **T**ransfer，It is the presentation layer resource state transfer)
- HttpMessageConverter: this is one of the most interresting plugins I learned. It is used to convert request packets into Java objects or Java objects into response packets. It provides two annotations and two entity class which are @RequestBody，@ResponseBody，RequestEntity，ResponseEntity.
- MyBatis: is a first class persistence framework with support for custom SQL, stored procedures and advanced mappings. It helps us by eliminating almost all of the JDBC code and manual setting of parameters and retrieval of results. MyBatis can use simple XML or Annotations for configuration and map primitives, Map interfaces and Java POJOs (Plain Old Java Objects) to database records.
- CRUD:
- Mybatis provides corresponding annotations for CRUD operations which have annotations: @Select @Insert @Update @Delete. it allows simplify these 4 main operations in a database.
- Spring Unit Testing and logback: these two are my favorites debugging code tools. they help log out the whole process and detailed verion of results and failures the program may encounter while running.
- Idea: I could use this IDE but I learned much more details about it. I can use database and web development tools inside.