

# CSC191 - CSU, Sacramento Fall 2013

CSC 191 @ CSU, Sacramento is the 2nd part of a "Senior Project" class.

CSC 191. Senior Project-Part II. Continuation of the group project begun in CSC 190. Teams apply software engineering principles to the design, implementation and testing of their software product. All technical work is published using guidelines modeled after IEEE documentation standards along with an appropriate user manual. Oral and written reports are required. Senior project is completed with the successful delivery, installation and demonstration of the software along with all approved documentation. Lecture one hour, laboratory three hours.

The goal of this class is to actively develop a software using industry standards.

That includes but not limited to:

Documentation  
Testing  
Development

Our Sponser, the business for whom we are developing this system is Dragonfly Hair and Nail Salon.

They have asked us to develop a scheduling system that will enable their cliental to schedule appointments. In turn, this enables their staff to accept and schedule customers as well.

## Salon Scheduler

### What is it?

---

The Salon Scheduler is an online scheduling system that gives clients the ability to schedule appointments online without calling into a salon. The system utilizes a REST API backend written in SpringMVC with MongoDB, and a HTML/CSS3/JavaScript frontend written in AngularJS

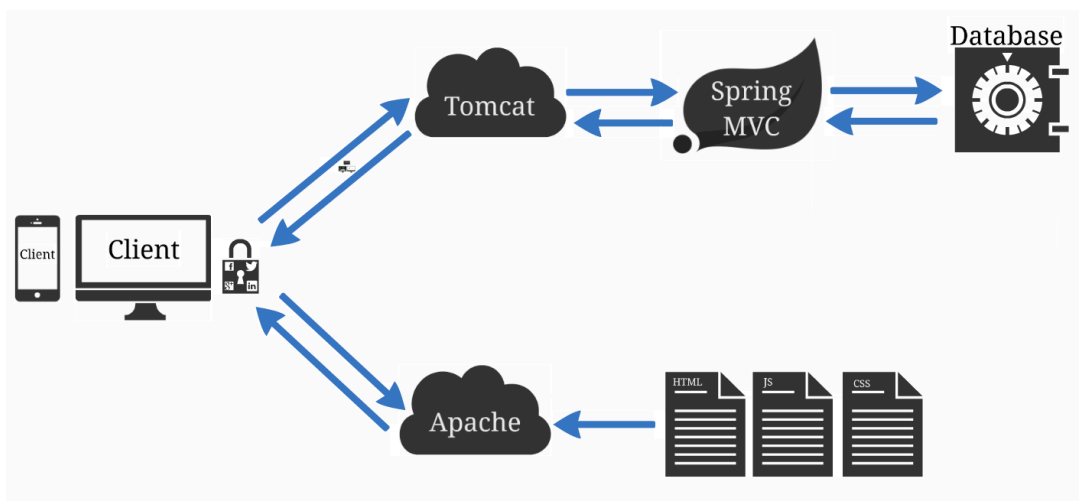
## Configuration

---

### Facebook

In order for facebook to authenticate clients on the frontend, it requires that all calls are made from the same domain.

However, the frontend code and the backend code technically live on two different servers show in the diagram below.



> The design descion for this can be found in the SDS and the SRS in the documentation folder.

Becasue of this sepration modern browsers will not allow the communication of the backend and the front end.

### Solution

During development, we utilized [mod\\_proxy](#)! which will rerout calls to example.com:8080 which is intended to hit the backend (Spring or the top layer in the diagram) example.com/api. Otherwise, modern browsers will itreperate x:8080 as a different url as just x.

Example redirect

```
ProxyPass /salon-scheduler-api/ http://localhost:8080/salon-scheduler-api/  
ProxyPassReverse /salon-scheduler-api/ http://localhost:8080/salon-scheduler-api/
```

## Installation

---

### Backend

The backend requires an application server such as Tomcat or JBoss as well as MongoDB to run. Once these have been installed, using Maven, simply run `mvn package`. This will generate a `.war` file which you can deploy to your application server.

### Frontend

The frontend only requires a webserver to run. There are no special requirements so any web server will work because it will only be serveing static files. Simply copy the frontend folder to the root document folder of your webserver and you are ready to go.