

Design Patterns with Java Spring

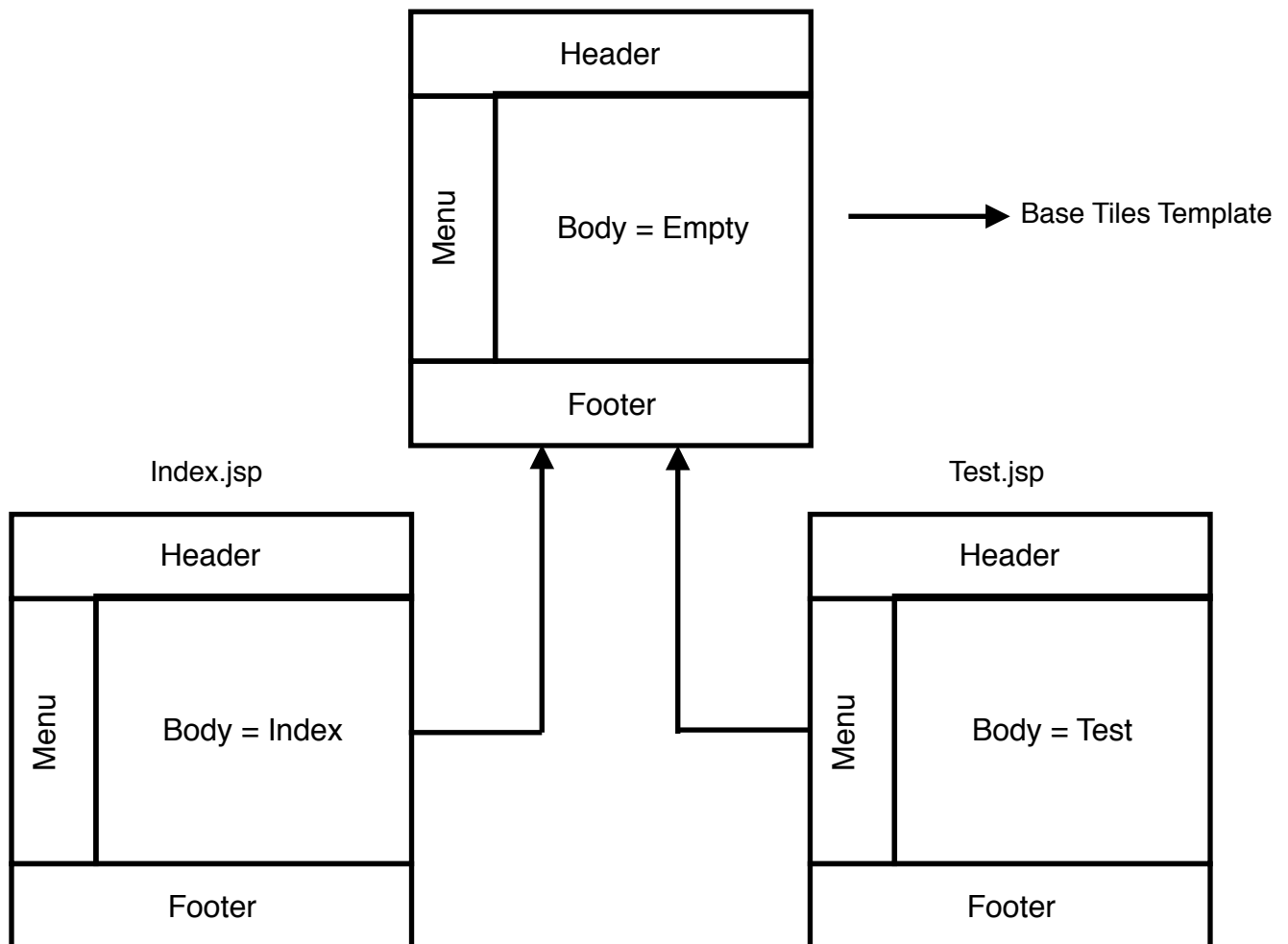
01219448 Software Patterns and Architecture

Implement a Spring Enterprise Design Pattern project with the following features :

1. Composite View Pattern with Apache Tile and Spring MVC
2. Application Controller Pattern with ViewResolver and HandlerMapping
3. Front Controller Pattern with DispatcherServlet
4. ViewHelper Pattern with Spring Tag Library or JSTL

Needed Installation :

1. Eclipse any version or Spring Tool Suite
2. Java Development Kit version 7+
3. Spring Library + Tile
4. Apache Maven (optional)



AppConfig.java : Configuration

```
package com.kundjanasith.springmvc.configuration;

import org.springframework.context.annotation.Bean;

@Configuration
@EnableWebMvc
@ComponentScan(basePackages = "com.kundjanasith.springmvc")
public class AppConfig extends WebMvcConfigurerAdapter{

    /**
     * Configure TilesConfigurer.
     */
    @Bean
    public TilesConfigurer tilesConfigurer(){
        TilesConfigurer tilesConfigurer = new TilesConfigurer();
        tilesConfigurer.setDefinitions(new String[] {"/WEB-INF/views/**/*.tiles.xml"});
        tilesConfigurer.setCheckRefresh(true);
        return tilesConfigurer;
    }

    /**
     * Configure ViewResolvers to deliver preferred views.
     */
    @Override
    public void configureViewResolvers(ViewResolverRegistry registry) {
        TilesViewResolver viewResolver = new TilesViewResolver();
        registry.viewResolver(viewResolver);
    }

    /**
     * Configure ResourceHandlers to serve static resources like CSS/ Javascript etc.
     */
    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/static/**").addResourceLocations("/static/");
    }
}

}
```

AppInitializer.java : Configuration

```
package com.kundjanasith.springmvc.configuration;

import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

public class AppInitializer extends AbstractAnnotationConfigDispatcherServletInitializer {

    @Override
    protected Class<?>[] getRootConfigClasses() {
        return new Class[] { AppConfig.class };
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return null;
    }

    @Override
    protected String[] getServletMappings() {
        return new String[] { "/" };
    }

}
```

AppController.java : Controller

```
package com.kundjanasith.springmvc.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.servlet.mvc.annotation.annotation.AnnotationMethodMapping;

@Controller
@RequestMapping("/")
public class AppController {

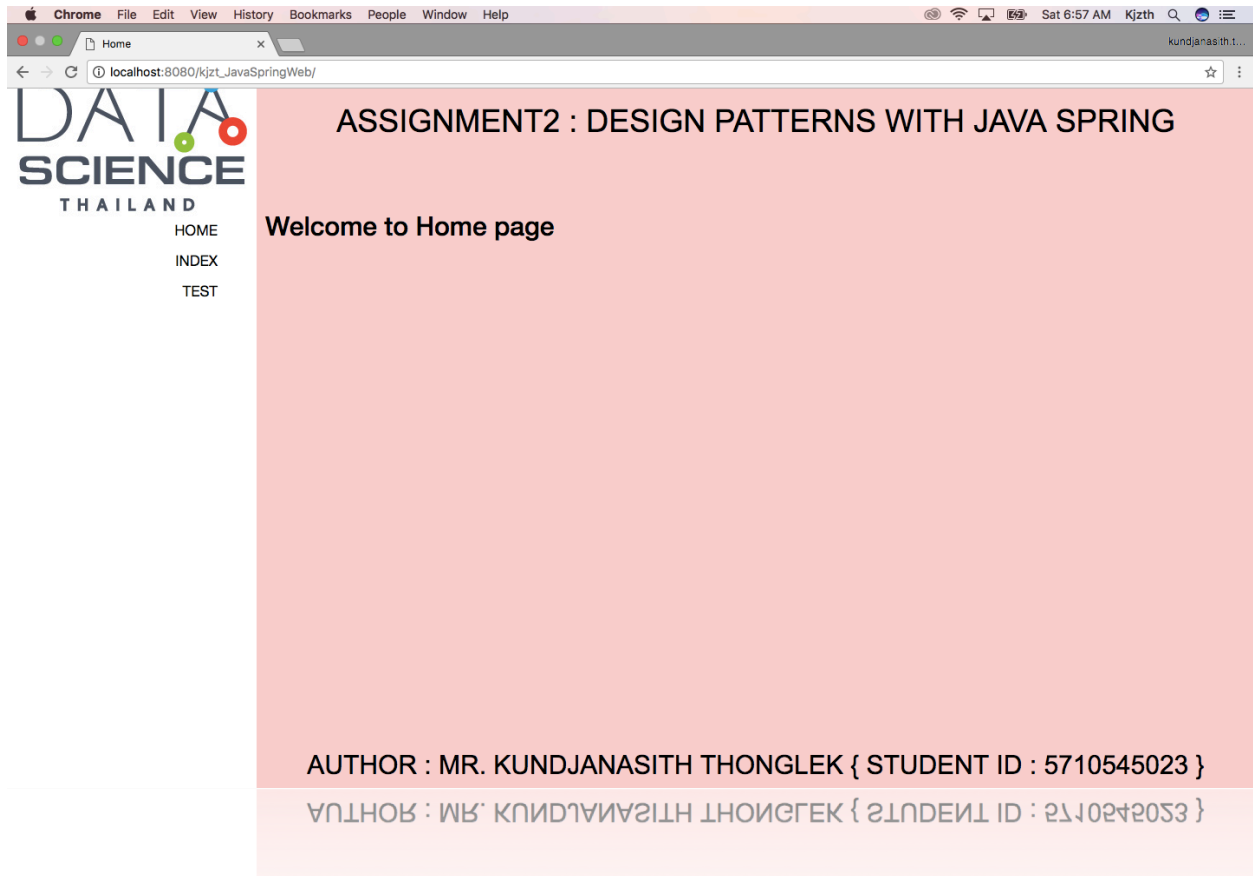
    @RequestMapping(value = { "/" }, method = RequestMethod.GET)
    public String homePage(ModelMap model) {
        return "home";
    }

    @RequestMapping(value = { "/index" }, method = RequestMethod.GET)
    public String indexPage(ModelMap model) {
        return "index";
    }

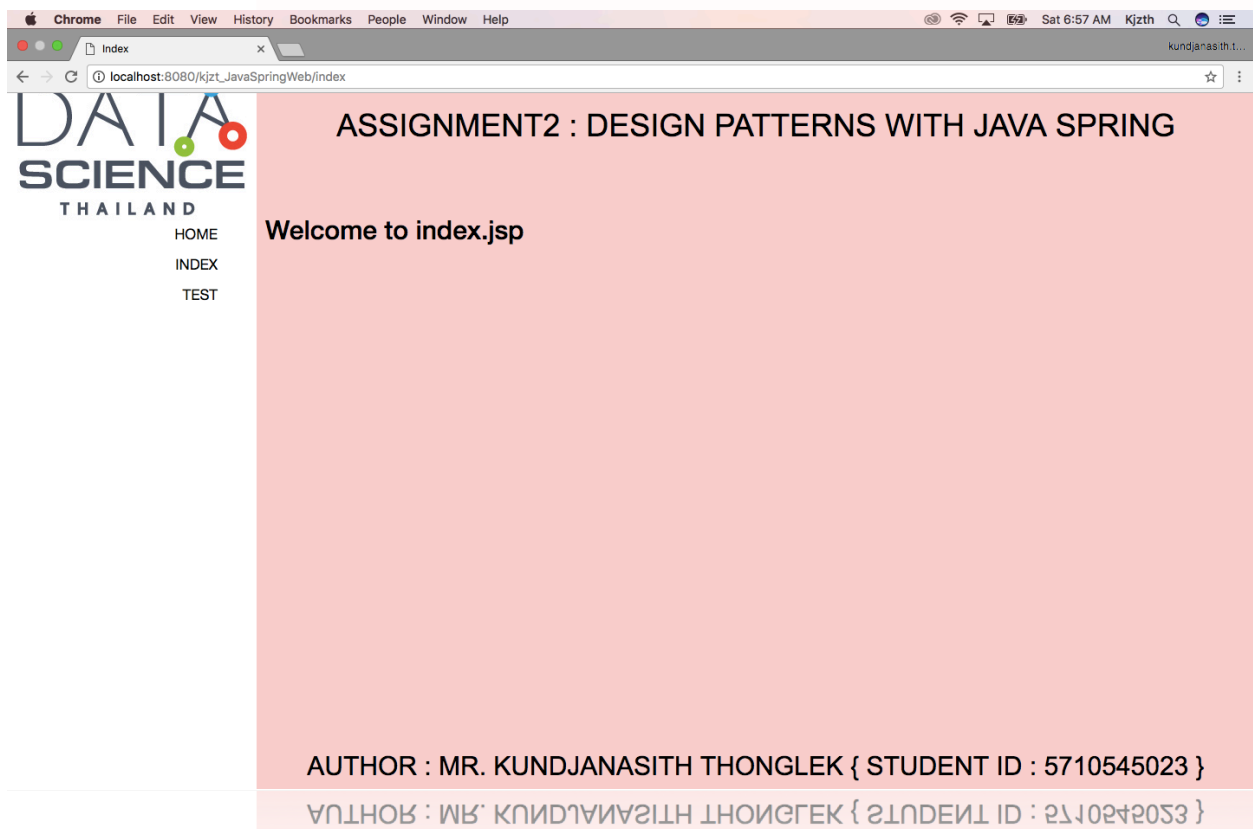
    @RequestMapping(value = { "/test" }, method = RequestMethod.GET)
    public String testPage(ModelMap model) {
        return "test";
    }

}
```

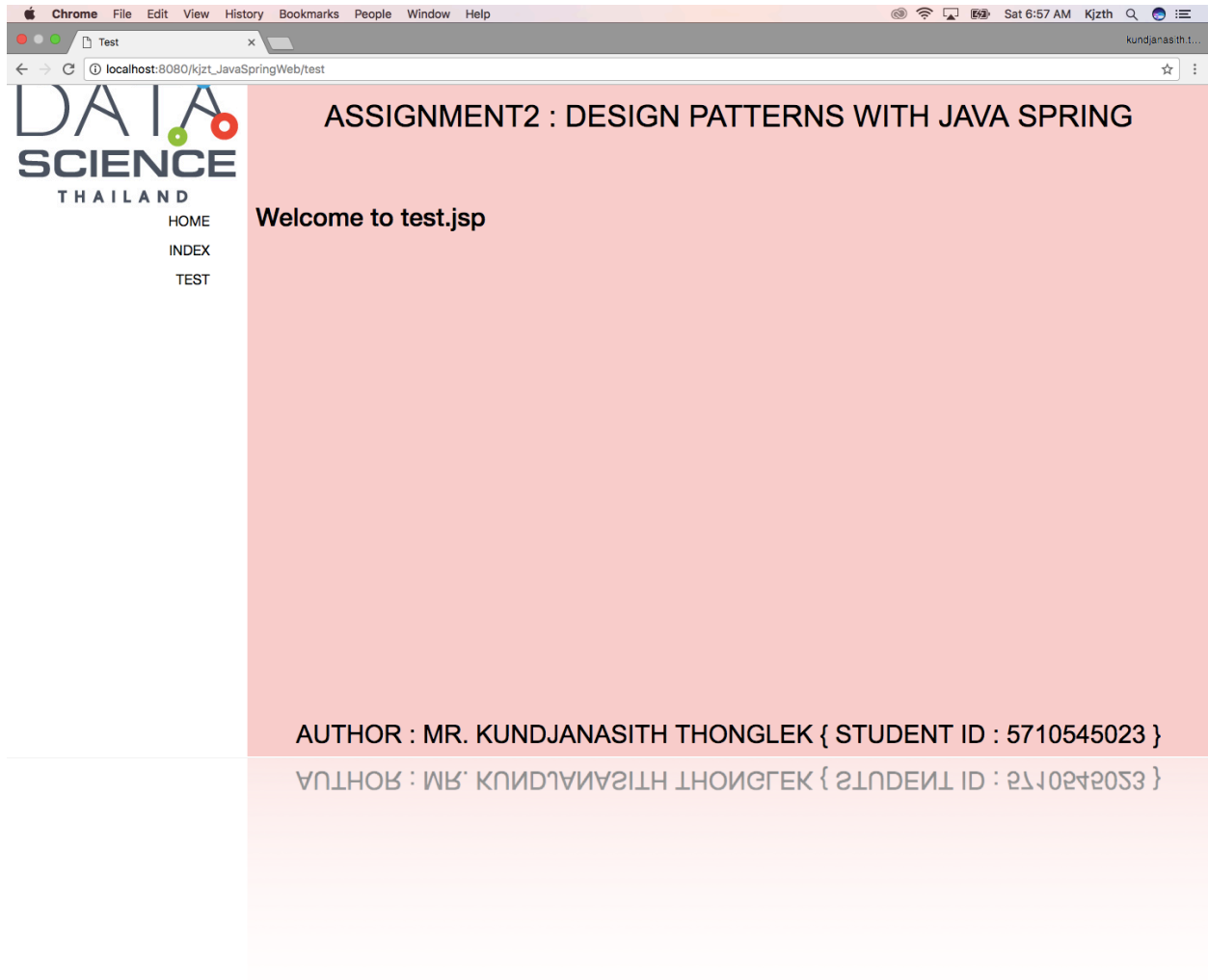
Result : home.jsp



Result : index.jsp



Result : test.jsp



Reference information for the project

Spring Tool Suit :

- <https://spring.io/tools/sts/all>

Apache Tomcat :

- <http://tomcat.apache.org/>

Apache Tile :

- <https://tiles.apache.org/>