Mr. Kundjanasith Thonglek
Student ID : 5710545023

Design Patterns with Java Spring

01219448 Software Patterns and Architecture

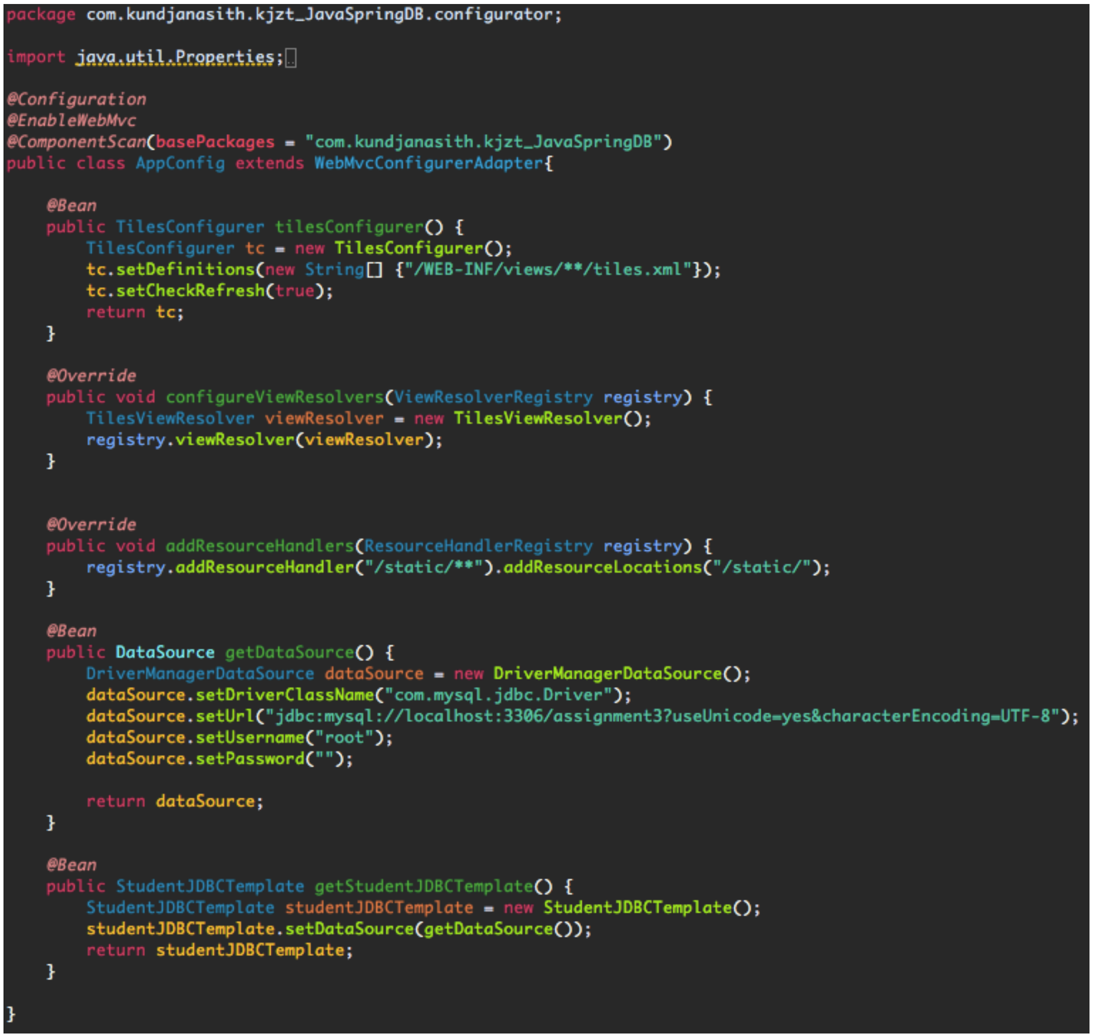
Instruction :

Implementing Spring JDBC Template using Spring Library with the following screens as the requirements.

Required Installation :

1. Eclipse any version or Spring Tool Suite
2. Java Development Kit Version 7+
3. Spring Library
4. MySQL

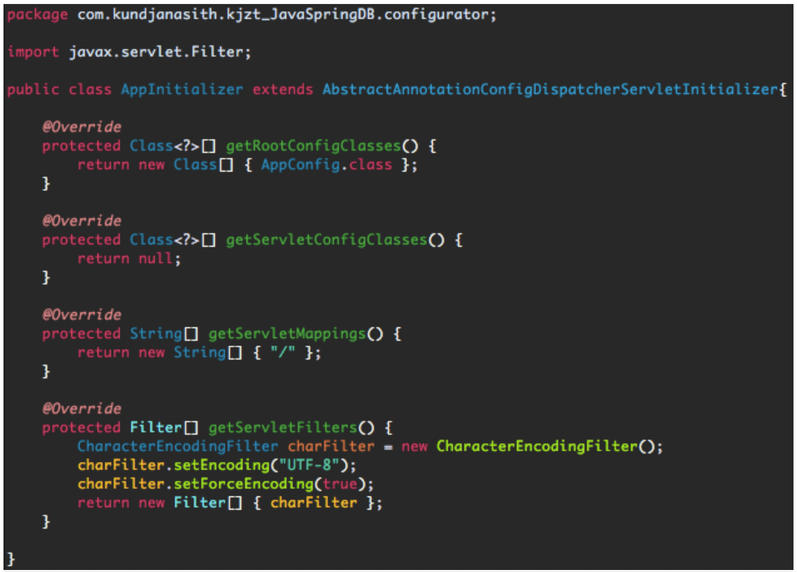
AppConfig.java : Configuration



```java
package com.kundjanasith.kjzt_JavaSpringDB.configurator;

import java.util.Properties;

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

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

    @Override
    public void configureViewResolvers(ViewResolverRegistry registry) {
        TilesViewResolver viewResolver = new TilesViewResolver();
        registry.viewResolver(viewResolver);
    }


    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/static/**").addResourceLocations("/static/");
    }

    @Bean
    public DataSource getDataSource() {
        DriverManagerDataSource dataSource = new DriverManagerDataSource();
        dataSource.setDriverClassName("com.mysql.jdbc.Driver");
        dataSource.setUrl("jdbc:mysql://localhost:3306/assignment3?useUnicode=yes&characterEncoding=UTF-8");
        dataSource.setUsername("root");
        dataSource.setPassword("");

        return dataSource;
    }

    @Bean
    public StudentJDBCTemplate getStudentJDBCTemplate() {
        StudentJDBCTemplate studentJDBCTemplate = new StudentJDBCTemplate();
        studentJDBCTemplate.setDataSource(getDataSource());
        return studentJDBCTemplate;
    }

}
```

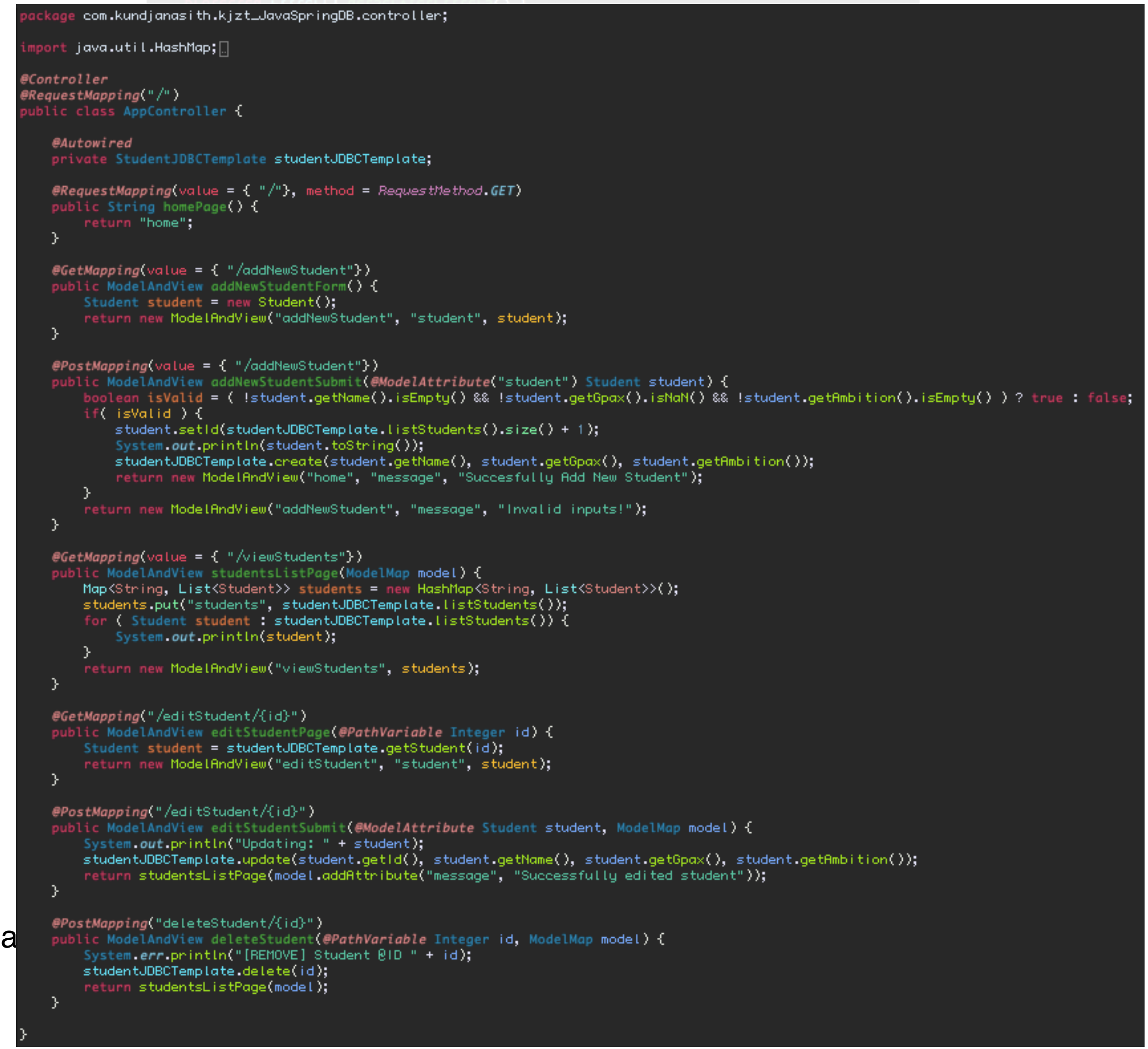
Mr. Kundjanasith Thonglek
Student ID : 5710545023

AppInitializer.java : Configuration



```java
package com.kundjanasith.kjzt_JavaSpringDB.configurator;

import javax.servlet.Filter;

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[] { "/" };
    }

    @Override
    protected Filter[] getServletFilters() {
        CharacterEncodingFilter charFilter = new CharacterEncodingFilter();
        charFilter.setEncoding("UTF-8");
        charFilter.setForceEncoding(true);
        return new Filter[] { charFilter };
    }

}
```

AppController.java : Controller

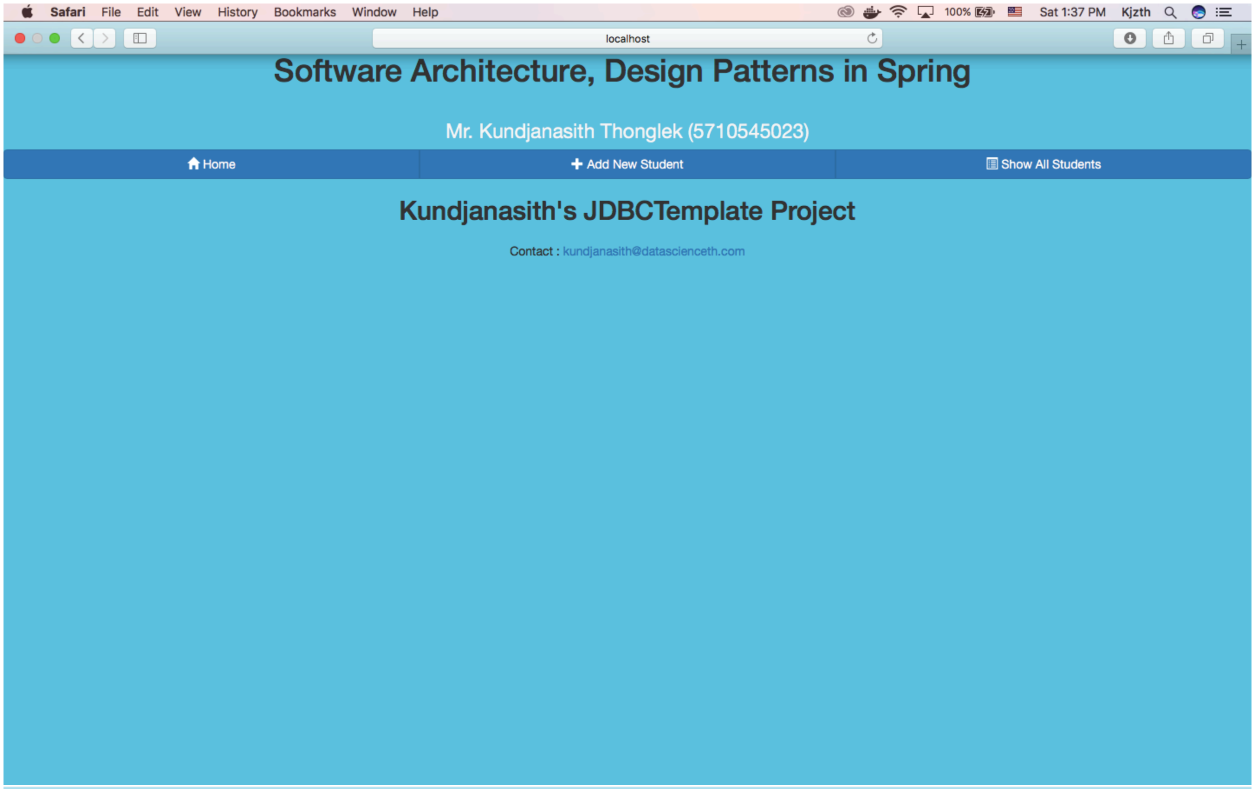


```java
package com.kundjanasith.kjzt_JavaSpringDB.controller;

import java.util.HashMap;

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

    @Autowired
    private StudentJDBCTemplate studentJDBCTemplate;

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

    @GetMapping(value = { "/addNewStudent"})
    public ModelAndView addNewStudentForm() {
        Student student = new Student();
        return new ModelAndView("addNewStudent", "student", student);
    }

    @PostMapping(value = { "/addNewStudent"})
    public ModelAndView addNewStudentSubmit(@ModelAttribute("student") Student student) {
        boolean isValid = ( !student.getName().isEmpty() && !student.getGpax().isNaN() && !student.getAmbition().isEmpty() ) ? true : false;
        if( isValid ) {
            student.setId(studentJDBCTemplate.listStudents().size() + 1);
            System.out.println(student.toString());
            studentJDBCTemplate.create(student.getName(), student.getGpax(), student.getAmbition());
            return new ModelAndView("home", "message", "Succesfully Add New Student");
        }
        return new ModelAndView("addNewStudent", "message", "Invalid inputs!");
    }

    @GetMapping(value = { "/viewStudents"})
    public ModelAndView studentsListPage(ModelMap model) {
        Map<String, List<Student>> students = new HashMap<String, List<Student>>();
        students.put("students", studentJDBCTemplate.listStudents());
        for ( Student student : studentJDBCTemplate.listStudents()) {
            System.out.println(student);
        }
        return new ModelAndView("viewStudents", students);
    }

    @GetMapping("/editStudent/{id}")
    public ModelAndView editStudentPage(@PathVariable Integer id) {
        Student student = studentJDBCTemplate.getStudent(id);
        return new ModelAndView("editStudent", "student", student);
    }

    @PostMapping("/editStudent/{id}")
    public ModelAndView editStudentSubmit(@ModelAttribute Student student, ModelMap model) {
        System.out.println("Updating: " + student);
        studentJDBCTemplate.update(student.getId(), student.getName(), student.getGpax(), student.getAmbition());
        return studentsListPage(model.addAttribute("message", "Successfully edited student"));
    }

    @PostMapping("deleteStudent/{id}")
    public ModelAndView deleteStudent(@PathVariable Integer id, ModelMap model) {
        System.err.println("[REMOVE] Student @ID " + id);
        studentJDBCTemplate.delete(id);
        return studentsListPage(model);
    }

}
```

Softwa

Mr. Kundjanasith Thonglek
Student ID : 5710545023

Result : index.jsp



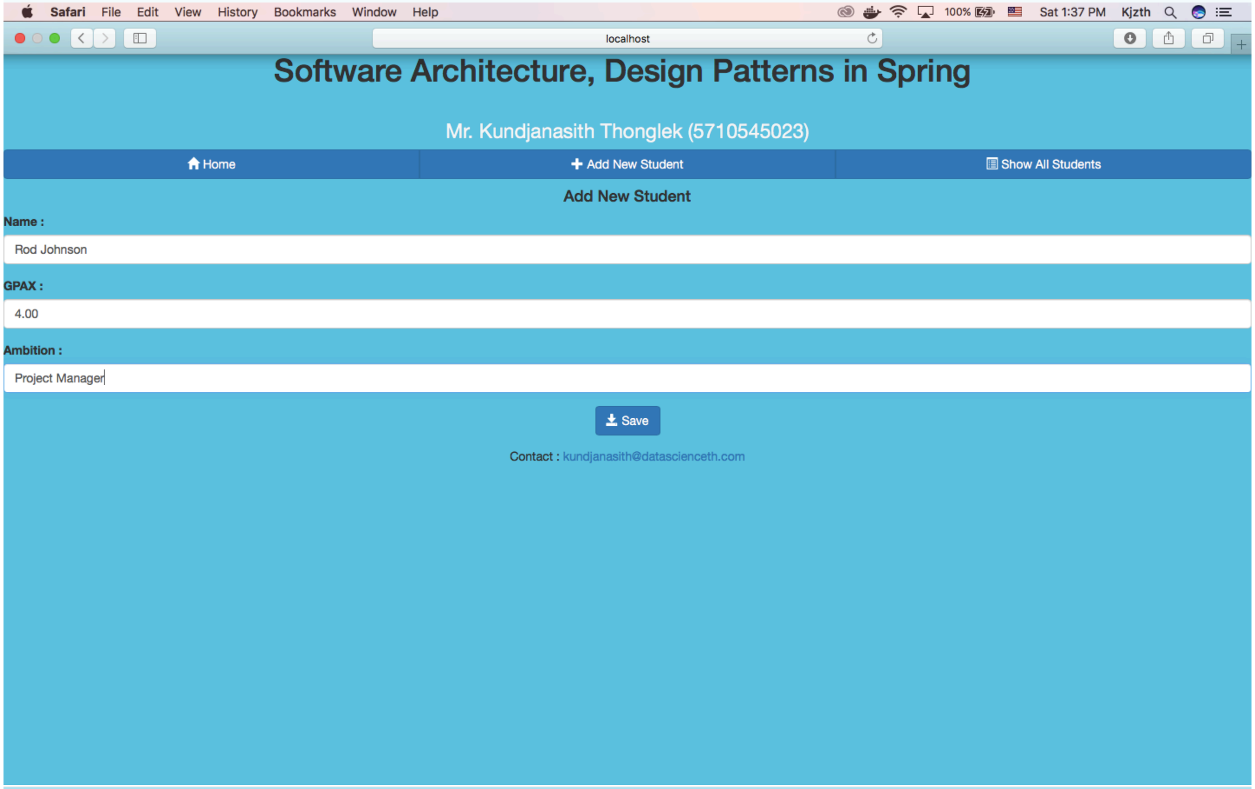
Software Architecture, Design Patterns in Spring

Mr. Kundjanasith Thonglek (5710545023)

Kundjanasith's JDBCTemplate Project

Contact : kundjanasith@datascienceth.com

Result : addNewStudent.jsp



Software Architecture, Design Patterns in Spring

Mr. Kundjanasith Thonglek (5710545023)

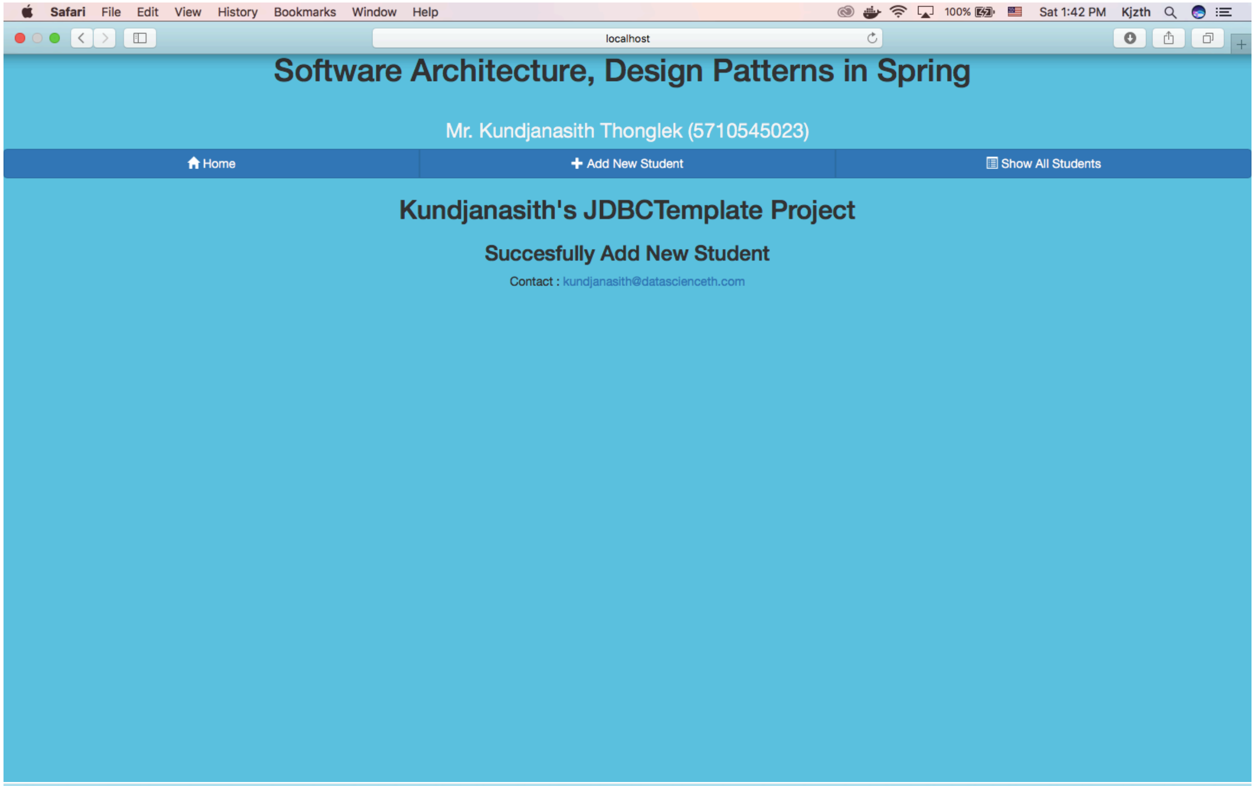
Add New Student

Name :
Rod Johnson

GPAX :
4.00

Ambition :
Project Manager

Contact : kundjanasith@datascienceth.com



Software Architecture, Design Patterns in Spring

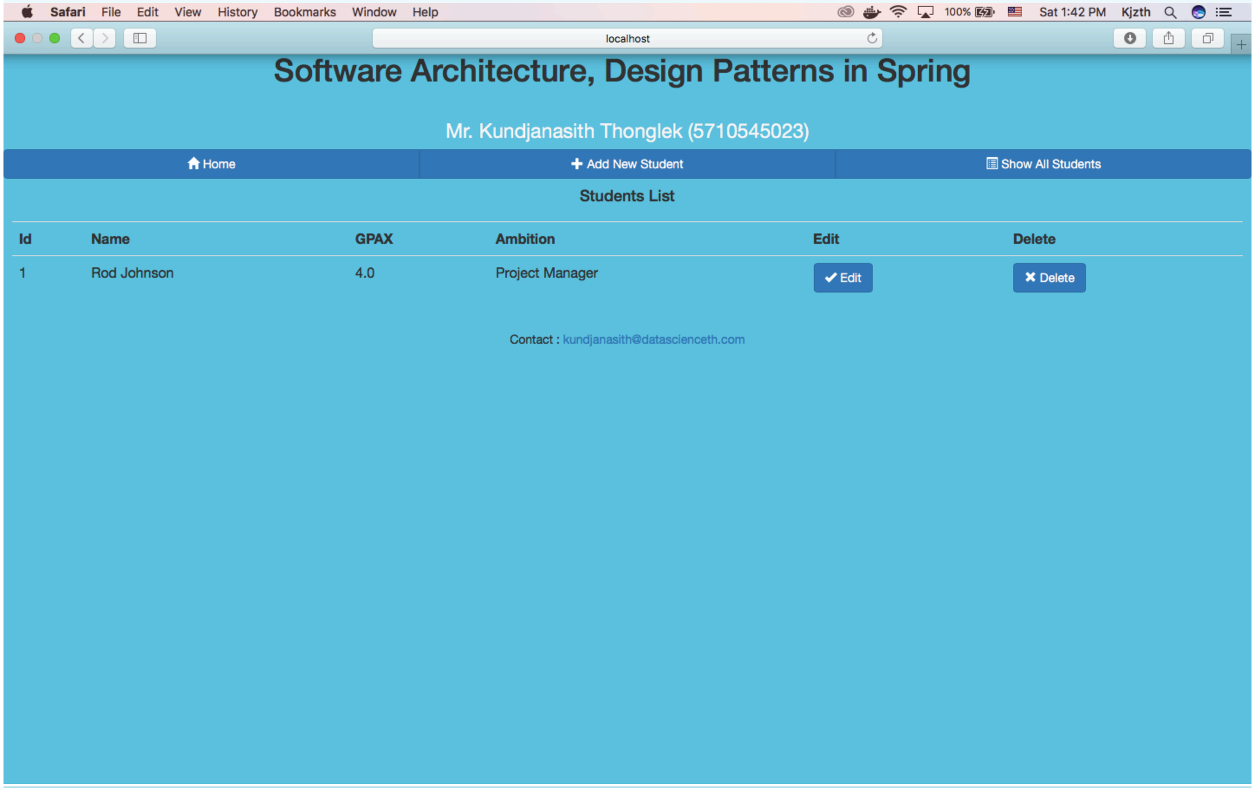
Mr. Kundjanasith Thonglek (5710545023)

Kundjanasith's JDBCTemplate Project

Succesfully Add New Student

Contact : kundjanasith@datascienceth.com

Result : viewStudent.jsp



Software Architecture, Design Patterns in Spring

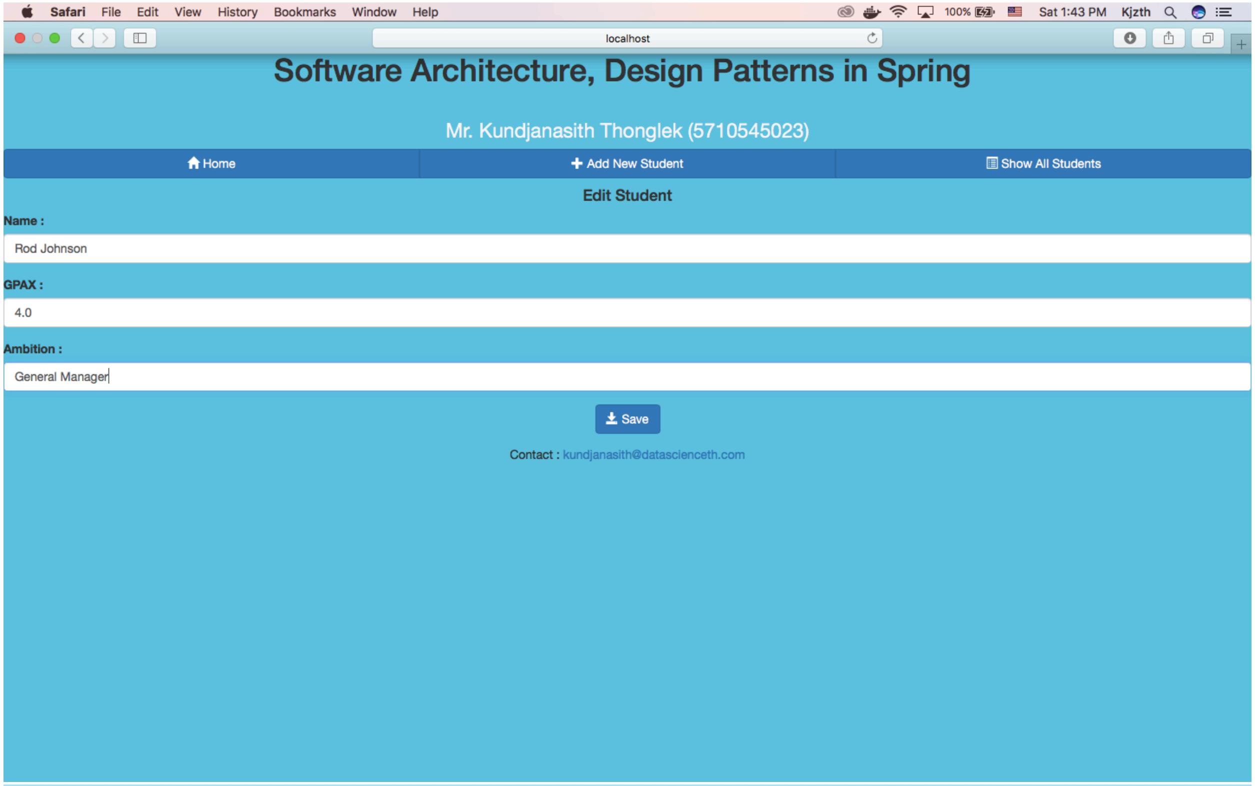
Mr. Kundjanasith Thonglek (5710545023)

Students List

Id	Name	GPAX	Ambition	Edit	Delete
1	Rod Johnson	4.0	Project Manager	✓ Edit	✖ Delete

Contact : kundjanasith@datascienceth.com

Result : editStudent.jsp



Software Architecture, Design Patterns in Spring

Mr. Kundjanasith Thonglek (5710545023)

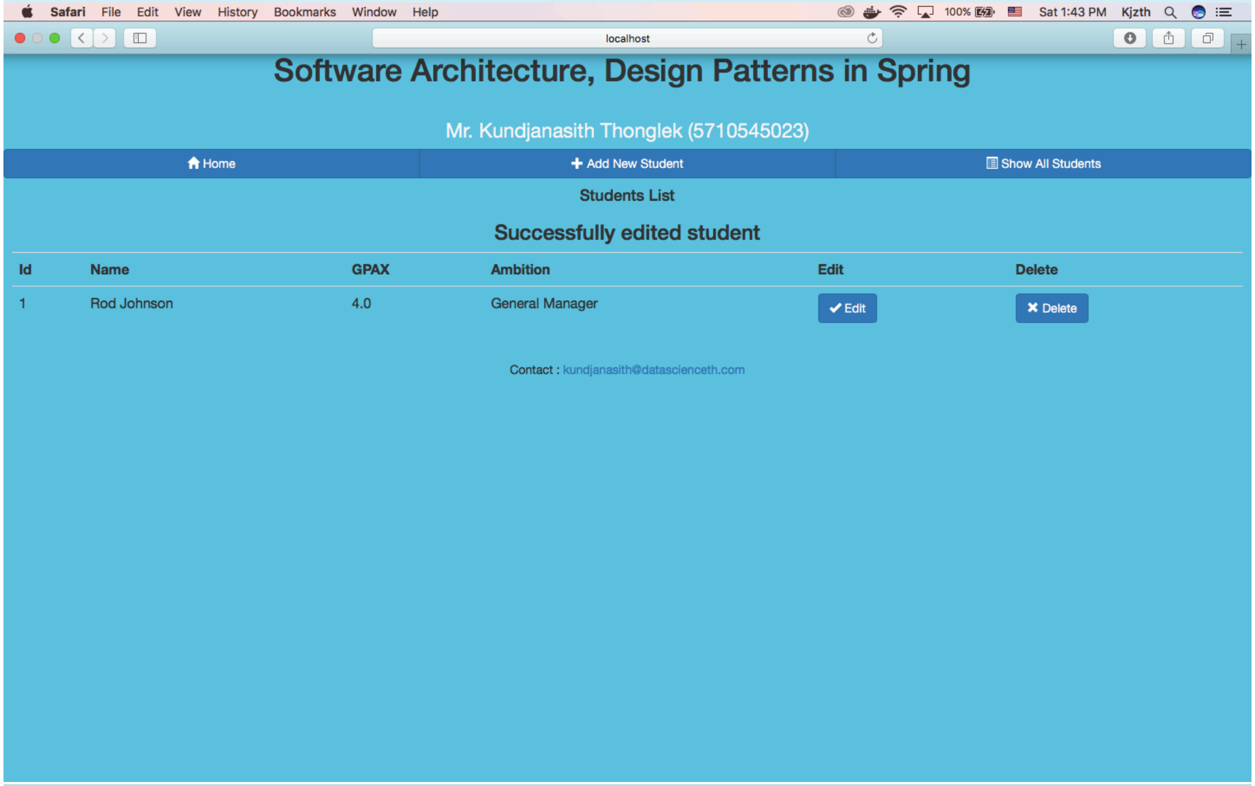
Edit Student

Name : Rod Johnson

GPAX : 4.0

Ambition : General Manager

Contact : kundjanasith@datascienceth.com



Software Architecture, Design Patterns in Spring

Mr. Kundjanasith Thonglek (5710545023)

Students List

Successfully edited student

Id	Name	GPAX	Ambition	Edit	Delete
1	Rod Johnson	4.0	General Manager	✓ Edit	✖ Delete

Contact : kundjanasith@datascienceth.com

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/