**Developing a Backend Admin for Learner’s AcademySpecification document**

Product’s capabilities, User interactions:

* In this application admin can list the all the subjects for all the classes
* In this application admin can list of all the teachers
* In this application admin can list of all the classes
* A teacher is assigned to different classes for different subjects.
* Admin can list of students (Each student must be assigned to a single class)

**Number and duration of sprints required :**

3 sprints, each of 5 days.

Sprint #1: Planning and Requirement analysis

This stage deals with planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Sprint #2:Design and develop

Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification

JDBC used to connect mysql database.

Sprint #3:Testing

This stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

DATABASE:MYSQL

CONECTIVITY:JDBC

USERNAME:admin

PASSWORD:1234

**GitHub account**

<https://github.com/chandangowdatp/java-fsd.git>

**Flow:**

1. Admin loigin
2. If user id is: **admin** and password is :**1234** then goto step3 other wise goto step 1.
3. Dashbord containg navigation bar for student list,classes list, subjects list, teachers list and logout.
4. In classes page admin can we the students for each subjects and teachers.

**Code:**

1. **AdminServlet**

**import** java.io.IOException;

**import** java.util.List;

**import** javax.annotation.Resource;

**import** javax.servlet.RequestDispatcher;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.Cookie;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** javax.sql.DataSource;

**import** com.clases.Class;

**import** com.clases.Student;

**import** com.clases.Subject;

**import** com.clases.Teacher;

**import** javax.servlet.\*;

@WebServlet("/AdminServlet")

**public** **class** AdminServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** ManageData ManageData;

@Resource(name = "jdbc\_database")

**private** DataSource datasource;

@Override

**public** **void** init() **throws** ServletException {

**super**.init();

**try** {

ManageData = **new** ManageData(datasource);

}

**catch** (Exception e) {

**throw** **new** ServletException(e);

}

}

**public** AdminServlet() {

**super**();

}

@Override

**protected** **void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {

doGet(req, resp);

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

String command = request.getParameter("command");

**if** (command == **null**) {

command = "CLASSES";

}

**if** (!getCookies(request, response) && (!command.equals("LOGIN"))) {

response.sendRedirect("login.jsp");

}

**else** {

**switch** (command) {

**case** "STUDENTS":

studentsList(request, response);

**break**;

**case** "TEACHERS":

teachersList(request, response);

**break**;

**case** "SUBJECTS":

subjectList(request, response);

**break**;

**case** "CLASSES":

classestList(request, response);

**break**;

**case** "ST\_LIST":

classStudentsList(request, response);

**break**;

**case** "LOGIN":

login(request, response);

**break**;

**default**:

classestList(request, response);

}

}

} **catch** (Exception e) {

**throw** **new** ServletException(e);

}

}

**private** **void** studentsList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

List<Student> students = ManageData.getStudents();

request.setAttribute("STUDENT\_LIST", students);

RequestDispatcher dispatcher = request.getRequestDispatcher("/studentsList.jsp");

dispatcher.forward(request, response);

}

**private** **void** teachersList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

List<Teacher> teachers = ManageData.getTeachers();

request.setAttribute("TEACHERS\_LIST", teachers);

RequestDispatcher dispatcher = request.getRequestDispatcher("/teachersList.jsp");

dispatcher.forward(request, response);

}

**private** **void** subjectList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

List<Subject> subjects = ManageData.getSubjects();

request.setAttribute("SUBJECTS\_LIST", subjects);

RequestDispatcher dispatcher = request.getRequestDispatcher("/subjectsList.jsp");

dispatcher.forward(request, response);

}

**private** **void** classestList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

List<Class> classes = ManageData.getClasses();

request.setAttribute("CLASSES\_LIST", classes);

RequestDispatcher dispatcher = request.getRequestDispatcher("/classesList.jsp");

dispatcher.forward(request, response);

}

**private** **void** login(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

String username = request.getParameter("username");

String password = request.getParameter("password");

**if** (username.toLowerCase().equals("admin") && password.toLowerCase().equals("1234")) {

Cookie cookie = **new** Cookie(username, password);

HttpSession session = request.getSession(**true**);

session.setAttribute("username", username);

cookie.setMaxAge(86400);

response.addCookie(cookie);

classestList(request, response);

}

**else** {

RequestDispatcher dispatcher = request.getRequestDispatcher("login.jsp");

dispatcher.forward(request, response);

}

}

**private** **void** classStudentsList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

**int** classId = Integer.*parseInt*(request.getParameter("classId"));

String section = request.getParameter("section");

String subject = request.getParameter("subject");

List<Student> students = ManageData.loadClassStudents(classId);

request.setAttribute("STUDENTS\_LIST", students);

request.setAttribute("SECTION", section);

request.setAttribute("SUBJECT", subject);

RequestDispatcher dispatcher = request.getRequestDispatcher("/classStudentsList.jsp");

dispatcher.forward(request, response);

}

**private** **boolean** getCookies(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

**boolean** check = **false**;

Cookie[] cookies = request.getCookies();

**try** {

**for** (Cookie cookie : cookies) {

**if** (cookie.getName().equals("admin") && cookie.getValue().equals("1234")) {

check = **true**;

**break**;

}

}

}

**catch** (Exception e) {

String site = **new** String("login.jsp");

response.setStatus(response.***SC\_MOVED\_TEMPORARILY***);

response.setHeader("Location", site);

**return** **false**;

}

**return** check;

}

}

1. **ManageData.class**

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.sql.DataSource;

**import** com.clases.Class;

**import** com.clases.Student;

**import** com.clases.Subject;

**import** com.clases.Teacher;

**public** **class** ManageData {

**private** DataSource dataSource;

**public** ManageData(DataSource dataSource) {

**this**.dataSource = dataSource;

}

**public** List<Student> getStudents() {

List<Student> students = **new** ArrayList<>();

Connection myConn = **null**;

Statement myStmt = **null**;

ResultSet myRs = **null**;

**try** {

myConn = dataSource.getConnection();

String sql = "SELECT \* FROM students";

myStmt = myConn.createStatement();

myRs = myStmt.executeQuery(sql);

**while** (myRs.next()) {

**int** id = myRs.getInt("id");

String firstName = myRs.getString("fname");

String lastName = myRs.getString("lname");

**int** aclass = myRs.getInt("class");

Student tempStudent = **new** Student(id, firstName, lastName, aclass);

students.add(tempStudent);

}

} **catch** (Exception e) {

System.***out***.println("Exception: " + e);

} **finally** {

close(myConn, myStmt, myRs);

}

**return** students;

}

**public** List<Teacher> getTeachers() {

List<Teacher> teachers = **new** ArrayList<>();

Connection myConn = **null**;

Statement myStmt = **null**;

ResultSet myRs = **null**;

**try** {

myConn = dataSource.getConnection();

String sql = "SELECT \* FROM teachers";

myStmt = myConn.createStatement();

myRs = myStmt.executeQuery(sql);

**while** (myRs.next()) {

**int** id = myRs.getInt("id");

String firstName = myRs.getString("fname");

String lastName = myRs.getString("lname");

Teacher temp = **new** Teacher(id, firstName, lastName);

teachers.add(temp);

}

} **catch** (Exception e) {

System.***out***.println("Exception: " + e);

}

**finally** {

close(myConn, myStmt, myRs);

}

**return** teachers;

}

**public** List<Subject> getSubjects() {

List<Subject> subjects = **new** ArrayList<>();

Connection myConn = **null**;

Statement myStmt = **null**;

ResultSet myRs = **null**;

**try** {

myConn = dataSource.getConnection();

String sql = "SELECT \* FROM subjects";

myStmt = myConn.createStatement();

myRs = myStmt.executeQuery(sql);

**while** (myRs.next()) {

**int** id = myRs.getInt("id");

String name = myRs.getString("name");

String shortcut = myRs.getString("shortcut");

Subject temp = **new** Subject(id, name,shortcut);

subjects.add(temp);

}

} **catch** (Exception e) {

System.***out***.println("Exception: " + e);

}

**finally** {

close(myConn, myStmt, myRs);

}

**return** subjects;

}

**public** List<Class> getClasses() {

List<Class> classes = **new** ArrayList<>();

Connection myConn = **null**;

Statement myStmt = **null**;

ResultSet myRs = **null**;

**try** {

myConn = dataSource.getConnection();

String sql = "SELECT \* FROM classes";

myStmt = myConn.createStatement();

myRs = myStmt.executeQuery(sql);

**while** (myRs.next()) {

**int** id = myRs.getInt("id");

**int** section = myRs.getInt("section");

**int** subject = myRs.getInt("subject");

**int** teacher = myRs.getInt("teacher");

String time = myRs.getString("time");

Teacher tempTeacher = loadTeacher(teacher);

Subject tempSubject = loadSubject(subject);

String teacher\_name = tempTeacher.getFname() + " " + tempTeacher.getLname();

Class temp = **new** Class(id, section, teacher\_name, tempSubject.getName(), time);

classes.add(temp);

}

} **catch** (Exception e) {

System.***out***.println("Exception: " + e);

}

**finally** {

close(myConn, myStmt, myRs);

}

**return** classes;

}

**public** Teacher loadTeacher(**int** teacherId) {

Teacher theTeacher = **null**;

Connection myConn = **null**;

Statement myStmt = **null**;

ResultSet myRs = **null**;

**try** {

myConn = dataSource.getConnection();

String sql = "SELECT \* FROM teachers WHERE id = " + teacherId;

myStmt = myConn.createStatement();

myRs = myStmt.executeQuery(sql);

**while** (myRs.next()) {

**int** id = myRs.getInt("id");

String fname = myRs.getString("fname");

String lname = myRs.getString("lname");

theTeacher = **new** Teacher(id, fname, lname);

}

}

**catch** (Exception e) {

System.***out***.println("Exception: " + e);

}

**finally** {

close(myConn, myStmt, myRs);

}

**return** theTeacher;

}

**public** Subject loadSubject(**int** subjectId) {

Subject theSubject = **null**;

Connection myConn = **null**;

Statement myStmt = **null**;

ResultSet myRs = **null**;

**try** {

myConn = dataSource.getConnection();

String sql = "SELECT \* FROM subjects WHERE id = " + subjectId;

myStmt = myConn.createStatement();

myRs = myStmt.executeQuery(sql);

**while** (myRs.next()) {

**int** id = myRs.getInt("id");

String name = myRs.getString("name");

String shortcut = myRs.getString("shortcut");

theSubject = **new** Subject(id, name,shortcut);

}

}

**catch** (Exception e) {

System.***out***.println("Exception: " + e);

}

**finally** {

close(myConn, myStmt, myRs);

}

**return** theSubject;

}

**public** Class loadClass(**int** classId) {

Class theClass = **null**;

Connection myConn = **null**;

Statement myStmt = **null**;

ResultSet myRs = **null**;

**try** {

myConn = dataSource.getConnection();

String sql = "SELECT \* FROM clasess WHERE id = " + classId;

myStmt = myConn.createStatement();

myRs = myStmt.executeQuery(sql);

**while** (myRs.next()) {

**int** id = myRs.getInt("id");

**int** section = myRs.getInt("section");

**int** subject = myRs.getInt("subject");

**int** teacher = myRs.getInt("teacher");

String time = myRs.getString("time");

Teacher tempTeacher = loadTeacher(teacher);

Subject tempSubject = loadSubject(subject);

String teacher\_name = tempTeacher.getFname() + " " + tempTeacher.getLname();

}

}

**catch** (Exception e) {

System.***out***.println("Exception: " + e);

} **finally** {

close(myConn, myStmt, myRs);

}

**return** theClass;

}

**public** List<Student> loadClassStudents(**int** classId) {

List<Student> students = **new** ArrayList<>();

Connection myConn = **null**;

Statement myStmt = **null**;

ResultSet myRs = **null**;

**try** {

myConn = dataSource.getConnection();

String sql = "SELECT \* FROM students WHERE class = " + classId;

myStmt = myConn.createStatement();

myRs = myStmt.executeQuery(sql);

**while** (myRs.next()) {

**int** id = myRs.getInt("id");

String firstName = myRs.getString("fname");

String lastName = myRs.getString("lname");

**int** aclass = myRs.getInt("class");

Student tempStudent = **new** Student(id, firstName, lastName, aclass);

students.add(tempStudent);

}

}

**catch** (Exception e) {

System.***out***.println("Exception: " + e);

}

**finally** {

close(myConn, myStmt, myRs);

}

**return** students;

}

**private** **void** close(Connection myConn, Statement myStmt, ResultSet myRs) {

**try** {

**if** (myRs != **null**) {

myRs.close();

}

**if** (myStmt != **null**) {

myStmt.close();

}

**if** (myConn != **null**) {

myConn.close();

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}