**Learners Academy – Specification Document**

This document contains sections for:

* [Sprint planning and Task completion](#Sprint_plan)
* [Core concepts used in project](#Core_concepts)
* [Flow of the](#Flow) Project
* [Demonstrating the project code and its output.](#Product_capability)
* [Unique Selling Points of the Application](#USP)

The code for this project is located at https://github.com/\*\*\*\*\*\*\*\*\*\*

The project is developed by Santhi mogili.

## **Sprints planning and Task completion**

The project is planned to be complete the project in 2 sprints.

Tasks to be completed in the sprint are:

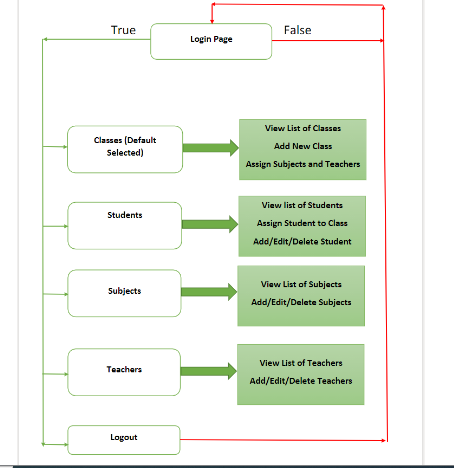
* Creating the flow of the application.
* Writing the Java program to fulfill the requirements of the project.
* Testing the Java program with different kinds of User input.
* Pushing code to GitHub.
* Creating this specification document to show project code and output of project.

## **Core concepts used in project**

Hibernate, DbUtil connections, webapp pages like html,jsps,xml, Classes setters & getters, Servlet classes, Tomcat apache server 9.0, MySQL database to create tables.

## 

## **Flow of the Application**



## **Demonstrating the project code and output of the code.**

To demonstrate the project code, below are the sub-sections configured to highlight output of the project:

1. [**Creating the project, package , class and webpages in webapp in Eclipse**](#Step_1)
2. **Writing a program in Java for Lockers Pvt Ltd app .**
3. [**Pushing the code to GitHub repository**](#Step_6)

## **Step 1: Creating a new project, package and class in Eclipse**

* Open Eclipse
* Go to File -> New -> Project -> Dynamic Web Project -> Next.
* Type in **LearnersAcademyPhase2** Project in project name and click on “Finish.”
* Select your project and go to File -> New -> Package.
* Enter **com.simplilearn.dao, com.simplilearn.DbUtil, com.simplilearn.pojo** Packages and click on “Finish.”
* Select your respective package and go to File -> New ->Class
* EnterClass names according to requirement and click on “Finish.”

## **Step 2: Writing a program in Java for Lockers Pvt Ltd app .**

**In package dao classes created are**

**2.1** package com.simplilearn.dao;

public class AdminControllerDAO {

public int ChooseOption(){

return 0;

}

}

**2.2** package com.simplilearn.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import com.simplilearn.dbutil.DbUtil;

import com.simplilearn.pojo.Classes;

import com.simplilearn.pojo.Student;

import com.simplilearn.pojo.Subject;

import com.simplilearn.pojo.Teachers;

public class DbRetrieveDAO {

public List<Student> display() throws ClassNotFoundException, SQLException

{

Connection con=DbUtil.dbConn();

if(con!=null) {

System.out.println("connection with dB is established ");

}

else {

System.out.println("connection failed ");

}

List<Student> list = new ArrayList<>();

String sql="select \* from Student";

PreparedStatement st=con.prepareStatement(sql);

//address of the table if it exists

ResultSet rs=st.executeQuery();

while(rs.next()) {

Student s = new Student();

s.setid(rs.getInt(1));

s.setSname(rs.getString(2));

s.setSage(rs.getInt(3));

s.setSclass(rs.getInt(4));

list.add(s);

}

return list;

}

public List<Subject> displaysub() throws ClassNotFoundException, SQLException

{

Connection con=DbUtil.dbConn();

if(con!=null) {

System.out.println("connection with dB is established ");

}

else {

System.out.println("connection failed ");

}

List<Subject> listSub= new ArrayList<>();

String sqls="select \* from Subjects";

PreparedStatement sts=con.prepareStatement(sqls);

//address of the table if it exists

ResultSet rss=sts.executeQuery();

while(rss.next()) {

Subject sub = new Subject();

sub.setId(rss.getInt(1));

sub.setName(rss.getString(2));

listSub.add(sub);

}

return listSub;

}

public List<Teachers> displayTeachers() throws ClassNotFoundException, SQLException

{

Connection con=DbUtil.dbConn();

if(con!=null) {

System.out.println("connection with dB is established ");

}

else {

System.out.println("connection failed ");

}

List<Teachers> listT = new ArrayList<>();

String sqlt="select \* from Teachers";

PreparedStatement stt=con.prepareStatement(sqlt);

//address of the table if it exists

ResultSet rst=stt.executeQuery();

while(rst.next()) {

Teachers t = new Teachers();

t.setId(rst.getInt(1));

t.setName(rst.getString(2));

t.setAge(rst.getInt(3));

listT.add(t);

}

return listT;

}

public List<Classes> displayClass(int query) throws ClassNotFoundException, SQLException

{

Connection con=DbUtil.dbConn();

if(con!=null) {

System.out.println("connection with dB is established ");

}

else {

System.out.println("connection failed ");

}

List<Classes> listC = new ArrayList<>();

if(query == 1) {

String sqlc="select \* from Classes";

PreparedStatement stc=con.prepareStatement(sqlc);

ResultSet rsc=stc.executeQuery();

while(rsc.next()) {

Classes c = new Classes();

c.setId(rsc.getInt(1));

c.setSection(rsc.getInt(2));

c.setTeacher(rsc.getString(3));

c.setSubject(rsc.getString(4));

c.setTime(rsc.getString(5));

listC.add(c);

}

}

else if (query == 2){

String sqlc="select \* from Classes where subject='HINDI'";

PreparedStatement stc=con.prepareStatement(sqlc);

ResultSet rsc=stc.executeQuery();

while(rsc.next()) {

Classes c = new Classes();

c.setId(rsc.getInt(1));

c.setSection(rsc.getInt(2));

c.setTeacher(rsc.getString(3));

c.setSubject(rsc.getString(4));

c.setTime(rsc.getString(5));

listC.add(c);

}

}

else {

String sqlc="select \* from Classes";

PreparedStatement stc=con.prepareStatement(sqlc);

ResultSet rsc=stc.executeQuery();

while(rsc.next()) {

Classes c = new Classes();

c.setId(rsc.getInt(1));

c.setSection(rsc.getInt(2));

c.setTeacher(rsc.getString(3));

c.setSubject(rsc.getString(4));

c.setTime(rsc.getString(5));

listC.add(c);

}

}

//address of the table if it exists

return listC;

}

}

**In package DbUtil**

**2.3** package com.simplilearn.dbutil;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DbUtil {

public static final String DRIVER\_CLASS = "com.mysql.jdbc.Driver";

public static final String DB\_URL="jdbc:mysql://localhost:3306/db9";

public static final String USERNAME="root";

public static final String PASSWORD ="1526";

public static Connection dbConn() throws SQLException, ClassNotFoundException {

//register the driver

Class.forName(DRIVER\_CLASS);

//connection statement

Connection con=DriverManager.getConnection(DB\_URL,USERNAME,PASSWORD);

return con;

}

}

**In package pojo**

2.4 package com.simplilearn.pojo;

public class Classes {

private int id;

private int section;

private String teacher;

private String subject;

private String time;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getSection() {

return section;

}

public void setSection(int section) {

this.section = section;

}

public String getTeacher() {

return teacher;

}

public void setTeacher(String teacher) {

this.teacher = teacher;

}

public String getSubject() {

return subject;

}

public void setSubject(String subject) {

this.subject = subject;

}

public String getTime() {

return time;

}

public void setTime(String time) {

this.time = time;

}

}

2.5 package com.simplilearn.pojo;

public class Student {

private int id;

private String sname;

private int sage;

private int sclass;

public int getid() {

return id;

}

public void setid(int id) {

this.id = id;

}

public String getSname() {

return sname;

}

public void setSname(String sname) {

this.sname = sname;

}

public int getSage() {

return sage;

}

public void setSage(int sage) {

this.sage = sage;

}

public int getSclass() {

return sclass;

}

public void setSclass(int sclass) {

this.sclass = sclass;

}

@Override

public String toString() {

return "Student [id=" + id + ", sname=" + sname + ", sage=" + sage + ", sclass=" + sclass + "]";

}

}

2.6 package com.simplilearn.pojo;

public class Subject {

private int id;

private String name;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

2.7 package com.simplilearn.pojo;

public class Teachers {

private int id;

private String name;

private int age;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

}

2.8 package com.simplilearn.pojo;

public class ReportCard {

private int id;

private String sname;

private int sage;

private int sclass;

private int cid;

private int section;

private String teacher;

private String subject;

private String time;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getSname() {

return sname;

}

public void setSname(String sname) {

this.sname = sname;

}

public int getSage() {

return sage;

}

public void setSage(int sage) {

this.sage = sage;

}

public int getSclass() {

return sclass;

}

public void setSclass(int sclass) {

this.sclass = sclass;

}

public int getCid() {

return cid;

}

public void setCid(int cid) {

this.cid = cid;

}

public int getSection() {

return section;

}

public void setSection(int section) {

this.section = section;

}

public String getTeacher() {

return teacher;

}

public void setTeacher(String teacher) {

this.teacher = teacher;

}

public String getSubject() {

return subject;

}

public void setSubject(String subject) {

this.subject = subject;

}

public String getTime() {

return time;

}

public void setTime(String time) {

this.time = time;

}

}

In webapp creates jsps are

**2.9 index.jsp**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<form action="Login.jsp">

<h3 style="text-align:center;"> <i> - Welcome to Learner's Academy - </i></h3><br>

Admin\_Username<input type="text" name="admin"><br>

Password<input type="password" name="pwd"><br>

<input type="submit" value="login">

</form>

</body>

</html>

**2.10 Login.jsp**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h3 style="text-align:center;"><i> Administrator Login </i></h3>

<%

if(request.getParameter("admin").equals(request.getParameter("pwd"))){

out.print("successfull login");

}

else{

out.print("login failed");

response.sendRedirect("logout.jsp");

}%>

<h3>Go to... </h3>

<form action = StudentsList.jsp>

<input type = "submit" value= "Students List"></form><br>

<form action = TeachersList.jsp>

<input type = "submit" value = " Teacher's List"></form><br>

<form action = "ClassesList.jsp">

<input type = "submit" value=" Classes List"></form><br>

<form action = "SubjectList.jsp">

<input type = "submit" value=" Subject List"></form><br>

<form action = "ClassesSubjectList.jsp">

<input type = "submit" value=" Classes-Subject List"></form><br>

<form action = "ClassReport.jsp">

<input type = "submit" value= "Class Report card"></form><br>

</body>

</html>

**2.11 Logout.jsp**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<%= "check your credentials and login below" %>

<%@include file="index.jsp" %>

</body>

</html>

**2.12 ClassesList.jsp**

<%@ page import ="com.simplilearn.pojo.Classes" %>

<%@ page import ="com.simplilearn.dao.DbRetrieveDAO" %>

<%@page import="java.util.\*" %>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h3 style="text-align:center;"><i> Classes List </i></h3>

<table border="1">

<tr><th>Class id</th><th>Section Name</th><th>Teacher</th><th> Subject</th><th> Time</th></tr>

<%

DbRetrieveDAO dbdao = new DbRetrieveDAO();

List<Classes> clist = dbdao.displayClass(1);

for(Classes c:clist){

%>

<tr><td><%=c.getId()%></td><td><%=c.getSection()%></td><td><%=c.getTeacher()%></td><td><%=c.getSubject()%></td><td><%=c.getTime()%></td></tr>

<% }%>

</table>

</body>

</html>

**2.13 StudensList.jsp**

<%@ page import ="com.simplilearn.pojo.Student" %>

<%@ page import ="com.simplilearn.dao.DbRetrieveDAO" %>

<%@page import="java.util.\*" %>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h3 style="text-align:center;"><i> Students List </i></h3>

<table border="1">

<tr><th>Student id</th><th>Student Name</th><th>Age</th><th> Class</th></tr>

<%

DbRetrieveDAO dbdao = new DbRetrieveDAO();

List<Student> slist = dbdao.display();

for(Student s:slist){

%>

<tr><td><%=s.getid()%></td><td><%=s.getSname()%></td><td><%=s.getSage()%></td><td><%=s.getSclass()%></td></tr>

<% }%>

</table>

</body>

</html>

**2.14 SubjectsList.jsp**

<%@ page import ="com.simplilearn.pojo.Subject" %>

<%@ page import ="com.simplilearn.dao.DbRetrieveDAO" %>

<%@page import="java.util.\*" %>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h3 style="text-align:center;"><i> Subjects List </i></h3>

<table border="1">

<tr><th>Subject id</th><th>Subject Name</th></tr>

<%

DbRetrieveDAO dbdao = new DbRetrieveDAO();

List<Subject> list = dbdao.displaysub();

for(Subject sub:list){

%>

<tr><td><%=sub.getId()%></td><td><%=sub.getName()%></td><tr>

<% }%>

</table>

</body>

</html>

**2.15 TeachersList.jsp**

<%@ page import ="com.simplilearn.pojo.Teachers" %>

<%@ page import ="com.simplilearn.dao.DbRetrieveDAO" %>

<%@page import="java.util.\*" %>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h3 style="text-align:center;"><i> Teacher's List </i></h3>

<table border="1">

<tr><th>Teacher id</th><th>Teacher Name</th><th>Teacher's Age</th></tr>

<%

DbRetrieveDAO dbdao = new DbRetrieveDAO();

List<Teachers> tlist = dbdao.displayTeachers();

for(Teachers t:tlist){

%>

<tr><td><%=t.getId()%></td><td><%=t.getName()%></td><td><%=t.getAge()%></td></tr>

<% }%>

</table>

</body>

</html>

**2.16 Classes-SubjectsList.jsp**

<%@ page import ="com.simplilearn.pojo.Classes" %>

<%@ page import ="com.simplilearn.dao.DbRetrieveDAO" %>

<%@page import="java.util.\*" %>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h3 style="text-align:center;"><i> Classes - Subjects List </i></h3>

<table border="1">

<tr><th>Class id</th><th>Section Name</th><th>Teacher</th><th> Time</th></tr>

<%

DbRetrieveDAO dbdao = new DbRetrieveDAO();

List<Classes> cslist = dbdao.displayClass(2);

for(Classes cs:cslist){

%>

<tr><td><%=cs.getId()%></td><td><%=cs.getSection()%></td><td><%=cs.getTeacher()%></td><td><%=cs.getTime()%></td></tr>

<% }%>

</table>

</body>

</html>

**2.17 ClasseReport.jsp**

<%@ page import ="com.simplilearn.pojo.Student" %>

<%@ page import ="com.simplilearn.pojo.Classes" %>

<%@ page import ="com.simplilearn.dao.DbRetrieveDAO" %>

<%@page import="java.util.\*" %>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h3 style="text-align:center;"><i> Report Card </i></h3>

<table border="1">

<tr><th>Student id</th><th>Student Name</th><th>Student Age</th><th>Student Class</th></tr>

<%

DbRetrieveDAO dbdao = new DbRetrieveDAO();

List<Student> slist = dbdao.display();

List<Classes> clist = dbdao.displayClass(0);%>

<% for(Student s:slist){%>

<tr><td><%=s.getid()%></td><td><%=s.getSname()%></td><td><%=s.getSage()%></td><td><%=s.getSclass()%></td></tr>

<% }%>

<tr><th>Class Id </th><th>Class Section </th><th>Class Teacher </th><th>Class Subject</th><th>Time of Class </th></tr>

<%for(Classes c:clist){%>

<tr><td><%=c.getId()%></td><td><%=c.getSection()%></td><td><%=c.getTeacher()%></td><td><%=c.getSubject()%></td><td><%=c.getTime()%></td></tr>

<% }%>

</table>

</body>

</html>

**2.18 mySQL code in database**

**create database db9;**

**use db9;**

**Create Table Student(**

**id int PRIMARY KEY,**

**sname varchar(255),**

**sage int,**

**sclass int);**

**desc Student;**

**insert into Student values(1,"Santhi",11,6);**

**insert into Student values(2,"Rekha",10,5);**

**insert into Student values(3,"Vasu",9,4);**

**insert into Student values(4,"Harini",12,7);**

**insert into Student values(5,"Pavan",13,8);**

**insert into Student values(6,"Santosh",14,9);**

**insert into Student values(7,"Pinky",15,10);**

**insert into Student values(8,"Chintu",14,10);**

**Alter table Student add Key class\_id (sclass);**

**select \* from Student;**

**desc Student;**

**create Table Subjects(**

**id int primary key,**

**subname varchar(255));**

**desc Subjects;**

**insert into Subjects values(1,"MATH");**

**insert into Subjects values(2,"ENG");**

**insert into Subjects values(3,"SCI");**

**insert into Subjects values(4,"HINDI");**

**create table Teachers(**

**id int primary key,**

**name varchar(255),**

**age int);**

**insert into Teachers values(1, "Ramakrishna", 35);**

**insert into Teachers values(2, "Haritha", 30);**

**insert into Teachers values(3, "Sridevi", 28);**

**insert into Teachers values(4, "Sravya", 30);**

**insert into Teachers values(5, "John", 35);**

**insert into Teachers values(6, "AbdulKhan", 38);**

**insert into Teachers values(7, "Mary", 30);**

**insert into Teachers values(8, "Saritha", 32);**

**select \* from Teachers;**

**create table Classes(**

**id int primary key,**

**section int,**

**teacher varchar(255) ,**

**subject varchar(255),**

**time varchar(255),**

**foreign key (section) references Student(sclass));**

**alter table Classes add constraint fk\_section foreign key(section) references Student(sclass);**

**select \* from classes;**

**#foreign key (teacher\_id) references Teachers(id),**

**#foreign key (subject\_id) references Subjects(id)**

**desc Classes;**

**insert into Classes values(1,6,"Ramakrishna","MATH",'9:00');**

**insert into Classes values(2,5,"Haritha","SCI",'10:00');**

**insert into Classes values(3,4,"Sridevi","HINDI",'11:00');**

**insert into Classes values(4,7,"Sravya","ENG",'9:00');**

**insert into Classes values(5,8,"John","ENG",'10:00');**

**insert into Classes values(6,9,"Ramakrishna","MATH",'9:00');**

**insert into Classes values(7,10,"Mary","SCI",'11:00');**

**insert into Classes values(8,10,"Saritha","HINDI",'10:00');**

**select \* from Classes;**

**#Select \* from Student right outer join Classes on Student.sclass=Classes.section;**

## **Pushing the code to GitHub repository**

* Open your command prompt and navigate to the folder where you have created your files.

**cd <** **D:\LearnersAcademyPhase2>**

* Initialize repository using the following command:

**git init**

* Remote add the repository using the following command

**https://github.com/Mogilisanthi/SimpliLearn-Phase-2-Project.git**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “learners academy project”**

* Push the files to the folder you initially created using the following command:

**git push -u origin main**