Backend Admin for Learner’s Academy: Writeup

(Phase 2 Project)

Name : K V Sagar

Project and Developer Details:

- The **Learner’s Academy** is a school and this project is an online management system.
- The user can manage the master list of the following:
  1. **Teachers**
  2. **Classes**
  3. **Students**
  4. **Subjects**
- In each of the master lists the user is allowed to perform the following operations:
  1. **Add**
  2. **Modify**
  3. **Delete**
  4. **View**
- The user also has options to assign relationships i.e. **assign a teachers to classes**, **assign classes for subjects** and **assign students to classes**.
- Each of the pages also have and option to go back to the home page.
- As per requirement the user can also view class reports for each class individually for the following relationships:
  1. **Class – Subject**
  2. **Class – Student**
  3. **Class – Teacher**
- A **cumulative report** can also be viewed by the user which shows all the relationship between all the attributes in one table.
- The developer’s name is **K V Sagar**.

Sprints Planned and Tasks Achieved

- The total duration of this is 16 days i.e. ~ 3.5 weeks.
- The has been divided into 4 Sprints each of 4 days.
- The goals of each of the sprints are mentioned below:

1. **Sprint 1:**

   ➢ A backlog that is well defined is first created with details of all the tasks to ensure on time delivery.

   ➢ The first sprint is dedicated to the design development of the application.

   ➢ The developer considers all the requirements of the client and decides the approach to be taken.

   ➢ In this sprint the multiple entities and their relationships are discovered to model the project close to the real world.

   ➢ Tasks Achieved: At the end of Sprint 1, the design was finalized.

2. **Sprint 2:**

   ➢ First the classes are defined for entities with their attributes to set up the master list.

   ➢ The following are the different classes that have been defined:

   a. MasterTeachers
   b. MasterClasses
   c. MasterSubjects
   d. MasterStudents

   ➢ The relationships between the entities, the primary key and the foreign key all have established in this sprint.

   ➢ The annotations have been used to connect the java classes to the database.

   ➢ Tasks Achieved : At the end of Sprint 2 the entities and their relationships have been developed.

3. **Sprint 3:**

   ➢ The Sprint 3 is dedicated for the development of the home page and the management pages of the master lists.

   ➢ First the **home page** is designed which is a simple html file.

   ➢ The home page offers options to **manage the master list** of Teachers, Classes, Subjects and Teachers.

   ➢ The home page also allows the user to view class report.

   ➢ Then each of the master list management pages are developed using html where the user can perform the **CRUD** operations on the database.

   ➢ Assignment of teachers to classes, classes for subjects and students to classes can also be done.

   ➢ Tasks Achieved: At the end of Sprint 3 web pages are developed.

4. **Sprint 4:**

   ➢ In the Sprint 3 the web pages were developed which redirect the requests to Servlets which is where the operation is performed.

   ➢ Separate Servlets have been developed for each of the CRUD operations for the different master lists.

   ➢ The Servlets for the assigning operation is also developed in the Sprint 4.

   ➢ All logical loopholes are taken care of as well as the exception handling and fixing bugs is done in this Sprint.

   ➢ Tasks Achieved: At the end of Sprint 4 Servlets that perform the actual operations are developed and the exception handling and bugs are fixed.

Flow of the Application

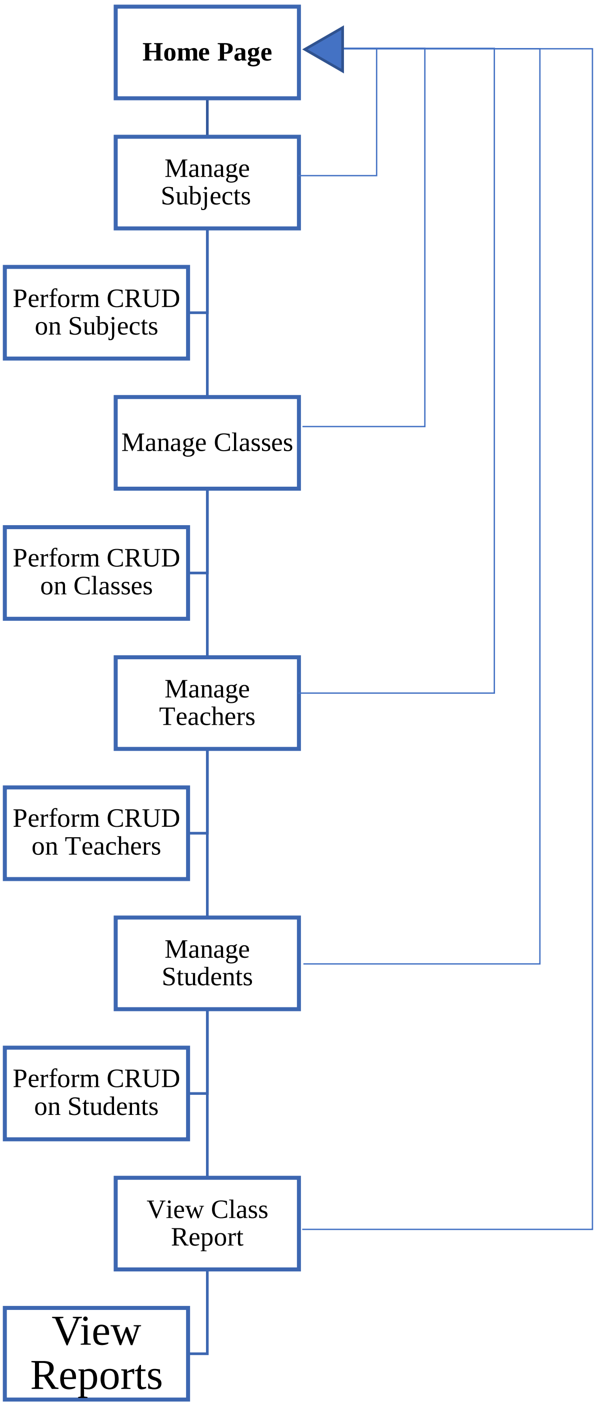
- The home page displays the following options to the user along with a welcome message:

1. **Manage Subjects**
2. **Manage Classes**
3. **Manage Teachers**
4. **Manage Students**
5. **View Class Report**

- Clicking on **Manage Subjects** will take the user to a page where the can perform the following operations on the master list of Subjects:
  1. **Add Subject**
  2. **Modify Subject**
  3. **Delete Subject**
  4. **View Subjects**
  5. **Assign Classes for Subjects**
  6. **Go back to homepage**
- Clicking on **Manage Classes** will take the user to a page where the can perform the following operations on the master list of Subjects:
  1. **Add Class**
  2. **Modify Class**
  3. **Delete Class**
  4. **View Classes**
  5. **Assign Teachers to a Class**
  6. **Go back to homepage**
- Clicking on **Manage Teachers** will take the user to a page where the can perform the following operations on the master list of Subjects:
  1. **Add Teacher**
  2. **Modify Teacher**
  3. **Delete Teacher**
  4. **View Teachers**
  5. **Go back to homepage**
- Clicking on **Manage Students** will take the user to a page where the can perform the following operations on the master list of Subjects:
  1. **Add Student**
  2. **Modify Student**
  3. **Delete Student**
  4. **View Students**
  5. **Assign Students to a Class**
  6. **Go back to homepage**
- Clicking on **View Class Report** will take the user to a page where the user can view the following reports:
  1. **Class 1**

2. **Class 2**
3. **Class 3**
4. **Cumulative Report**
5. **Go back to homepage**

Flow Chart



Core Concepts

- A html page is developed for the home page.
- **href attributes** are used to specify the URL links for the various pages that manage the master list of subjects, classes, teachers, students and for the page to view the class reports.
- Each of the master list management pages also contain **href attributes** to redirect the request to the corresponding Servlet.
- Theses html pages also use herf attributes to allow the user come back to the home page.
- **Hibernate** is used here which is Java framework that simplifies the interaction with databases.
- It is an open source and light weight ORM tool (Object-Relation Mapping).
- The Hibernate's **SessionFactory** is responsible for the creation of session objects.
- It provides methods such as **save**, **delete** and **update** which are used to perform the CRUD operations.
- Servlets are used to perform the CRUD operations on each of the master lists.
- They are used to create Dynamic Web Applications in Java that run on servers.
- It behaves as a class that extends from the HtppServlet.
- The Hibernate Configuration File (**hibernate.cfg.xml**) is an xml file which is loaded into the hibernate application to establish a connection with the database server.
- The various Java classes are created for the master lists and annotations are used to connect them to the database.
- The class **MasterSubjects.java** contains the attributes subjectID ,subjectName and classID from the class MasterClasses.java.
- Here, the subjectID is the primary key and classID is the foreign key as MasterSubjects.java has a **Many-to-One** relationship with MasterClasses.java.
- **MasterClasses.java** has the attributes classID, className and teacherID from the class MasterTeachers.java.
- The attribute classID is the primary key and the tecaherID is the foreign key. It has **Many-to-One** relationship with the class MasterTeachers.java.
- The class **MasterTeachers.java** has the attributes teacherID and teacherName.

- The attribute teacherID is the primary key.
- The class **MasterStudents.java** the attributes studentID, studentName and classID as it has a Many-to-One relationship with **MasterClasses.java**.
- studentID is the primary key and the classID is the foreign key.
- Each time the user performs either the add or the modify operation on any of the list the entry that is made is shown in a new page.
- This is done by using **ArrayList** that is made to store the objects of the Master List classes as its elements.
- After adding the object to the ArrayList the request is forwarded to a JSP form using the **RequestDispatcher**.
- The JSP form uses the **taglib** to use the forEach to print the elements of the ArrayList.

Conclusion

- This web application considers all the requirements specified.
- The project is divided into 4 Sprints to be able to work effectively and deliver on time.
- Java Hibernate, JSP, JDBC, JSTL, HTML, etc. are the different technologies used among others to simplify the Java-Database connectivity.
- CRUD operations are performed on the database which has entities that relate to the real world.
- The entities are created by using the annotations on the master list java classes for Subjects, Classes, Teachers and Students.
- Class Reports are created according to the specification.

Git Link: simpliLearn/Phase-2/Phase-2-Project-LearnersAcademy at main · K-V-Sagar/simpliLearn (github.com)