**SPECIFICATION DOCUMENT**

**Introduction**

Learner’s Academy is a school that has an online management system. The system keeps track of its classes, subjects, students, and teachers. It has a back-office application with a single administrator login. The administrator can:

* Set up a master list of all the subjects for all the classes
* Set up a master list of all the teachers
* Set up a master list of all the classes
* Assign classes for subjects from the master list
* Assign teachers to a class for a subject (A teacher can be assigned to different classes for different subjects)
* Get a master list of students (Each student must be assigned to a single class)

There will be an option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teachers

The goal of the company is to deliver a high-end quality product as early as possible.

My name is Sonu Ahluwalia and I am the developer who will be developing this prototype of Learners Academy.

**PLANNING DOCUMENT**

Plan more than two sprints to complete the application.

Document the step-by-step process starting from sprint planning to the product release.

**PRODUCT BACKLOG**

* **USER STORIES**
  1. The login screen.
  2. The home page.
  3. Master List of Subjects screen.
  4. Mater List of Teachers screen.
  5. Master List of Classes screen.
  6. Master List of Students screen.

**SPRINT 1**

**SPRINT BACKLOG**

To add the following functionality to the application:

1. The login screen.
2. The home page.
3. Master List of Subjects screen.
4. Mater List of Teachers screen.

**SPRINT PLANNING MEETING**

* USER STORIES INTO TECHNICAL TASKS
  1. The login screen.
     + The admin will use the username and password to login into the application which would be verified from the database.
     + After the login screen, the admin would be displayed the home page.
  2. The home page
     + Here four options would be displayed to the admin, to show the four master lists of subjects, teachers, classes, and students.
  3. Master List of Subjects screen.
     + Here all the subjects details would be retrieved from the database and displayed to the admin.
     + Add subject option – Option will be given to the admin to add a subject. When adding a subject, class would also be assigned to it. Same subject name and same class name would not be allowed.
     + Delete subject option – Option will be given to the admin to delete a subject.
     + Update subject option – Option will be given to the admin to update any subject details.
  4. Master List of Teachers screen.
     + Here all the teachers details would be retrieved from the database and displayed to the admin.
     + Add teacher option – Option will be given to the admin to add a teacher. When adding a teacher, class and subject would also be assigned to it. Same teacher name, class name and subject name would not be allowed.
     + Delete teacher option – Option will be given to the admin to delete a teacher.
     + Update teacher option – Option will be given to the admin to update any teacher details.

* DEFINITION OF DONE

All the features in the sprint backlog would be considered done when all the relevant test cases have been passed.

* CAPACITY FOR THE SPRINT

Sprint capacity = 8hrs per day X 5 days X 1 = 40 hours

**SPRINT 2**

**SPRINT BACKLOG**

To add the following functionality to the application:

1. Master List of Classes screen.
2. Master List of Students screen.

**SPRINT PLANNING MEETING**

* USER STORIES INTO TECHNICAL TASKS
  1. Master List of Classes screen.
     + Here all the classes details would be retrieved from the database and displayed to the admin.
     + Add class option – Option will be given to the admin to add a class.
     + Delete class option – Option will be given to the admin to delete a class.
     + Update class option – Option will be given to the admin to update any class details.
  2. Master List of Students screen.
     + Here all the students details would be retrieved from the database and displayed to the admin.
     + Add student option – Option will be given to the admin to add a student. Student will be assigned a class while adding.
     + Delete student option – Option will be given to the admin to delete a student.
     + Update student option – Option will be given to the admin to update any student details.
* DEFINITION OF DONE

All the features in the sprint backlog would be considered done when all the relevant test cases have been passed.

* CAPACITY FOR THE SPRINT

Sprint capacity = 8hrs per day X 5 days X 1 = 40 hours

**Step to run the application:**

1. Download the code from github repository.

<https://github.com/sonuahluwalia/learnersacademy.git>

1. In DBDocs folder, learner.sql is the main database script which acts as a backend to the application. Run the script in the oracle database 11g or higher. This will setup all the tables, sequences, procedures and initial data of the application.
2. In src -> com -> learner -> util ->DBConnection.java class, you can change the username, password, dbServiceName or dbDatabaseName for your own jdbc connection.
3. Run the application as a maven project in apache tomcat server. The application has been tested with apache tomcat server 9.
4. Login page would be displayed and the username is admin, password is admin.
5. Home page is displayed with options to choose from master list of subjects, teachers, classes, students.
6. Each master list has options to add, delete and update the respective details.

Existing business rules:

1. A teacher should be assigned to class and a subject
2. Subject should be assigned to a class.
3. Student should be assigned to a class.

I have added some business rules of my own to make it more interesting:

1. A subject cannot be added without a class.
2. A teacher cannot be added without a class and a subject.
3. Class names should be in the pattern of Class 1 to Class 12 as it doesn’t make sense to add haywire names.
4. Same subject name and class name cannot be added as the admin does not keep hitting the add button and the same subject name and class name keeps getting added in the subjects master list.
5. Same teacher name, class name and subject name cannot be added as the admin does not keep hitting the add button and the same teacher name and same subject name and class name keeps getting added in the teachers master list.
6. Same student name, class name could not be added as the admin does not keep hitting the add button and the same student name and class name keeps getting added in the students master list.
7. While adding subject, class, teacher or student fresh values are generated from the database as there could be more than one admin using the system.

**Algorithm:**

Step 1: Admin logins into the application using admin as username and admin as password.

Step 2: Home page is displayed with four options of Master List of subjects, teachers, classes and students.

Step 3: If the Master list of subjects option is chosen, then

Step 3a: All the subjects details in the database will be displayed.

Step 3b: Option to add the subject.

Step 3c: Option to delete the subject.

Step 3d: Option to update the subject.

Step 4: If the Master list of teachers option is chosen, then

Step 4a: All the teachers details in the database will be displayed.

Step 4b: Option to add the teacher.

Step 4c: Option to delete the teacher.

Step 4d: Option to update the teacher.

Step 5: If the Master list of classes option is chosen, then

Step 5a: All the classes details in the database will be displayed.

Step 5b: Option to add the class.

Step 5c: Option to delete the class.

Step 5d: Option to update the class.

Step 6: If the Master list of students option is chosen, then

Step 6a: All the students details in the database will be displayed.

Step 6b: Option to add the student.

Step 6c: Option to delete the student.

Step 6d: Option to update the student.

**Flowchart:**

A close up of text on a white background

Description automatically generated

**Database Diagram**

**A screenshot of a cell phone

Description automatically generated**

**CORE CONCEPTS USED IN THIS PROJECT:**

Technologies used in the project are JSP, EL, JSTL, Servlets, JDBC, Oracle 11g database.

MVC design pattern is used for the layout of the application.

Presentation layer, Controller, Business Layer, DAO layer and Database.

**Github Repositry Link:**

<https://github.com/sonuahluwalia/learnersacademy.git>

[git@github.com:sonuahluwalia/learnersacademy.git](mailto:git@github.com:sonuahluwalia/learnersacademy.git)

**Conclusion:**

In conclusion, I have added some business rules of my own:

1. A subject cannot be added without a class.
2. A teacher cannot be added without a class and a subject.
3. Class names should be in the pattern of Class 1 to Class 12 as it doesn’t make sense to add haywire names.
4. Same class name could be added as the admin does not keep on hitting the add button the same class name keeps getting added in the class master list.
5. Same subject name and class name cannot be added as the admin does not keep hitting the add button and the same subject name and class name keeps getting added in the subjects master list.
6. Same teacher name, class name and subject name cannot be added as the admin does not keep hitting the add button and the same teacher name and same subject name and class name keeps getting added in the teachers master list.
7. Same student name, class name could not be added as the admin does not keep hitting the add button and the same student name and class name keeps getting added in the students master list.

Existing business rules:

1. A teacher should be assigned to class and a subject
2. Subject should be assigned to a class.
3. Student should be assigned to a class.

Further presentation layer could be switched to using Javascript frameworks, bootstrap etc.

The DAO layer could also be written in Hibernate to make it database independent.

Additional business rules which could have been implemented:

1. Same class name should not be added as the admin does not keep on hitting the add button the same class name keeps getting added in the class master list.