**AIM :** To draw the class diagram for School Management System

**THEORY :** A class diagram is a type of static structure diagram that describes the structure

of a system by showing the system classes, their attributes, and the relationships between the classes.

Class diagrams show the classes of the system, their inter-relationships, and the operations

and attributes of the classes. Class diagrams are typically used, although not all at once, to:

* Explore domain concepts in the form of a domain model
* Analyze requirements in the form of a conceptual/analysis model
* Depict the detailed design of object-oriented or object-based software

A class model is comprised of one or more class diagrams and the supporting specifications

that describe model elements including classes, relationships between classes, and interfaces.

There are guidelines

1. General issues
2. Classes
3. Interfaces
4. Relationships
5. Inheritance
6. Aggregation and Composition

**GENERAL GUIDELINES**

Because class diagrams are used for a variety of purposes – from understanding requirements

to describing your detailed design – it is needed to apply a different style in each

circumstance. This section describes style guidelines pertaining to different types of class

Diagrams.

**CLASSES:** A class in the software system is represented by a box with the name of the class

written inside it. A compartment below the class name can show the class&#39;s attributes (i.e. its properties). Each attribute is shown with at least its name, and optionally with its type, initial

value, and other properties. A class is effectively a template from which objects are created

(instantiated). Classes define attributes, information that is pertinent to their instances, and

operations, functionality that the objects support. Classes will also realize interfaces (more on

this later). Class diagrams are widely used to describe the types of objects in a system and

their relationships. Class diagrams model class structure and contents using design elements

such as classes, packages and objects. Class diagrams describe three different perspectives

when designing a system, conceptual, specification, and implementation. These perspectives

become evident as the diagram is created and help solidify the design.

**INTERFACES:** An interface is a collection of operation signature and/or attribute definitions

that ideally defines a cohesive set of behaviors. Interface a class or component must

implement the operations and attributes defined by the interface. Any given class or

component may implement zero or more interfaces and one or more classes or components

can implement the same interface.

**RELATIONSHIPS:** A relationship is a general term covering the specific types of logical

connections found on a class and object diagram. Class diagrams also display relationships

such as containment, inheritance, associations and others. The association relationship is the

most common relationship in a class diagram. The association shows the relationship

between instances of classes. Another common relationship in class diagrams is a

generalization. A generalization is used when two classes are similar, but have some

Differences.

**AGGREGATION:** Aggregation is a variant of the &quot;has a&quot; or association relationship; composition is more specific than aggregation. Aggregation occurs when a class is a collection or container of other classes, but where the contained classes do not have a strong

life cycle dependency on the container--essentially, if the container is destroyed, its contents

are not.

**ASSOCIATION:** Association are semantic connection between classes. When an association

connects two classes , each class can send messages to other in a sequence or a collaboration

diagram . Associations can be bidirectional or unidirectional.

**DEPENDENCIES:** Dependencies connect two clases . Dependencies are always

unidirectional and show that one class, depends on the definitions in another class .

**GENERALIZATION:** The generalization relationship indicates that one of the two related

classes (the supertype) is considered to be a more general form of the other (the subtype). In

practice, this means that any instance of the subtype is also an instance of the supertype . The

generalization relationship is also known as the inheritance or &quot;is a&quot; relationship. The

supertype in the generalization relationship is also known as the &quot;parent&quot;, superclass, base

class, or base type. The subtype in the generalization relationship is also known as the

&quot;child&quot;, subclass, derived class, derived type, inheriting class, or inheriting type.

**MULTIPLICITY:** The association relationship indicates that (at least) one of the two related

classes makes reference to the other.

1. **Teacher**
   1. **Introduction:** The class Teacher represents all the attributes or information of
      1. the teacher that must be known by the school and all the operations performed by the teacher. It has its associativity with Admin, Students and TestPaper.
   2. **Attributes**
      1. Name: It contains the name of the teacher.
      2. Id: It contains the unique id that has been given to the teacher.
      3. PhNo.: It contains the Phone no of the teacher.
      4. Address: It contains the latest residential address of the teacher.
   3. **Operations**
      1. Prepare\_Test\_Papers(): The teacher prepars the test papers to be given to the students for assessments.
      2. Mark\_Attendence(): Teachers marks the attendance of the students present in the class daily
      3. Check\_Papers(): Teachers checks the papers that were given by the students.
      4. Prepare\_Report\_Cards(): teacher has to compile the result of all the test for each student and save them
      5. Declare\_Result(): teachers need to declare the result of all the students.
2. **Student**
   1. **Introduction:** The class student represents the children who comes to the school for the studying. It is associated to teacher, testpaper and class
   2. **Attributes**
      1. Name: It contains the name of the student.
      2. PhNo: It contains the phone number of the student.
      3. Address: It contains the address of the student
      4. Id: It contains the id of the student, a unique number given to the student.
   3. **Operations**
      1. FIll\_Admission\_Form(): Student need to fill this before getting admitted to the school
      2. Get\_enrolled(): Student need to get themselves enrolled to the school before attending the shcool
      3. Write\_exam(): Student need to write there exams to get promoted to a new class.
3. **Test Paper**
   1. **Introduction:** Students need to attempt the papers made by the teacher so as to promote to the next division in school.
   2. **Attributes**
      1. ID: It contains Unique id of the testpaper
      2. Name: It contains Name of the exam
      3. Duration: it contains duration of the exam
      4. Classid: It contains the unique class id for which the exam is to be conducted
4. **ADMIN**
   1. **Introduction:** The Admin class is handled at the reception of the School , to

manage all the activities of the school - to provide Add and modify teachers information ; for login and logout ; to add and modify Students information; to Add and modify class information; to add and modify division information

* 1. **Attributes**
     1. ID: It contains the Unique Id No. of the Admin
     2. Name:It contains Name of the admin
     3. Password: It contains the unique password of a particular admin
  2. **Operations**
     1. Login(): this helps admin to log in to the portal
     2. Logout(): this helps admin to log out of the portal
     3. Add\_New\_Teachers(): admin can add new teacher information using this
     4. Modify\_Teachers(): admin can modify teacher information using this.
     5. Add\_New\_Student(): admin can add new student information using this
     6. Modify\_Student(): admin can modify student information using this
     7. Add\_New\_Subject(): admin can add new subject using this
     8. Modify\_Subject(): admin can modify current subjects using this
     9. Add\_New\_Class(): admin can add new class using this
     10. Modify\_Class(): admin can modify already formed class using this
     11. Add\_New\_Division(): admin can add new division to the school using this
     12. Modify\_Division(): admin can modify current divisions of the school using this

1. **Subject**
   1. **Introduction:** These are the current subjects that are taught to a particular class. These can be changed or modified only by the admin.
   2. **Attributes:**
      1. ID: This contains the unique subject ID
      2. Name: This contains the name of the subject
      3. ClassID: This contains the ID of the class the subject is taught to
   3. **Operations**
      1. Display\_Subjects(): used to display the information about all the subjects
2. **Division**
   1. **Introduction:** These are the different division present in the school. These can only be changed by the Admin.
   2. **Attributes:**
      1. ID: it contains the unique division id for each division
      2. Name: It contains the name of the division
   3. **Operations:**
      1. Display\_Divisions(): This helps in displaying all the information about all the divisions.
3. **Class** 
   1. **Introduction:** There are various classes in a school. These are contained in this block and can be modified only by the admin. Each class contains Students that attend their classes and teachers who teaches these students.
   2. **Attributes:**
      1. ID: contains unique id of each class.
      2. Name: contains the name of the class or else known as section.
      3. Division: contains the information about which division the class belongs to.
   3. **Operations:**
      1. Display\_Class(): It displays all the attributes of class block.

