ASSIGNMENT 4

Title: Design and implementation of Analysis Model-Class Model.

Problem Statement:

- Prepare Analysis Model-Class Model
- Identify Analysis Classes and assign responsibilities
- Prepare Data Dictionary
- Draw Analysis class Model using UML2.0 Notations
- Implement Analysis class Model-class diagram with a suitable object oriented language.

Objective:

- To Identify Analysis Classes and assign responsibilities
- To Draw Analysis class Model
- To Implement Analysis class Model-class diagram

Theory:

Class Diagram

The Class diagram shows the building blocks of any object-orientated system. Class diagrams depict the static view of the model or part of the model, describing what attributes and behaviors it has rather than detailing the methods for achieving operations. Class diagrams are most useful to illustrate relationships between classes and interfaces. Generalizations, aggregations, and associations are all valuable in reflecting inheritance, composition or usage, and connections, respectively.

Classes

A class is an element that defines the attributes and behaviors that an object is able to generate. The behavior is described by the possible messages the class is able to understand along with operations that are appropriate for each message. Classes may also contain definitions of constraints, tagged values and stereotypes.

Class Notation

Classes are represented by rectangles which show the name of the class and optionally the name of the operations and attributes. Compartments are used to divide the class name, attributes and operations. Additionally constraints, initial values and parameters may be assigned to classes.Classes are composed of three things: a name, attributes, and operations.

Classes

- Event
- Registration
- User
- Team
- Participant
- Admin
- Manager
- Volunteer

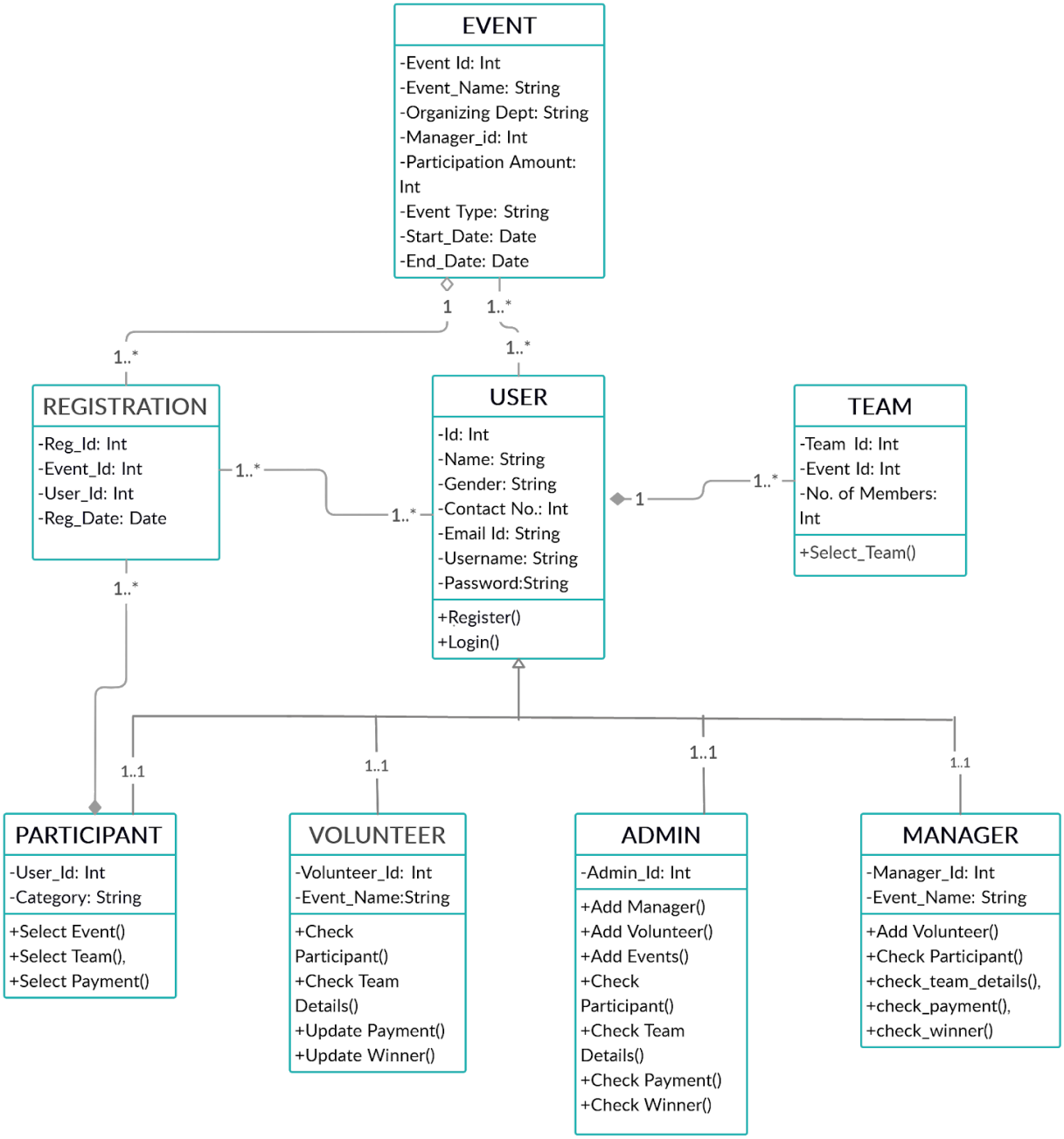
Attributes

- **Event:** Event_Id, Event_Name, Organising_Dept, Manager_Id, Participation_Amount, Event_Type, Start_Date, End_Date
- **Registration:** Reg_Id, Event_Id, User_Id, Reg_Date
- **User:** User_Id, Name, Gender, Contact_No., Email_Id, Username, Password
- **Team:** Team_Id, Event_Id, No_of_Members
- **Participant:** User_Id, Category
- **Admin:** Admin_Id
- **Manager:** Manager_Id, Event_Name
- **Volunteer:** Volunteer_Id, Event_Name

Operations

- **User:** Register(), Login()
- **Team:** Select_Team()
- **Participant:** Select_Event(), Select_Team(), Select_Payment()
- **Admin:** Add_Manager(), Add_Volunteer(), Add_Event(), Check_Participant(), Check_Team_Details, Check_Payment(), Check_Winner()
- **Manager:** Add_Volunteer(), Check_Participant(), Check_Team_Details(), Check_Payment(), Check_Winner()
- **Volunteer:** Check_Participation(), Check_Team_Details(), Update_Payment(), Update_Winner()

Analysis Model for Event Registration System



Conclusion:

We have thus implemented accepted standards and procedures to develop a Class Model for the project idea we have picked (Event Registration System). UML is the standard language for specifying, designing, and visualizing the artifacts of software systems. Thus we have understood that Class Diagram provides an overview of how the application is structured before studying the actual code. It certainly reduces the maintenance time.