**EVENT MANAGEMENT SYSTEM Group TE C21**

**2103080 Hriday Kampani**

**2103072 Suyash Kabra**

**2103082 Ansh Kapil**

**Aim: Identify all scenarios and develop use case diagram for the project.**

**Theory:**

# Use Case Diagram:

# Actors

Actors represent anyone or anything that interact with the system. An actor may

* Only input information to a system
* Only retrieve information from a system
* Both input and retrieve information to and from a system

Typically, the actors are found in the problem statement, and also from conversation with the customers and domain experts.

There are three types of actors:

1. users of the system,
2. external application systems, and
3. external devices that can independently interact with the system.

In UML, an actor is represented stickman symbol, as shown below:

**Use cases:**

Use cases eventually map to the menu option. Use cases represent the functionality provided by the system. Each individual functionality provided by a system is captured as a use case.

A use case thus represents a dialog between an actor and the system. A collection of use cases for a system reflects all the defined ways in which a system can be used.

A use case can be defined as a sequence of transactions performed by a system, that yields a measurable result of values for a particular actor.

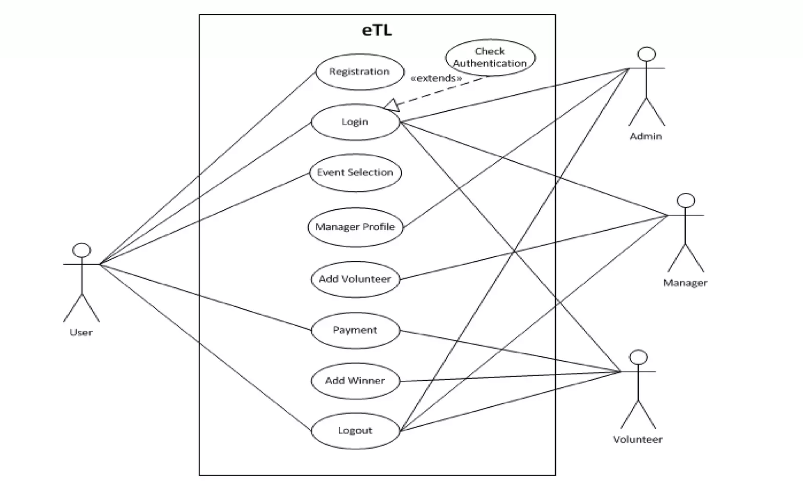
In UML, a use case is represented as an oval, as shown below:

**Use case Diagram:**

A use case diagram is an interaction view of a some or all the actors, use cases and their interactions identified for a system. Each system typically has a main use case diagram, which is a picture of the system boundary (actors) and the major functionality provided by the system (use cases). Other use case diagrams may be created as needed. Some examples are:

* A diagram showing all the use cases for a selected actor.
* A diagram showing all the use cases being implemented in an interaction.
* A diagram showing all the use cases and all its relationship.

**Example:**

****

**Actors:**

1. User
2. Volunteer
3. Manager
4. Admin

**Use cases:**

1. Registeration
2. Login
3. Check Authentication
4. Event Selection
5. Manager Profile
6. Add Volunteer
7. Payment
8. Add Winner
9. Logout

**Description of Use case Diagram:**

* + **User:** User places request for event registration
  + **Volunteer :** Checks payment authentication and creates winners list
* **Manager** : Selects suitable volunteer for event and supervises Volunteer.
* **Admin**:. Authenticates Logins & registerations of manager and user profiles. Selects suitable manager for event , supervises whole process.

**Appendix:**

* 1. **Stereotype:** Stereotypes defines a new model element in terms of another model element. It is represented by <<stereotypes>>
  2. **System boundary boxes:** We can draw a rectangle around the use cases, called the System boundary box, to indicate the scope of your system.
  3. **Abstract use case:** Use case, which is inherited, by some use case is called as abstract use case.
  4. **Concrete use case:** Use case, which is directly inherited by actor, is called as concrete use case.