Assignment 4
Design and Implementation of Analysis Model - Class Model

PROBLEM STATEMENT:

- Prepare Analysis Model-Class Model.
- Identify Analysis Classes and assign responsibilities.
- Prepare a 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.

RELEVANT 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. Below is an example of a class.

Interfaces

An interface is a specification of behavior that implementers agree to meet. It is a contract. By realizing an interface, classes are guaranteed to support a required behavior, which allows the system to treat non-related elements in the same way – i.e. through the common interface.

Associations

An association implies two model elements have a relationship - usually implemented as an instance variable in one class.
This connector may include named roles at each end, multiplicity or cardinality, direction and constraints.
Association is the general relationship type between elements. For more than two elements, When code is generated for class diagrams, associations become instance variables in the target class.

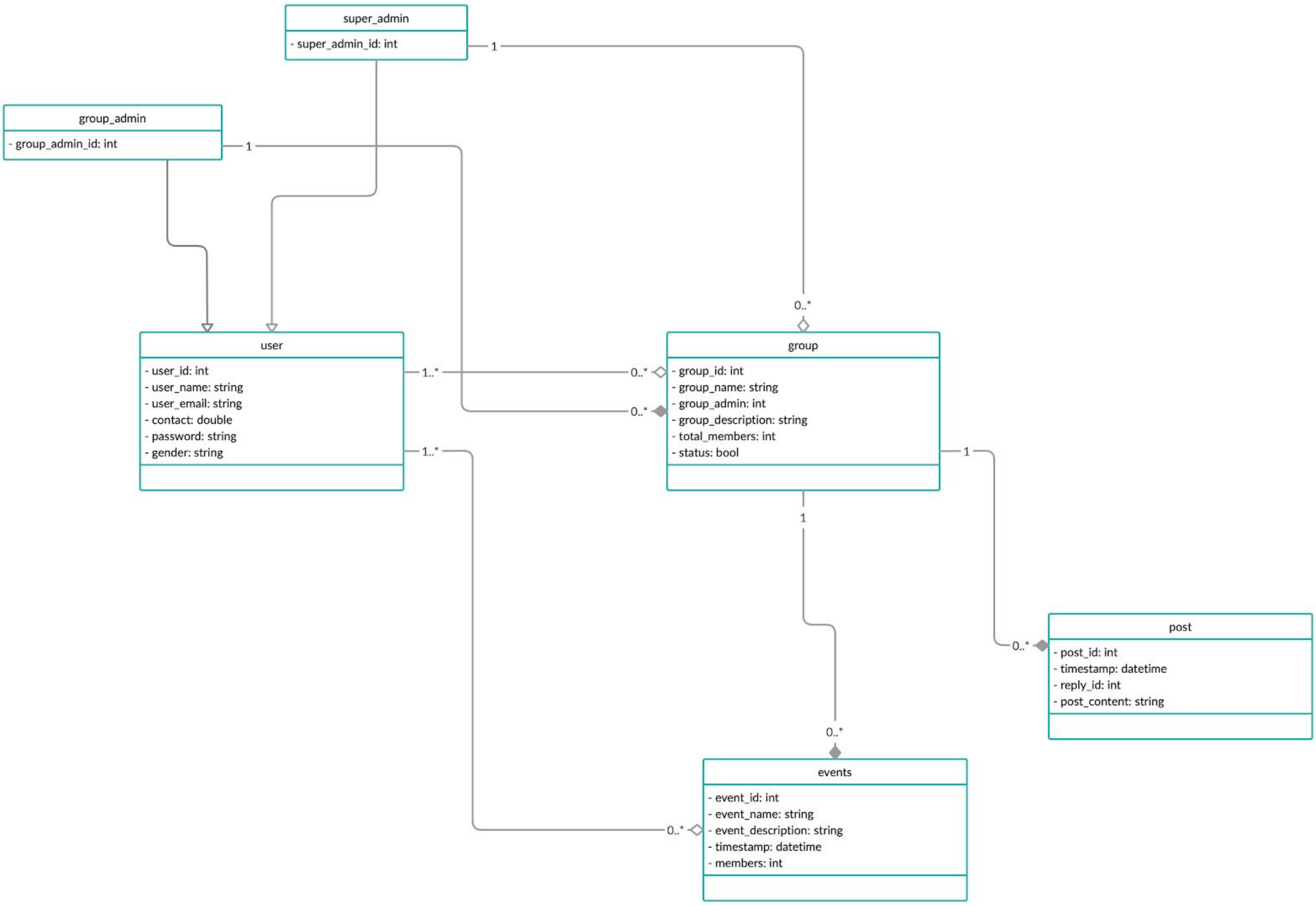
Generalizations

A generalization is used to indicate Inheritance. Drawn from the specific classifier to a general classifier,the generalized implication is that the source inherits the target's characteristics.
The following diagram shows a parent class generalizing a child class. Implicitly, an instantiated object of the Circle class will have attributes x_position, y_position and radius and a method display().

Aggregations

Aggregations are used to depict elements which are made up of smaller components. Aggregation relationships are shown by a white diamond-shaped arrowhead pointing towards the target or parent class. A stronger form of aggregation - a composite aggregation is shown by a black diamond-shaped arrowhead and is used where components can be included in a maximum of one composition at a time. If the parent of a composite aggregation is deleted, usually all of its parts are deleted with it; however a part can be individually removed from a composition without having to delete the entire composition. Compositions are transitive, asymmetric relationships and can be recursive. The following diagram illustrates the difference between weak and strong aggregations. An address book is made up of a multiplicity of contacts and contact groups. A contact group is a virtual grouping of contacts; a contact may be included in more than one contact group. If you delete an address book, all the contacts and contact groups will be deleted too; if you delete a contact group, no contacts will be deleted.

ANALYSIS MODEL DIAGRAM:



CONCLUSION:

Hence we have studied class diagrams and implemented them for our project.