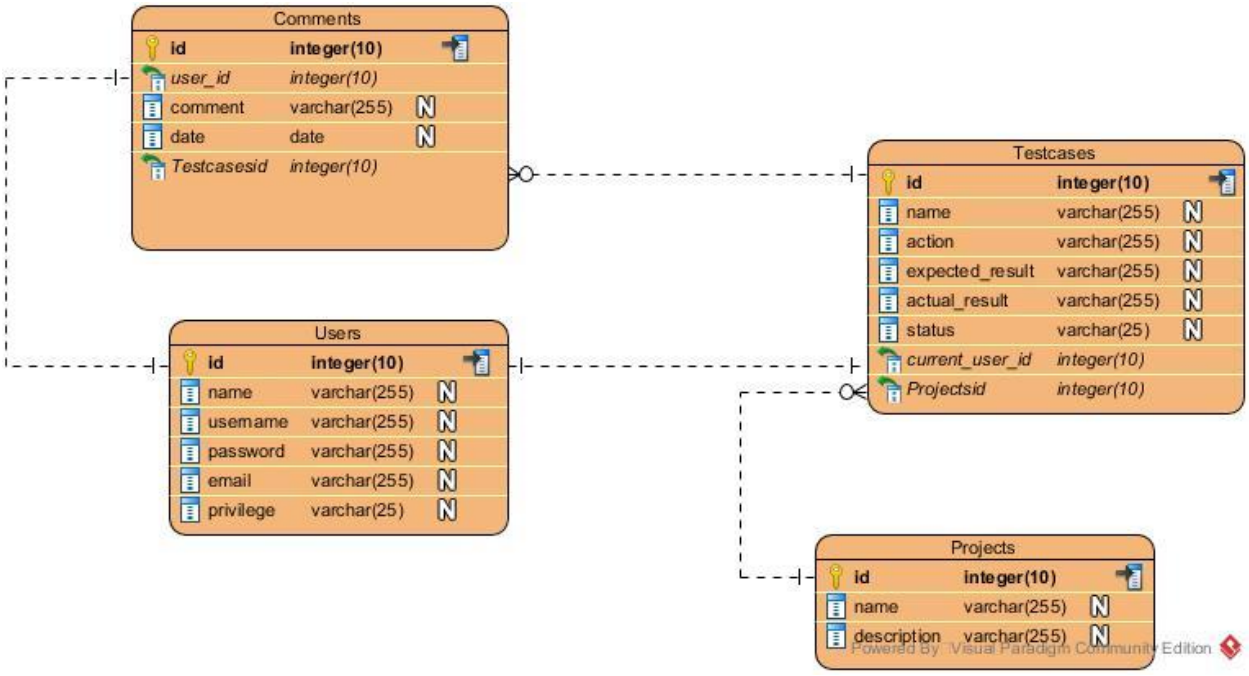
Database Design

Database Type

For the purpose of this project, I decided to use a relational database. I am planning to use PHP language to build the project. Since PHP is most compatible with MySQL therefore, I deiced to choose MySQL as my database management systems (DBMS). My project requires to store data about test cases. Each test case belongs to a project. Therefore, test cases and projects need to be stored in two separate tables to improve normalization. Many comments can be associated with one test case. Therefore, comments also need to be stored in a separate table. Since data need to be stored in various tables I decided to choose a relational database for this project.

Entity Relationship Diagram



Projects Table

In this project, test cases are associated with a project. Therefore, the administrator should be able to add/edit and delete projects. Projects table store name and description of each project.

Users Table

In this project, manager should be able to add, edit and delete users. Therefore, the users table is being used to store the information related to each user. Privilege field store a string value that indicates the privilege of the user. Following options will be available as privilege levels in an array in the source code: “Manager”, “Tester”, “Developer”.

Testcases Table

In this project information related to each testcase is being stored in the testcases table. An ID of the associated project is being stored in the projectsid field. Each project can have many testcases.

At any point in time, each test case is associated with one user in the system. The ID of the associated user is being stored in a current_user_id column in the testcase table. Testcases and users have one to one relationship.

Comments Table

Each testcase can have many comments associated with it. Therefore, the ID of testcase is being stored in testcaseid column.

Each comment is being posted by one user. Therefore, the ID of the author of the comment is being stored in the user_id. Comments and users have one to one relationship.