# Naming Convention for Data

* **User**: external entity using the software being developed
* **Project**: An object in the software that contains several information about the project
* **Owner**: The name (and other attributes) that are linked to a person who is the owner of a particular Project.
* **Project Manager:** The name (and other attributes) that are linked to a person who is the project manager of a particular Project.
* **Team:** A collection of Team Member’s that is linked to a Project.
* **Team Member:** The name (and other attributes) that are linked to a person who is a part of a team linked to a Project.
* **Requirements**: A list of functional and nonfunctional requirements that are linked to a project
* **Functional Requirements:** A requirement that shows what the system needs to do, and the priority of the requirement (1-10, 1 being the highest priority)
* **Nonfunctional Requirement:** A requirement that constraints the system or defines performance standards. The priority of the requirement will be set from 1-10.
* **Task:** A specific task that is assigned to a project. A Team Member is also assigned to all tasks.
* **Task Category:** A task is assigned one of the following categories: Requirements Analysis, Designing, Coding, Testing, Project Management.
* **Category Hours:** The hours spent on a particular category for a particular project
* **Task Hours:** The hours expended on each particular task
* **Project Hours**: The total hours expended on a particular project (sum of all tasks)
* **Risk**: The risk contains a description of a potential risk, probability of likelihood of happening, and probability of mitigation.

# Data Structures

**Person {**

String firstName;

String lastName;

String title;

}

**Requirement {**

String description;

Int priority; (1-10 only)

List <Task> tasks;

}

**Enum TaskCategory** {REQ\_ANALYSIS, DESIGN, CODING, TESTING, PM}

**Task {**

String description;

Person taskOwner;

TaskCategory category;

Float hoursExpended;

}

**Risk {**  
 String description;

Float probabilityLikelihood;

Foat probabilityMitigation;

}

**Project {**

Person projectManager;

Person owner;

List<Requirement> functionalRequirements;

List<Requirement> nonfunctionalRequirements;

List<Person> teamMembers;

List<Risk> projectRisks;

List <Person> projectTeam;

}