

```

<?php
/*
 * models/project.php
 */
class Project extends AppModel {
    var $name = "Project";
    var $customValidate = array("name");

    //The Associations below have been created with all possible keys,
    //those that are not needed can be removed
    var $belongsTo = array(
        "Member" =>
            array("className" => "Member",
                "foreignKey" => "member_id",
                "conditions" => "",
                "fields" => "",
                "order" => "",
                "counterCache" => ""
            ),
    );

    function validateName() {
        if(isset($this->data["Project"]["name"])){
            $name = $this->data["Project"]["name"];
            if(preg_match("/^[a-z][a-z0-9-]*$/i",$name)){
                if($this->findByName($name)){
                    $this->validationErrors["name"] = "1";
                }
            }else{
                $this->validationErrors["name"] = "1";
            }
        }
    }
}
?>

```

```

<?php
/*
 * tests/cases/models/project.test.php
 */
loadModel('Project');

class ProjectTestCase extends UnitTestCase {
    var $object = null;

    function setUp() {
        $this->object = new Project();
    }

    function tearDown() {
        unset($this->object);
    }

    /*
    function testMe() {
        $result = $this->object->doSomething();
        $expected = 1;
        $this->assertEqual($result, $expected);
    }
    */

    function testValidation(){
        //disable Model queries caching
        $this->object->cacheQueries = false;

        $provided["Project"]["name"] = "";
        $this->object->validationErrors = null;
        $this->object->data = $provided;
        $this->assertFalse($this->object->validates());

        $provided["Project"]["name"] = " ";
        $this->object->validationErrors = null;
        $this->object->data = $provided;
        $this->assertFalse($this->object->validates());

        $provided["Project"]["name"] = "a b";
    }
}

```

```

$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertFalse($this->object->validates());

$provided["Project"]["name"] = "a_b";
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertFalse($this->object->validates());

$provided["Project"]["name"] = "a";
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertTrue($this->object->validates());

$provided["Project"]["name"] = "a";
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertTrue($this->object->validates());

$provided["Project"]["name"] = "A";
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertTrue($this->object->validates());

$provided["Project"]["name"] = "0";
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertFalse($this->object->validates());

$provided["Project"]["name"] = "abc01239";
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertTrue($this->object->validates());

$provided["Project"]["name"] = "abc-01239";
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertTrue($this->object->validates());

//test unique project name
$projectName = "test" . time();
$this->__deleteByName($projectName); //ensure initial state

$provided["Project"]["name"] = $projectName;
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertTrue($this->object->validates());

$this->__saveByName($projectName);
$provided["Project"]["name"] = $projectName;
$this->object->validationErrors = null;
$this->object->data = $provided;
$this->assertFalse($this->object->validates());

$this->__deleteByName($projectName);
}

function __deleteByName($projectName) {
    $data = $this->__findByName($projectName);
    if(!empty($data)) {
        $this->object->delete($data["Project"]["id"]);
    }
}

function __saveByName($projectName) {
    $data["Project"]["name"] = $projectName;
    return $this->object->save($data);
}

function __findByName($projectName) {
    return $this->object->find(array("Project.name"=>$projectName));
}
}
?>

```