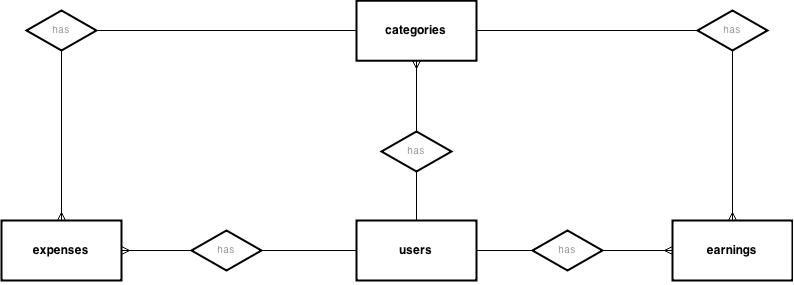
**Documentación de Accounting Book**

**Información general**

* **Lenguaje de programación**: PHP 5.0
* **Framework**: Laravel 4.2
* **Repositorio**: <https://github.com/sanmen1593/AccountingBook>

**Información técnica**

**Diagrama de base de datos**



**Tabla users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Nulleable** | **Unique** |
| Id (PK) | int |  | x |
| name | string |  |  |
| lastname | string |  |  |
| username | string |  |  |
| email | string |  |  |
| currency | string |  |  |

**Tabla earnings**

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Nulleable** | **Unique** |
| Id (PK) | int |  | x |
| amount | string |  |  |
| date | datestring |  |  |
| timestamps | datetime |  |  |
| category\_id (FK) | int |  |  |
| user\_id (FK) | int |  |  |

**Tabla expenses**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Nulleable** | **Unique** | **Description** |
| Id (PK) | int |  | x |  |
| amount | double |  |  |  |
| description | string |  |  |  |
| date | datestring |  |  | AAAAMMDD |
| timestamps | datetime |  |  |  |
| user\_id (FK) | int |  |  |  |
| category\_id (FK) | int |  |  |  |

**Tabla categories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Nulleable** | **Unique** |
| Id (PK) | int |  | x |
| slug | string |  |  |
| superior\_cat | int | x |  |
| type | int |  |  |
| user\_id (FK) | int |  | x |

**Sesiones**

La autenticación a la aplicación se hace mediante la funcionalidad que ofrece Laravel. Para este caso, el usuario se autentica con su usuario (**username**) y contraseña (**password**). Todas las páginas están “prohibidas” al menos que el usuario ya se haya autenticado. En caso tal de que no, simplemente será redirigido a la página de login.

Para manejar las sesiones (login, logout) utilizamos **SessionsController**, que posee 3 funciones:

* **Create**: Esta función se encarga simplemente de retornar la vista **login.blade.php,** que es la vista donde se encuentra el formulario para hacer login. En caso de que el usuario ya esté autenticado, le hace una redirección al dashboard del usuario.
* **Store**: Esta función reciba los parámetros para iniciar sesión (username y password). Después de verificar que ambos datos vengan, realiza un intento de autenticación. Si es correcta, lo reenvía al dashboard de usuario autenticado. Sino, lo reenvía a la página de login con un mensaje de “Login failed!”.
* **Destroy**: Esta función no recibe ningún parámetro. Simplemente cierra la sesión del usuario que se encuentre autenticado, y lo reenvía a la página de login.

**Usuarios**

Se pueden registrar usuarios, actualizar los ya existentes, eliminarse (soft-deleting, no se borran sus datos, pero si el acceso) y por supuesto, consultar su información. Para esto, está la clase UserController, con las siguientes funciones:

* **Index**:
* **Create**:
* **Store**:
* **Update**:
* **Edit**:
* **Destroy**:

**Categorías**