

## Framework:

The project will be developed using php and Laravel 4 (In beta but can be used for production), Laravel 4 is a framework for web artisans, applies the Model View Control (MVC) architecture/design pattern. Laravel owns a very strong database Object Relationship Mapper (ORM) called [Eloquent](#) (ActiveRecord Implementation) which will help us with our complex relations.

Laravel 4 also implements restful controllers and restful routes pattern in core.

## Given Scenarios:

1. At the start of the year Budget owner set limits to regions (NA,EMEA,LATAM,APAC)
2. FAm make a fund request.
3. Once approved budget owner should approve the request and then deduct the amount from the allocated budget.
4. Once the FAm submit the receipts it should be forwarded to the Ledger(assume this is like updating a table for now).
5. At the end of the year FAm can submit their next year budget requirements to the system.
6. Budget owner can see those budget proposals and able to query them for region, amount, types and so on.

## Models:

- **User:** Will handle the user accounts and profiles.
- **Region:** Responsible on each region (NA,EMEA,LATAM,APAC).
- **YearBudget:** Collects year budgets for each region.
- **FundRequest:** Handles the fund request by FAm details.
- **Receipt:** Represents each receipt and related it to it's FundRequest.
- **BudgetProposal:** Represents Budget Proposal by FAm.
- **ProposalItems:** Represent each item in the BudgetProposal.
- **History:** A polymorphic model with will save all the history event, old\_value and new\_value in the application to be retrieved by the historable model. Can be used for undo actions later. See [Laravel Polymorphic Relations](#).

## Database Tables:

- users
  - id
  - email
  - password
  - created\_at
  - updated\_at

- regions
  - id
  - title
  - description
  - budget\_owner\_id (user\_id)
  - created\_at
  - updated\_at
- year\_budgets
  - id
  - region\_id
  - year
  - budget\_limit
  - created\_at
  - updated\_at
- fund\_requests
  - id
  - region\_id
  - FAm\_id
  - status (0 -> requested, 1 -> approved, 2 -> claimed, 3 -> disapproved)
  - value
  - created\_at
  - updated\_at
- receipts
  - id
  - fund\_request\_id
  - value
  - image\_path
  - created\_at
  - updated\_at
- region\_budget\_cache
  - id
  - region\_id
  - remaining\_budget
  - updated\_at
- budget\_proposals
  - id
  - FAm\_id

- region\_id
  - year
  - status (0 -> proposed, 1 -> approved)
  - Other\_Attributes
  - created\_at
  - updated\_at
- proposal\_items
    - id
    - title
    - details
    - proposal\_id
    - value
    - created\_at
    - updated\_at
- history
    - id
    - historable\_id
    - historable\_type
    - event
    - old\_value
    - new\_value
    - created\_at

## Relations:

- **User Model Relations:**
  - Has one **(Region Model)**, Represented by *owner\_id* in *regions* table.
  - Has many **(FundRequest Model)**, Represented by *FAM\_id* in *fund\_requests* table.
  - Has many **(BudgetProposal Model)**, Represented by *FAM\_id* in *budget\_proposal*
- **Region Model Relations:**
  - Has one Owner **(User Model)**, Represented by *owner\_id*.
  - Has many **(YearBudget)**, Represented by the *year\_budget* table.
  - Has many **(BudgetProposal)**, Represented by the *budget\_proposals* table.
- **YearBudget Model Relations:**
  - Belongs to one **(Region Model)**, Represented by *region\_id*.
  - Morph Many **(History Model)**.

- **FundRequest Model Relations:**
  - Belongs to one **(User Model)**, Represented by *FAm\_id*.
  - Has one **(Receipt Model)**, Represented by *fund\_request\_id* in *receipts* table.
  - The above can be Has many relation, So that each receipt could be submitted to one Fund Request, but at the end the total value of receipt should be equal to the value of the approved FundRequest.
  - Morph Many **(History Model)**.
  
- **Receipt Model Relations:**
  - Belongs to **(FundRequest Model)**, Represented by *fund\_request\_id*.
  - Morph Many **(History Model)**.
  
- **BudgetProposal Model Relations:**
  - Has many **(ProposalItems Model)**, Represented by *proposal\_id* in *proposal\_items* table .
  - Belongs to **(User Model)**, Represented by *FAm\_id*.
  - Belongs to **(Region Model)**, Represented by *region\_id*.
  - Morph Many **(History Model)**.
  
- **ProposalItems Model Relations:**
  - Belongs to one **(BudgetProposal Model)**, Represented by *proposal\_id*.
  
- **History**
  - Morph to many Models **(YearBudget, FundRequest, Receipt, ProposalItems)**.

### Assumptions:

- No Fedora Account System Integration (FAS).
- Region is controlled by one user only. [*budget\_owner\_id* is saved in the *regions* table].
- Region budget per year is saved in another table *region\_budgets* to keep year budgets for each region separate.
- Fund request has 4 status: requested, approved, claimed & disapproved.
- Fund request status will change to claimed when FAm submit receipt.
- FAm will not be able to edit the Fund Request details after Budget Owner changed the status to approved, He will be able to delete it (Soft Delete in our case or we can change the status to cancelled for example).
- *region\_budget\_cache* is a cache table for caching the actual amount remaining from budget, To keep the original allocated budget in place and keep the claimed *fund\_request* value in place. While not doing a performance burden to calculate the remaining budget for the ledger page by getting the original allocated budget and deduct the claimed *fund\_requests* from.
- *budget\_proposal* will keep a status field for knowing if it's *proposed* or *approved* (Maybe not needed).

