# Image

## Special Atribs

### Table name = images

## Fields

### id – Int, primary key

### file – string, unique

## Functions

### \_\_init\_\_(self)

# TeamMembers

## Special Atribs

### Table name = team\_members

### (Args: assignment = unique constraint team\_id member\_id)

## Fields

### tea\_id – int, primary key

### member\_id – int, primary key

# User

## Special Atribs

### Table name = users

### Serialize – id, username, name, email, phone, role

### Access – Enum

#### INACTIVE = inactive

#### STANDARD = standard

#### SUPERVISOR = supervisor

#### MANAGER = manager

#### ADMIN = admin

## Fields

### id – Int, primary key

### username – string, not null, unique

### password – string, not null

### name – string, not null

### email – string

### phone – string

### role – enum(Access), not null

### teams – relationship(Team, secondary- team\_members, backreef=member\_list)

## Functions

### \_\_init\_\_(self, username, password, name=None, email=None, phone=None, role=Access.INACTIVE)

# Team

## Special Atribs

### Table name = teams

### serialize – id, name, owner, description, supervisor.name

## Fields

### id – int, primary key

### name – string, not null, unique

### owner – int, foreign key(users.id)

### description – string

### supervisor – relationship, foreign\_keys(owner), backref=supervised\_teams

## Functions

### \_\_init\_\_(self, name, owner, description=’’

# Account

## Special Atribs

### Table name = accounts

### serialize – id, name, description, owner, balance, allocated, owner\_info.id, owner\_info.name)

## Fields

### id – int, primary key

### name – string, not null, unique

### description, string

### owner – int, foreign key(users.id)

### balance – float, not null

### allocated - float, not null

### owner\_info – relationship(User, backref=account\_owner)

### sub\_accounts – relationship(SubAccount, backref=accounts\_sub\_accounts)

### transactions – relationship(allocation, backref=account\_allocations)

## Functions

### \_\_init\_\_(self, name, owner, description=’’, balance=0, allocated=0)

### add\_to\_allocation(self, amount) interface with sub account

### remove\_from\_allocation(self, amount) interface with sub account

### settle\_allocation(self, amount) interface with sub account

# SubAccount

## Special Atribs

### Table name= sub\_accounts

### serialize – id, name, description, balance, allocated, main\_account

## Fields

### id – int, primary key

### name – string, not null, unique

### description – string

### balance – float, not null

### allocated – float, not null

### main\_account – int, foreign key= accounts.id, not null

### transactions – relationship(Allocation, backref=sub\_account\_allocations)

## Functions

### \_\_init\_\_(self, name, main\_account, description=’’, balance=0)

### add\_to\_allocation(self, amount) interface with account

### remove\_from\_allocation(self, amount) interface with account

### settle\_allocation(self, amount) interface with account

### transfer\_in(self, amount)

### release\_funds(self, amount)

# Allocation

## Special Atribs

### Table name = allocations

### serialize – id, account, sub\_account, amount, balance, description, settled, account\_allocations.name, sub\_account\_allocations.name, assigned\_to.name

## Fields

### id – int, primary key

### account – int, foreign key(accounts.id)

### sub\_account – int, foreign key(sub\_accounts.id)

### amount – float, not null

### balance – float, not null

### description – string

### settled – bool, not null

## Functions

### \_\_init\_\_(self, account=None, sub\_account=None, amount=0, balance=0, description, settled=False)

### settle(self, amount, done=False)

# Task

## Special Atribs

### TaskStatus - Enum

#### CREATED – created

#### ASSIGNED – assigned

#### VIEWED – viewed

#### INPROGRESS – in\_progress

#### ONHOLD – on\_hold

#### SUBMITED – submitted

#### REJECTED – rejected

#### VERIFIED – verified

#### COMPLETED – completed

### Table name = tasks

### Serialize – id, name, assigner, description, status, arequires\_verification, team, user, due\_date, notes, creator.name, assigned\_user.name, assigned\_team.name, budget

## Fields

### id – int, primary key

### name – string, not null

### assigner – int, foreign key(users.id), not null

### allocation – int, foreign key(allocations.id)

### due\_date – date, not null

### description – string

### status – enum(TaskStatus), not null

### requires\_verification – bool, not null

### team – int, foreign key(teams.id)

### user – int, foreign key(users.id)

### creator – relationship(TaskNote, backref=listed\_task)

### assigned\_user – relationship(User, foreign\_keys=assigner, backref=created\_tasks)

### assigned\_team – relationship(Team, foreign\_keys=team, backref=tasks)

### budget – relationship(Allocation, backref=assigned\_to)

## Functions

### make\_allocation(self, allocate)

### \_\_init\_\_(self, name=None, assigner=None, due\_date=None, allocation=None, description=’’, status=TaskStatus.CREATED, requires\_verification=False, team=None, user=None)

# TaskNote

## Special Atribs

### Table name = task\_notes

### Serialize – id, task, image, user, note, end\_status, recept, timestamp, creator.name

## Fields

### id – int, primary key

### task – int, foreign key(tasks.id), not null

### image – int, foreign key(images.id)

### user – int, foreign key(users.id), not null

### note – string, not null

### end\_status – enum(TaskStatus), not null

### recept – Float

### timestamp – DateTime, not null

### creator – relationship(User, backref=created\_notes)

### picture – relationship(Image, backref=note\_image)

## Functions

### Spend(self, amount, status)

\_\_init\_\_(self, task, user, end\_status, image=None, recept=0, note=’’)