Django DBMS Project

Team members:
Chao Chen
Yihang Zhao
Saheed Adepoju

Outline

- Introduction

- Requirements Analysis

- Implementation

- Demo

Introduction

- Project commitment:

  Delivering a web-based application for resource management and regular works conduct of general purpose store

- Users:

  Store Manger, Staff

Requirements Analysis

- User Story

Scenario 9
select Workflow Menu.
select Acitvity Function.
create Discount For specific product.
select Valid date For discount.
select Target For Membership or birthday.
submit button to complete.

Scenario 10
select workflow menu
select Acitvity Function.
select create promotion.
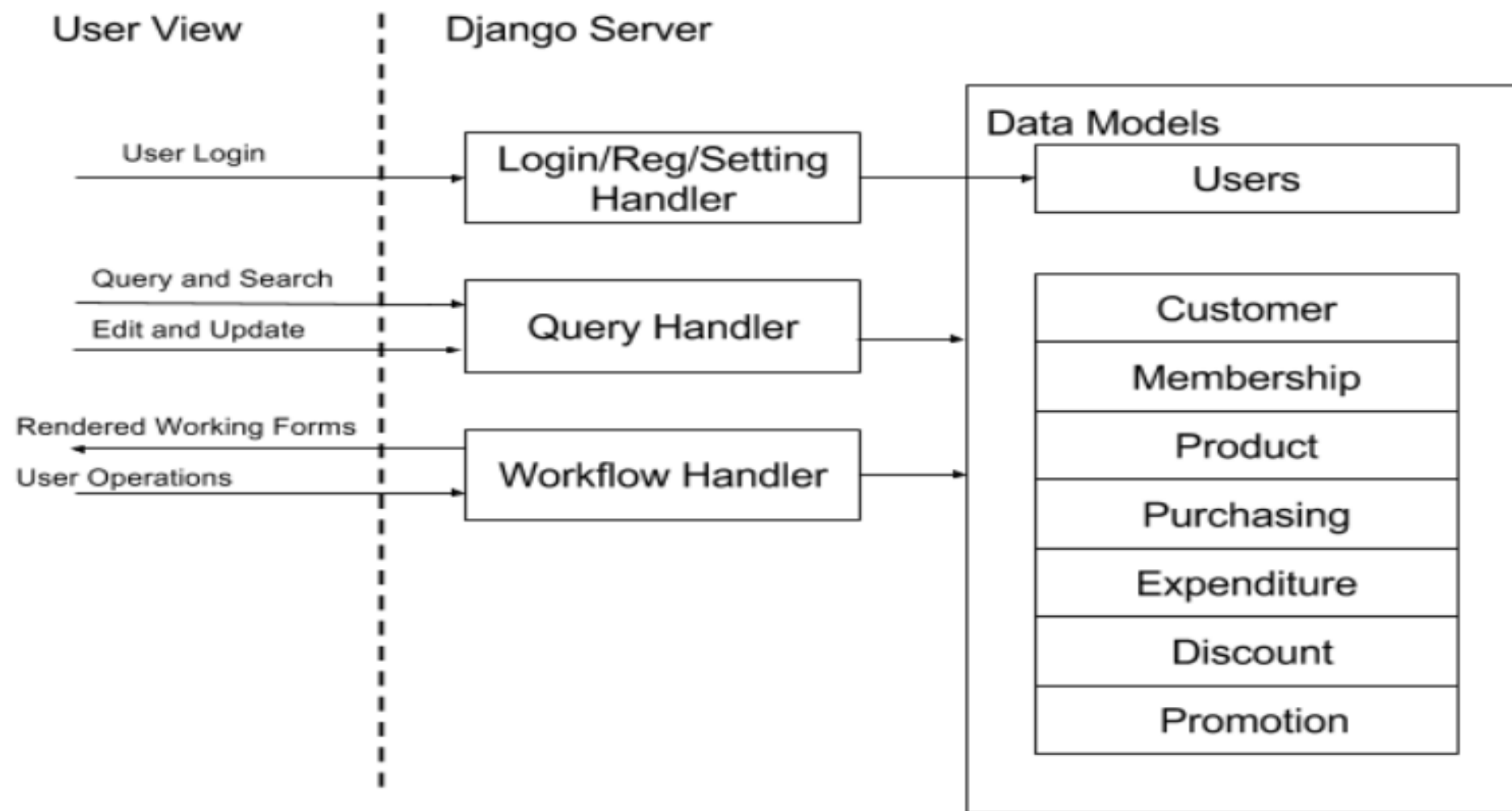select valid date for discount.
select product for Promotion.
give value for Buy N get one.
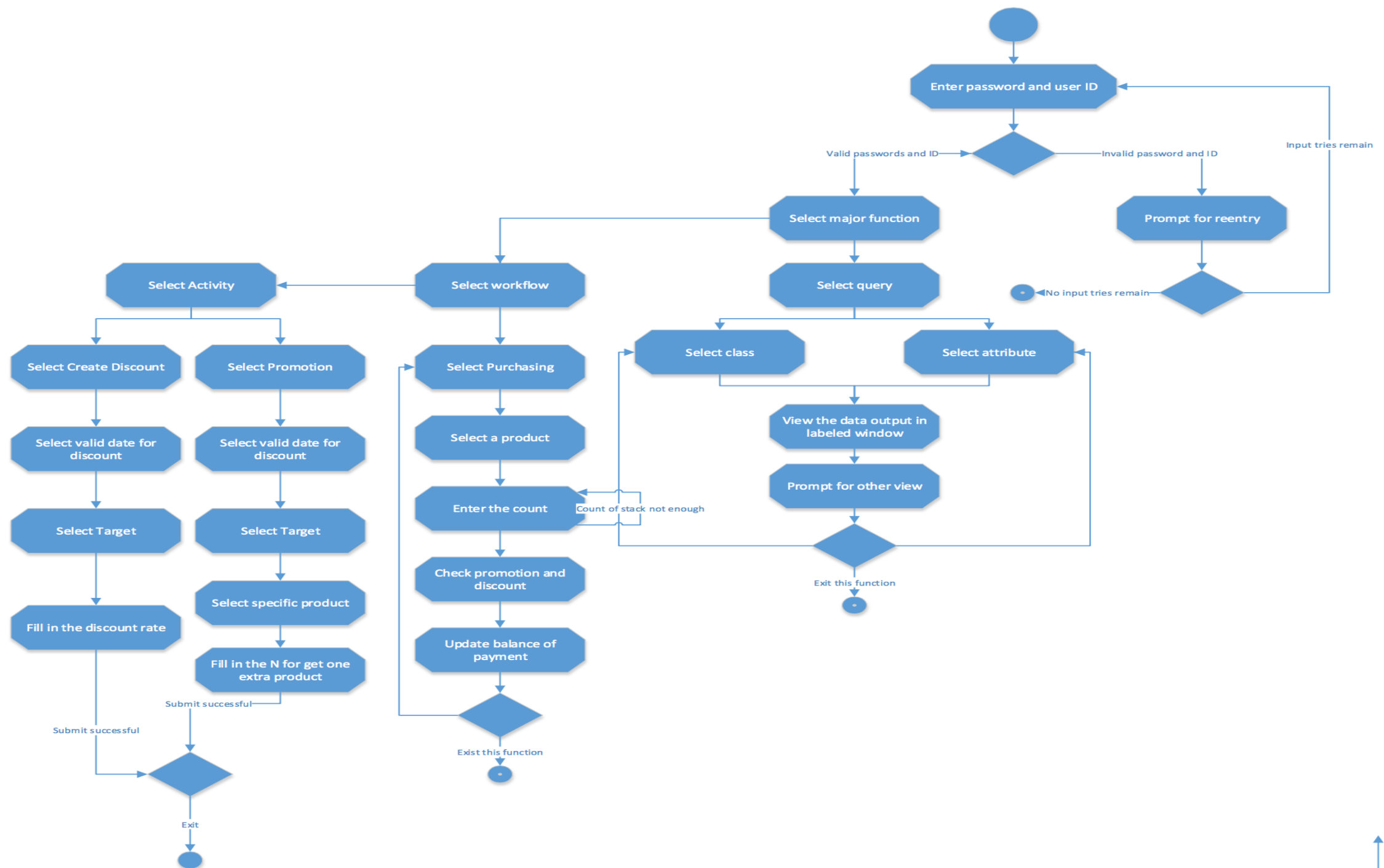submit button for complete.
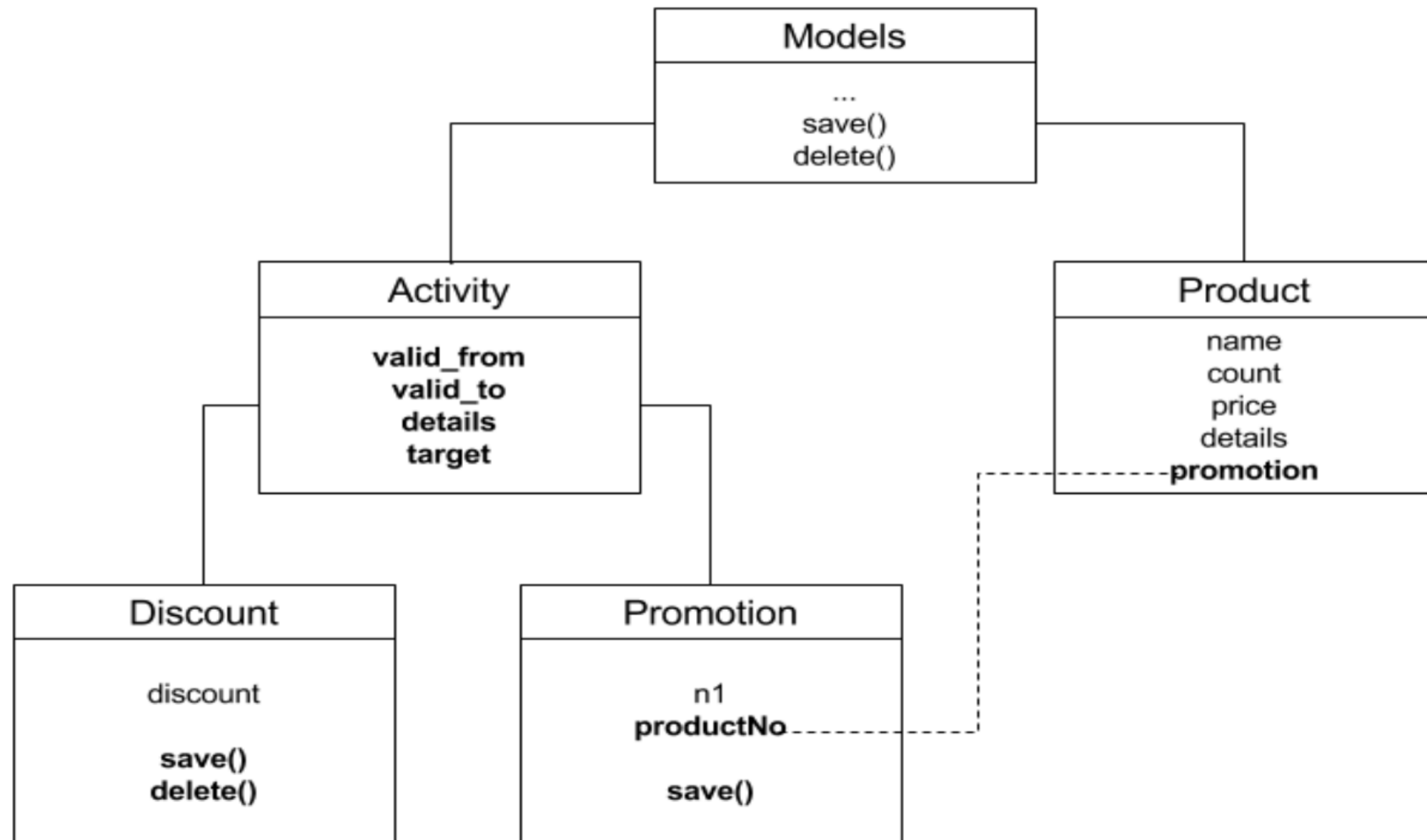
Design

- Architecture

Design

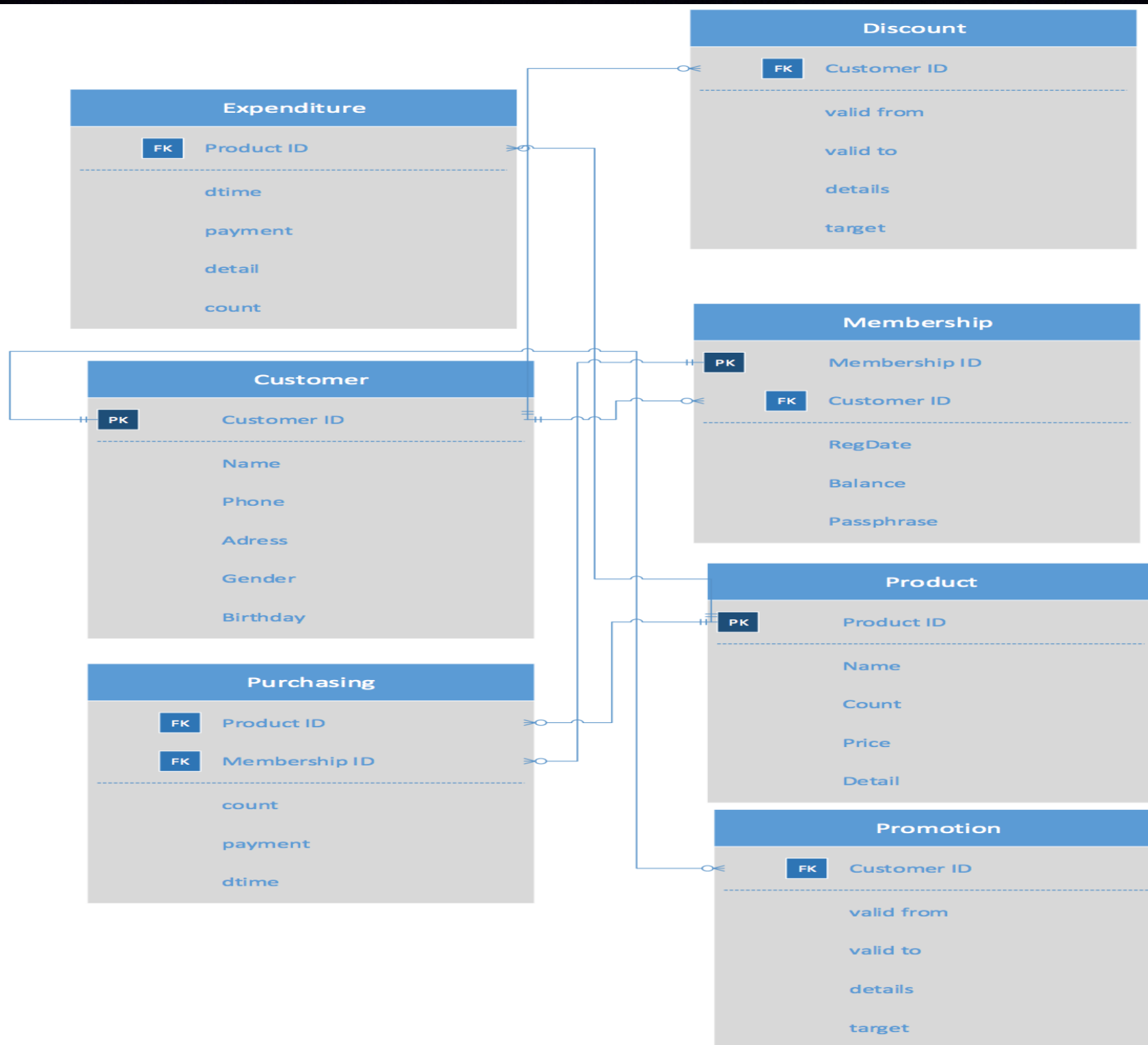- Activity Diagram

Design

- Class an Components Diagram

Database

Implementation

- Server Host:

  - Python2.7

  - Django 1.8.4

  - MySQL(Optional)

- UI and Interaction

  - JavaScript

  - jQuery/FancyBox/Bootstrap

Implementation

- Data Model

```python
# cchen @ 20161009
class Activity(models.Model):
    valid_from = models.DateTimeField(auto_now_add=True)
    valid_to = models.DateTimeField(auto_now_add=True)
    details = models.CharField(max_length=255)
    target = models.CharField(max_length=32)


# saheed @ 20161010
# Discount for all products
class Discount(Activity):
    disc = models.DecimalField(max_digits=19, decimal_places=2)

    def save(self):
        Discount.objects.all().delete()
        super(Discount, self).save()


# Yihang Zhao @  20161011
# Promotion for specific product
class Promotion(Activity):
    productNo = models.ForeignKey(Product, related_name='promotion', blank=True, null=True, on_delete=models.SET_NULL)
    n1 = models.BigIntegerField(null=True)
    #pmType = models.CharField(max_length=32)
    #discount = models.DecimalField(max_digits=19, decimal_places=2, null=True)

    def save(self):
        try:
            try:
                self.productNo.promotion.get()
                print 'replace the promotion of ' + self.productNo.name
                self.productNo.promotion.clear()
            except ObjectDoesNotExist:
                print 'place the promotion of ' + self.productNo.name
        except ObjectDoesNotExist:
            print 'specified product not exist.'
```

Implementation

- View Render and Request Handler

```python
# cchen @ 20161020
# Query discount and promotion info
@login_required
def act_info_get(request):
    productNo = request.POST.get('product')

    n1 = 0
    disc = 1.0
    promo_tgt = 'All'
    disc_tgt = 'All'

    try:
        today = datetime.datetime.now().date()

        # get active discount info
        dclist = Discount.objects.all()
        active_discount = None
        if len(dclist) > 0:
            active_discount = dclist[0]
        if active_discount and active_discount.valid_from.date() <= today and active_discount.valid_to.date() >= today:
            disc_tgt = active_discount.target
            disc = active_discount.disc

        # get promotion info for selected product
        promo = Promotion.objects.get(productNo=productNo)
        print promo
        if promo.valid_from.date() <= today and promo.valid_to.date() >= today:
            promo_tgt = promo.target
            n1 = promo.n1

    except:
        print "act_info_get:", sys.exc_info()[0]


    return JsonResponse({"n1": n1, "disc": disc, "promo_tgt":promo_tgt, "disc_tgt":disc_tgt})
```

Demo