Commerce Bank Project

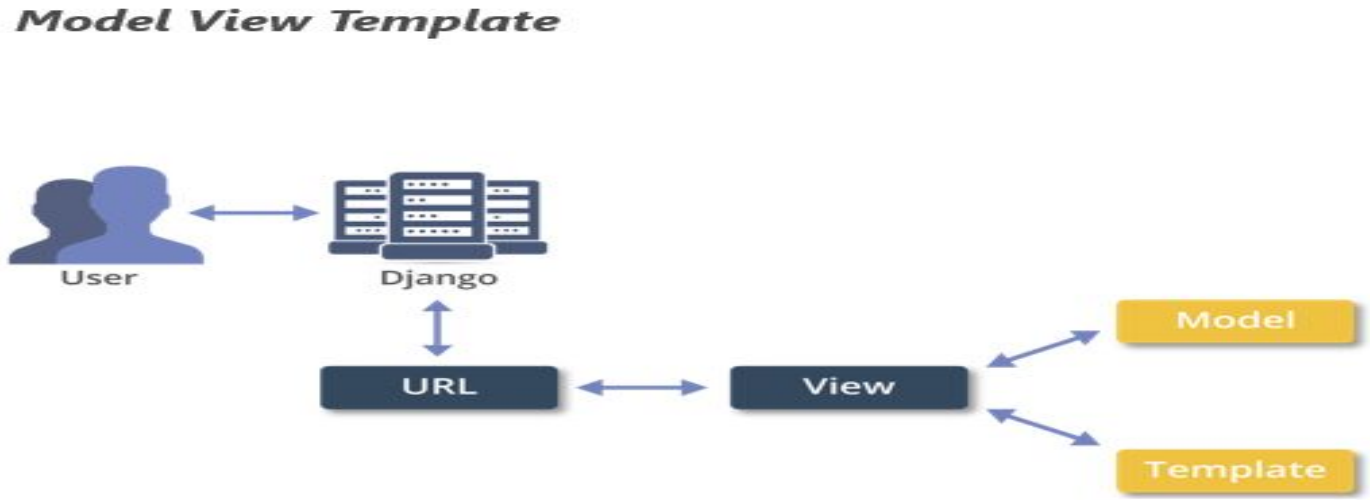
Coy Kwan - Project Manager
Cori Mroz - Fullstack
Anna Johnson - Database/Backend
Feng Zheng - Fullstack/testing
Daylan Quinn - Front End

Tech Stack

- Python
- Django
- HTML
- CSS
- Bootstrap
- MySQL

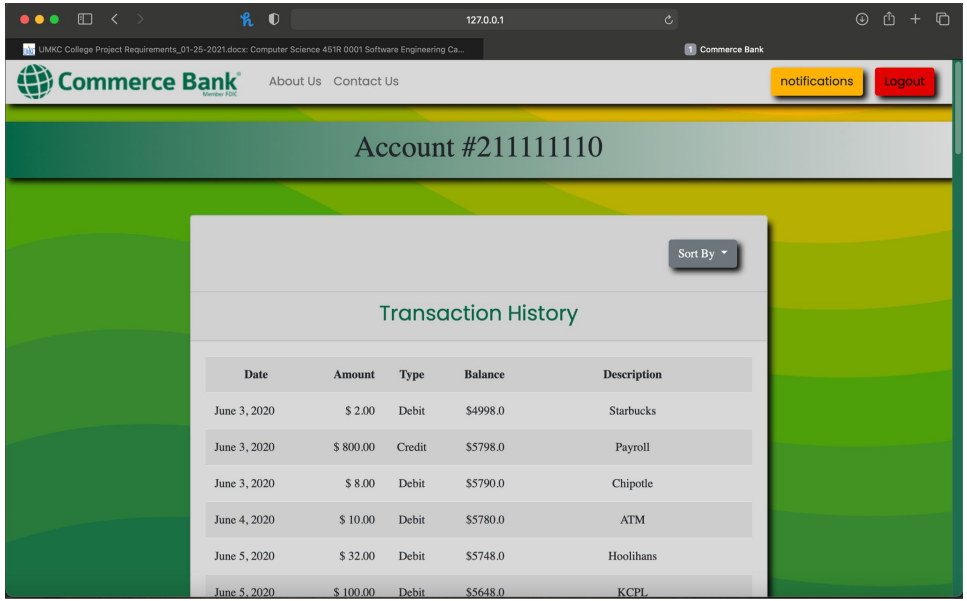
Django Recap

- Model -> View -> Template hierarchy
  - Models for database abstraction, views for middleware/api, templates for front end
- Provides admin tool for quick data visualization, manipulation for manual testing and added functionality.



Dashboard

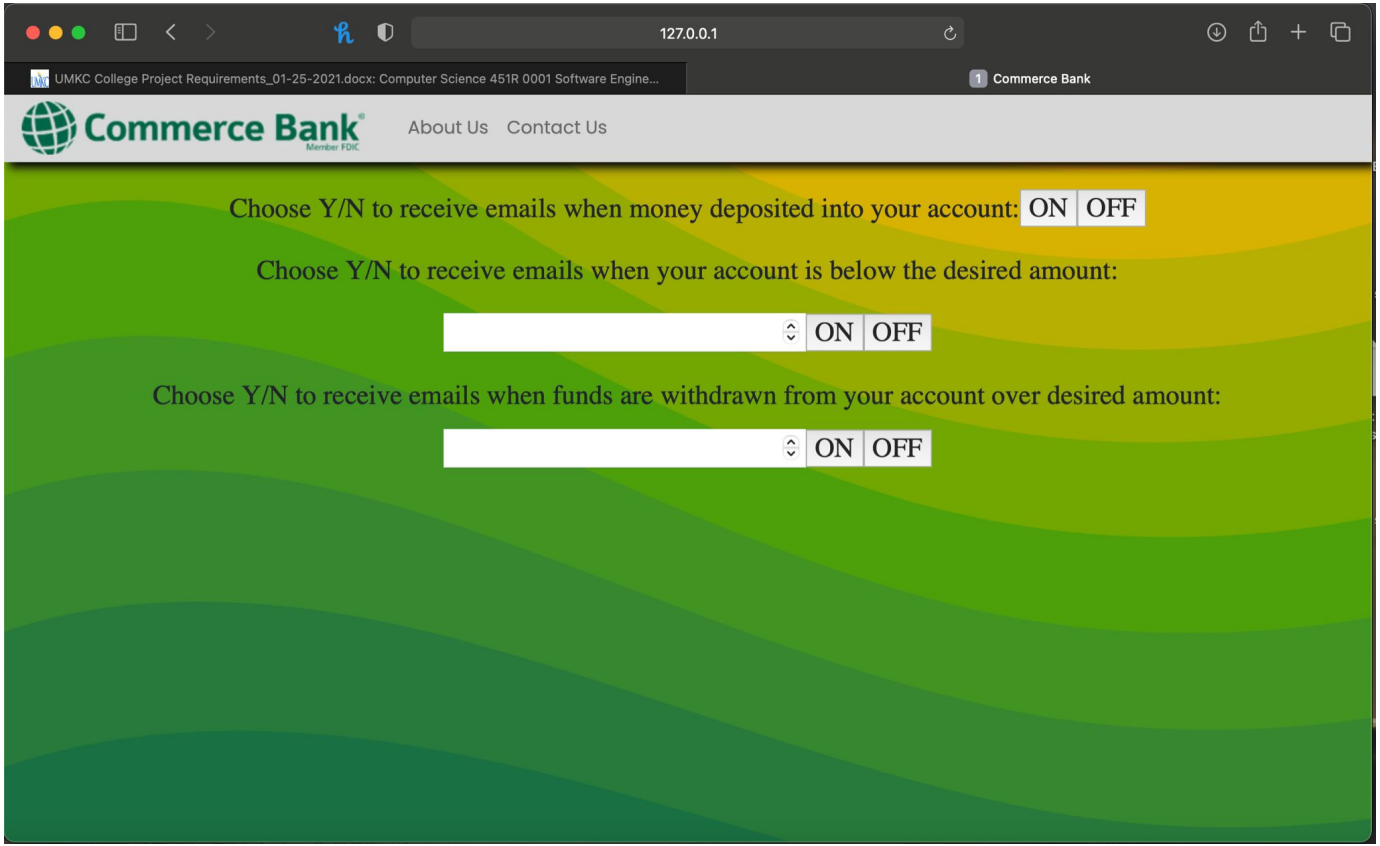
- General transaction information
  - Ability to retroactively sort transactions by any of the notification rules
  - Access to notification settings



Notification Center

- Three basic rules
  - When money is deposited
  - When account drops below a threshold set by a user
  - When a withdrawal exceeds a threshold
- Notifications are set via email
  - Emails currently only “sent” locally however can be provided with a SMTP port for true functionality (All the decent email servers we found cost money)
  - `EMAIL_BACKEND =`
  - `‘django.core.mail.backends.smtp.EmailBackend’`
  - `EMAIL_HOST = ‘smtp.gmail.com’`
  - `EMAIL_USE_TLS = True`
  - `EMAIL_PORT = 12345`
  - `EMAIL_HOST_USER = ‘your_account@gmail.com’`
  - `EMAIL_HOST_PASSWORD = ‘your account’s password’`

Notification Center



Database

- Utilizing a MySQL database. Tables for:

```
class Transactions(models.Model):
    transaction_id = models.AutoField(primary_key=True)
    account_id = models.IntegerField(blank=True, null=True)
    processing_date = models.DateField(blank=True, null=True)
    balance = models.FloatField(blank=True, null=True)
    transaction_type = models.CharField(max_length=10, blank=True, null=True)
    amount = models.FloatField(blank=True, null=True)
    descr = models.CharField(max_length=50, blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'transactions'
```

```
class WithdrawalNotif(models.Model):
    account_id = models.IntegerField(primary_key=True)
    is_true = models.CharField(max_length=1)
    amount = models.FloatField(blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'withdrawal_notif'
```

```
class DepositNotif(models.Model):
    account_id = models.IntegerField(primary_key=True)
    is_true = models.CharField(max_length=1)

    class Meta:
        managed = False
        db_table = 'deposit_notif'
```

```
class BalanceNotif(models.Model):
    account_id = models.IntegerField(primary_key=True)
    is_true = models.CharField(max_length=1)
    amount = models.FloatField(blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'balance_notif'
```

Database and Middleware

- Data represented in DDL like models in Django’s MVT structure

- Views handle the logic allowing front end, templates, to talk to database
  - HTTP GET and POST requests
  - Rapidly speeds up development times

Security

- Django natively protects against
  - Cross site scripting
  - Request forgery using built in secret keys for HTTP requests
  - MVT protects against SQL injection to an extent

- Measures we have taken
  - Completely parameterized queries - no raw sql
  - Inputs heavily restricted

Stretch Goals

- Github used for source control
  - Pull requests performed to combine all of the different pieces into the main branch
  - Code review done throughout design process
- Notifications sent via email

```
Terminal: Local
[07/May/2021 11:03:42] "GET /admin/jsi18n/ HTTP/1.1" 200 3195
transaction was a credit
Content-Type: text/plain; charset="utf-8"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Subject: Credit Update - Commerce Bank
From: coykwan@gmail.com
To: daylanq10@gmail.com
Date: Fri, 07 May 2021 16:04:05 -0000
Message-ID: <162040344579.11642.14220460741624278824@daylans-mbp.lan>

This email is to notify that a Credit has been added to your account.
--------------------------------------------------------------------------------
<class 'float'>
<class 'float'>
Problems  Python Packages  TODO  Terminal  Python Console  Event Log
Python 3.8 (commerce-site)
```

Email Notification Code

commerce-site – models.py

commerce-site 〉 banking 〉 models.py

```python
@receiver(post_save, sender=Transactions, dispatch_uid="check_deposit")
def check_deposit(sender, instance, **kwargs):

    if instance.transaction_type=="CR":
        query_results = DepositNotif.objects.get(account_id = instance.account_id)
        if query_results.is_true == "1":
            print("transaction was a credit")
            send_mail(
                'Credit Update - Commerce Bank',
                'This email is to notify that a Credit has been added to your account.',
                'coykwan@gmail.com',
                ['daylanq10@gmail.com'],
                fail_silently=False,
            )

    if instance.transaction_type=="DR":
        query_results = WithdrawalNotif.objects.get(account_id = instance.account_id)
        if query_results.is_true == "1":
            if instance.amount >= query_results.amount:
                print("transaction was a large withdrawal")
                send_mail(
                    'Withdraw Update - Commerce Bank',
                    'This email is to notify that an amount over $100 has been withdrawn from your account.',
                    'coykwan@gmail.com',
                    ['daylanq10@gmail.com'],
                    fail_silently=False,
                )
```

check_deposit()

Terminal: Local

Problems | Python Packages | TODO | Terminal | Python Console | Event Log

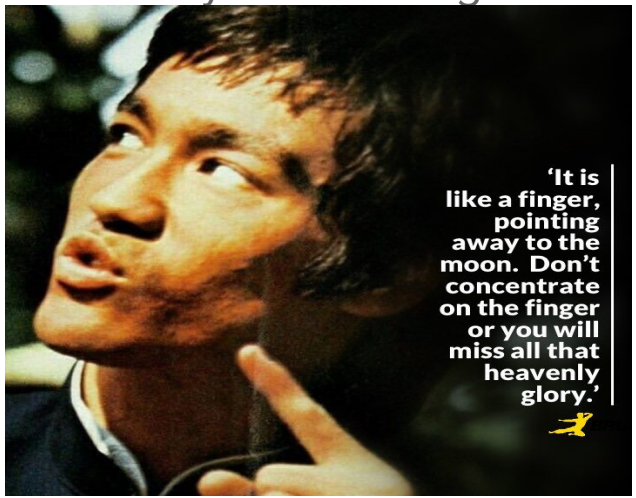
179:1 LF UTF-8 4 spaces Python 3.8 (commerce-site)

Live Demo

https://umkc.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=80a8921c-8c15-4fb3-8a3d-ad210117303d

Risks and lessons learned

- HTTP requests are a hard concept for first and second time users even with built in view API
  - This should've been attacked much earlier
- Division of functionality would allow for faster and easier development
  - Initial planning of a hyper mobile friendly single page responsive design did not lend itself well to division of labor
- If you spend too much time thinking about a think you'll never get it done
  - Planning is great but action is better
  - Slow and steady wins the race



Questions?