

中山大学数据科学与计算机学院本科生实验报告

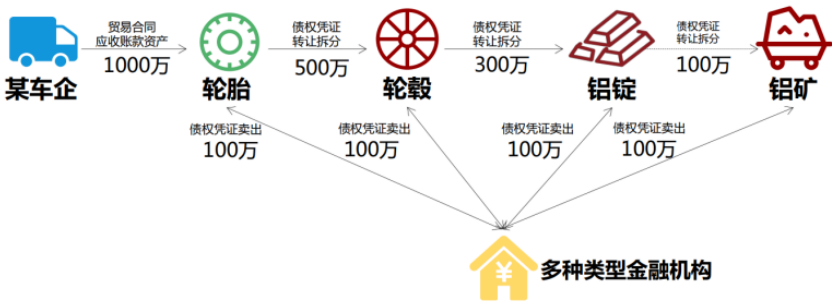
(2019年秋季学期)

课程名称：区块链原理与技术

任课教师： 郑子彬

年级	大三	专业（方向）	数据科学与计算机学院 软件工程专业
学号	17343132	姓名	许帆君
电话	15837998448	Email	2022878615@qq.com
开始日期	2019.12.6	完成日期	2019.12.13

一、项目背景



传统应收账款融资存在的问题

我国应收账款规模巨大，其中很大一部分来自于中小企业，生产总值约占GDP60%，但这些庞大的账面资金却很难给中小企业带来实际的融资帮助，很多小微企业在应收账款融资方面仍然面临诸多的困难。这是因为传统应收账款融资还存在许多痛点问题

- 信用体系不完善，中小企业融资难

核心企业有充沛的信用资源，但是在多级供应商模式中，传统的保理、应收账款质押、票据贴现等应收类供应链金融模式只能满足核心企业上游的一级供应商的融资需求，却无法将核心企业信用穿透给下游中小企业。所以由于信担保体系的不健全，加上中小企业可能出现的资产负债率高等原因，银行等放贷平台无法有效识别其发展潜力，往往抵押条件苛刻，资本市场的准入门槛非常高，导致中小企业在应收账款融资方面普遍存在融资难、融资贵等问题。

- 供应链条不互通，贸易信息不透明

在整个供应链中，各个参与企业间的信息并不完全互通，除了核心企业和一二级供应商外，其他中小企业的信息化程度较低，贸易信息无法做到实时共享，交易的真实性难以有效校验，进一步增大了向金融机构获取授信支持的难度。

- 操作方法局限多，流转保理确权难

传统的应收票据和应收账款等流动资产作为信用凭证的操作方法局限较多，流转困难，而且难以防范操作风险和欺诈风险，容易形成坏账，这些问题严重制约了应收账款融资业务发展。例如银票商票无法拆分、贴现门槛高且融资流程长、保理确权非常困难。

引入区块链技术的可行性

区块链技术具体的去中心化、信息不可篡改、去信用化等特性，如果把区块链技术应用到应收账款融资领域，建立一个基于区块链技术的应收账款链，参与各方通过此链完成交易，且交易能够获得该链的有效确认和记录，任何人都可以参与其共识过程，无疑能够解决诸多传统应收账款融资无法解决的问题。

区块链+供应链金融

将供应链上的每一笔交易和应收账款单据上链，同时引入第三方可信机构来确认这些信息的交易，例如银行，物流公司等，确保交易和单据的真实性。同时，支持应收账款的转让，融资，清算等，让核心企业的信用可以传递到供应链的下游企业，减小中小企业的融资难度。

二、 方案设计

1、 业务逻辑



2、 存储方案

- 公司结构体

```
1 struct Company{
2     string name;           //company name
3     uint credit;           //company credit
4     address addr;          //company address
5     string private_key;    //password
6 }
```

结构体 `Company` 中存储供应链上的公司信息，签发账款交易的功能通过创建新的账款交易上链完成

- 账款结构体

```
1 struct Receipt{
2     address buyer_addr;    //address of buyer
3     address solder_addr;   //address of solder
4     uint issure_time;      //issue time of the reciept
5     uint due_time;         //deadline to pay the reciept
6     string good;           //name of the good
7     uint price;            //loan amount
8     bool transfered;       //identify the receipt is transferred or
9     not
10 }
```

结构体 `Receipt` 中存储“采购商品--应收账款”的信息，签发账款交易的功能通过创建新的账款交易上链完成

- 账款链结构体

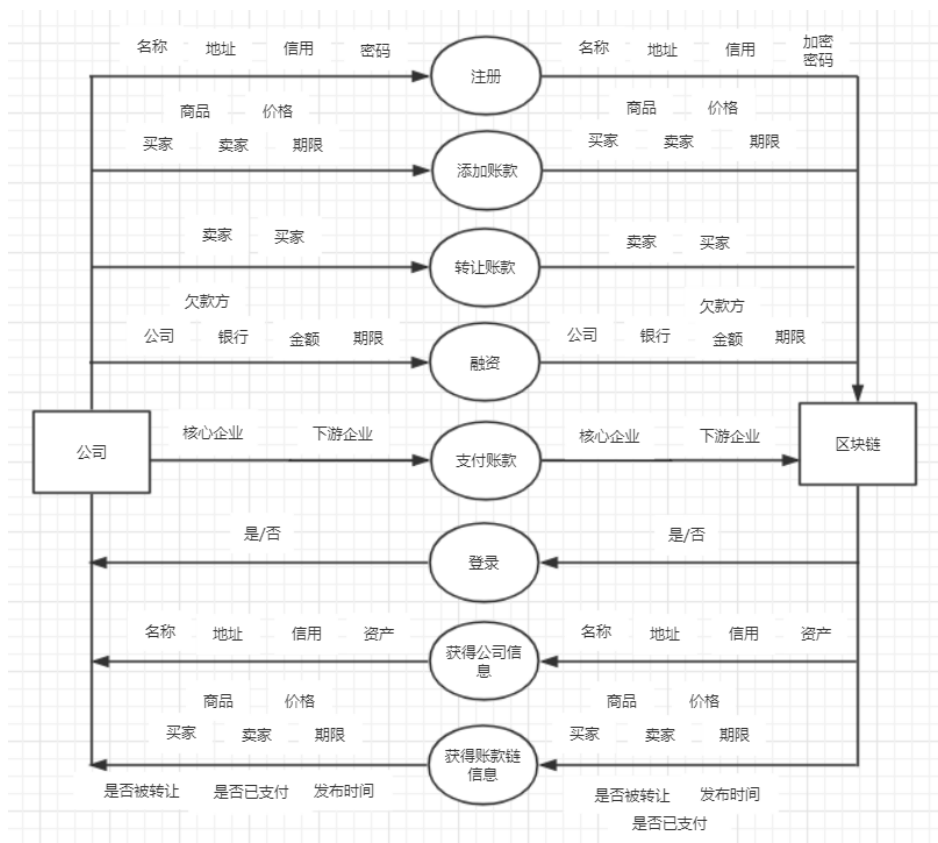
```
1 struct Receipt_chain{
2     bool paid;              //whether the receipts has been paid
3     Receipt[] receipts;    //all receipts on the chain
4 }
```

账款链中储存着该转让账款链上的所有账款信息，并且元素 `paid` 唯一标识该账款链是否已经被结清

- 融资数据结构体

```
1 struct Finance{
2     address bank_addr;     //address of the bank
3     address company_addr;  //address of the company
4     Receipt[] receipts;    //receipts used for financing
5     uint money;            //loan amout
6     uint due_time;         //due time
7 }
```

3、数据流图



4、核心功能

(1) 实现采购商品—签发应收账款交易上链

- 功能说明

- 签发应收账款实现了企业以账款的形式向其他企业购买物资的能力，通过将账款上链保证了账款的可信
- 企业进行账款发行的信息应该包括：
 - 买方、卖方企业信息、购买货物、总金额、发行时间、还款截止时间、标记账款是否被转让的标识位
- 签发账款交易的功能通过创建新的账款交易上链完成

- 新建账款 (internal)

```

1 function make_receipt(address _buyer, address _solder, string _good,
  uint _price, uint _due_time)internal view returns (Receipt){
2     Receipt memory new_receipt = Receipt({
3         buyer_addr : _buyer,
4         solder_addr : _solder,
5         good : _good,
6         price : _price,
7         issure_time : now,          //发行时间为该函数被调用时的系统时间
8         due_time : _due_time,
9         transfered : false          //初始状态的“transferd”为“false”，一旦账款
    被转让或用于融资，该参数被设置为“true”
10    });
11    return new_receipt;
12 }

```

• 贷款函数——新建账款链 (public)

```

1 //-----issue an account and create a new chain-----
  -----
2     Receipt_chain new_chain;
3
4     function issue_account(address _buyer, address _solder, string
      _good, uint _price, uint _due_time)public returns(bool){
5         //创建新的账款链
6         Receipt memory new_receipt = make_receipt(_buyer, _solder,
          _good, _price, _due_time);
7
8         //初始化账款链
9         new_chain.receipts.length = 0;          //设置链长度
10        new_chain.paid = false;                  //初始化“paid”属性
11        new_chain.receipts.push(new_receipt);    //在账款链上添加第一个账款
12        //将新的账款链加入到账款链的链表中
13        receipt_chains.push(new_chain);
14
15        //emit event
16        emit func_issue_account(_buyer, _solder);
17
18        return true;
19    }

```

(2) 实现应收账款的转让上链

• 功能说明

- 应收账款转让实现了贷款能力从上游企业过渡到下游的能力
- 企业进行账款转让需要提供的信息应该包括：
 - 被转让的账款信息、一个完备的新的账款信息（本次购买的货物，价格，支付时间等）
- 被转让的账款信息被用来查找链上的已有账款，因此只需要提供足以标识一个账款单的参数，包括：
 - 该账款单的买方、该账款单的卖方

- 查找账款链 (internal)

```
1 //该函数用于查找旧的receipt并返回
2 function find_receipt_chain(address _buyer, address _solder)internal
  view returns(uint256){
3     //寻找对应收据
4     for(uint i = 0; i < receipt_chains.length; i++){
5         for(uint j = 0; j < receipt_chains[i].receipts.length; j++){
6             if(receipt_chains[i].receipts[j].buyer_addr == _buyer
7                 && receipt_chains[i].receipts[j].solder_addr == _solder
8                 && receipt_chains[i].paid == false)
9                 return i;
10        }
11    }
12
13    //“10000”是最大值，意味着没有找到收据
14    return 10000;
15 }
```

- 实现账款转让功能 (public)

```
1     function account_transfer(address _prev_buyer, address _buyer,
  address _solder, string _good, uint _price, uint _due_time)public
  returns(bool){
2         //找到对应的账款链
3         uint256 index = find_receipt_chain(_prev_buyer, _buyer);
4         //未找到对应的收据则直接返回
5         if(index == 10000)return false;
6
7         Receipt_chain memory tmp = receipt_chains[index];
8         //账款链不符合要求，返回false
9         if(tmp.receipts[tmp.receipts.length - 1].solder_addr != _buyer)
  return false;
10
11         //将转让信息添加到账款链上
12         //将旧账款的'transferred'属性设为“true”
13         receipt_chains[index].receipts[tmp.receipts.length -
  1].transferred = true;
14         //创建一个新账款用于保存转让信息
15         Receipt memory new_receipt = make_receipt(_buyer, _solder,
  _good, _price, _due_time);
16         //添加新账款到账款链
17         receipt_chains[index].receipts.push(new_receipt);
18
19         //emit event
20         emit func_transfer(_buyer, _solder, index);
21     }
```

(3) 实现应收账款向银行融资上链

- 功能说明

- 应收账款向银行融资实现了通过企业的贷款单来证明企业还款能力，从而向银行融资的功能

- 企业通过账款向银行进行融资需要提供的信息应该包括：
 - 银行信息、企业信息、融资金额、还款时间、用于证明还款能力的贷款单
- 贷款单参数需要通过给定的信息从链上的所有贷款单中进行查询，保证贷款单的真实性与有效性。

需要提供的查询参数包括：

- 贷款单的买方、贷款单的卖方

• 账款单查询 (internal)

```

1      Receipt blank;
2      function find_receipt_unused_and_set(address buyer, address solder)
internal returns (Receipt){
3          //找到未被转让过的对应账款
4          for(uint i = 0; i < receipt_chains.length; i++){
5              for(uint j = 0; j < receipt_chains[i].receipts.length; j++){
6                  if(receipt_chains[i].receipts[j].buyer_addr == buyer
7                      && receipt_chains[i].receipts[j].solder_addr == solder
8                      && receipt_chains[i].receipts[j].transferred == false){
9                      receipt_chains[i].receipts[j].transferred = true;
10                     return receipt_chains[i].receipts[j];
11                 }
12             }
13         }
14         //如果没有找到则返回空账款
15         return blank;
16     }

```

• 银行融资功能 (public)

```

1      Finance new_finance;
2      function financing(address _bank_addr, address _company_addr,
address arrear_party, uint _money, uint _due_time)public returns(bool){
3          //初始化一个融资结构
4          new_finance.bank_addr = _bank_addr;
5          new_finance.company_addr = _company_addr;
6          new_finance.money = _money;
7          new_finance.due_time = _due_time;
8          delete new_finance.receipts;
9
10         //找到对应收据来证明公司有足够能力支付这笔钱
11         Receipt memory receipt =
find_receipt_unused_and_set(arrear_party, _company_addr);
12
13         //检查收据是否合理，以及是否有足够的资产批准公司的资产
14         if(receipt.buyer_addr == address(0) || receipt.price < _money)
return false;
15         new_finance.receipts.push(receipt);
16         //银行将贷款转让给公司
17         issue(_bank_addr, _company_addr, _money);
18         //将融资添加到融资链上
19         finances.push(new_finance);

```

```

20
21         //emit event
22         emit func_finance(_company_addr, _bank_addr);
23
24         return true;
25     }

```

(4) 实现应收账款支付结算上链

• 功能说明

- 支付账款上链使得最上游的企业能够层层对出资方（并非银行）进行还款
- 企业进行账款转让需要提供的信息应该包括：
 - 还款者地址信息
- 函数通过还款者地址找到该还款者所在的账款链，从链的下游到上游依次进行还款转账操作，其步骤为
 - 找到还款者所在的账款链，使得还款者是链的第一个买方
 - 从链的下游开始遍历
 - 还款者向目前查找到的账款单卖方所有者进行转账还款，金额为账单中的还款金额减去在该链下游卖方转移给下一个所有者的资产金额（剩余需支付金额）
 - 还款后将该链的信息删除（标志位设为已支付）

• 转账函数 (public)

```

1     function issue(address issuer, address receiver, uint amount) public
2     {
3         if(balances[issuer] < amount) return;
4         balances[issuer] -= amount;
5         balances[receiver] += amount;
6     }

```

• 还款函数 (public)

```

1     function payoff(address payer) public returns(bool){
2         //find the transaction
3         uint index = 0;
4         for(; index < receipt_chains.length; index++){
5             if(receipt_chains[index].paid == false &&
6                 receipt_chains[index].receipts[0].buyer_addr ==
7                 payer) break;
8         }
9         if(index >= receipt_chains.length) return;
10
11         //支付账款
12         //检查支付者是否有足够的金额来进行付款
13         if(balances[payer] < receipt_chains[index].receipts[0].price) return
14         false;
15
16         //将一定金额分别转给供应链上的所有企业

```



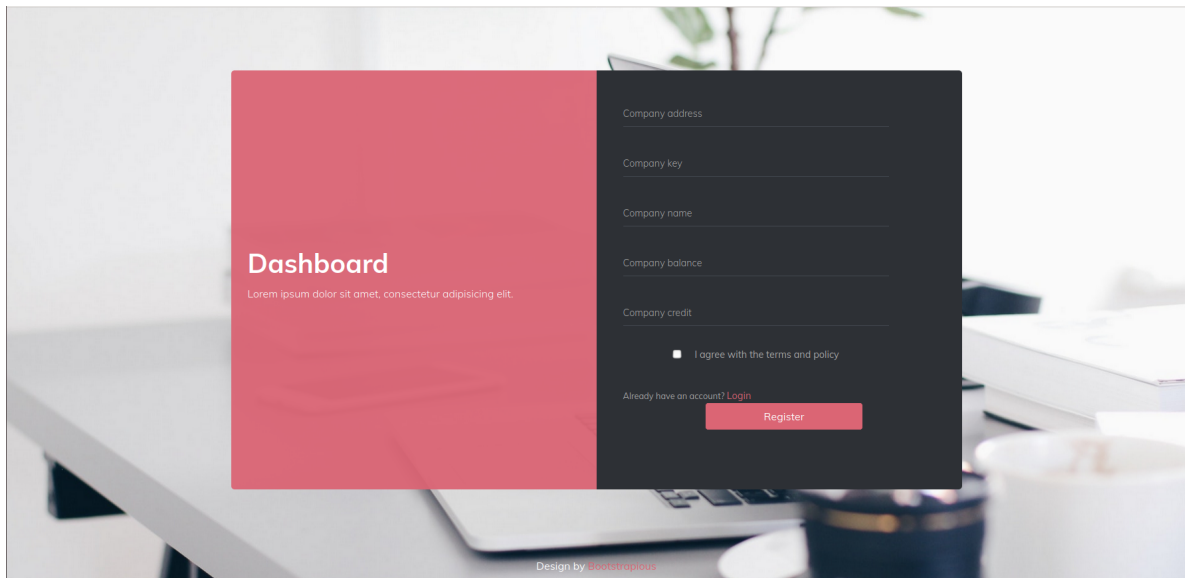
```

14     uint residue;           //需要支付的实际金额
15     uint paid_money = 0;    //已支付过的总金额
16     address receiver;      //发送地址
17     uint i = 0;
18     uint len = receipt_chains[index].receipts.length;
19     for(; i < len; i++){
20         receiver = receipt_chains[index].receipts[len - i - 1].solder_addr;
21         residue = receipt_chains[index].receipts[len - i - 1].price -
paid_money;
22         paid_money += residue;
23         balances[payer] -= residue;
24         balances[receiver] += residue;
25     }
26     //设置属性为已结清
27     receipt_chains[index].paid = true;
28
29     //emit event
30     emit func_payoff(payer, index);
31
32     return true;
33 }

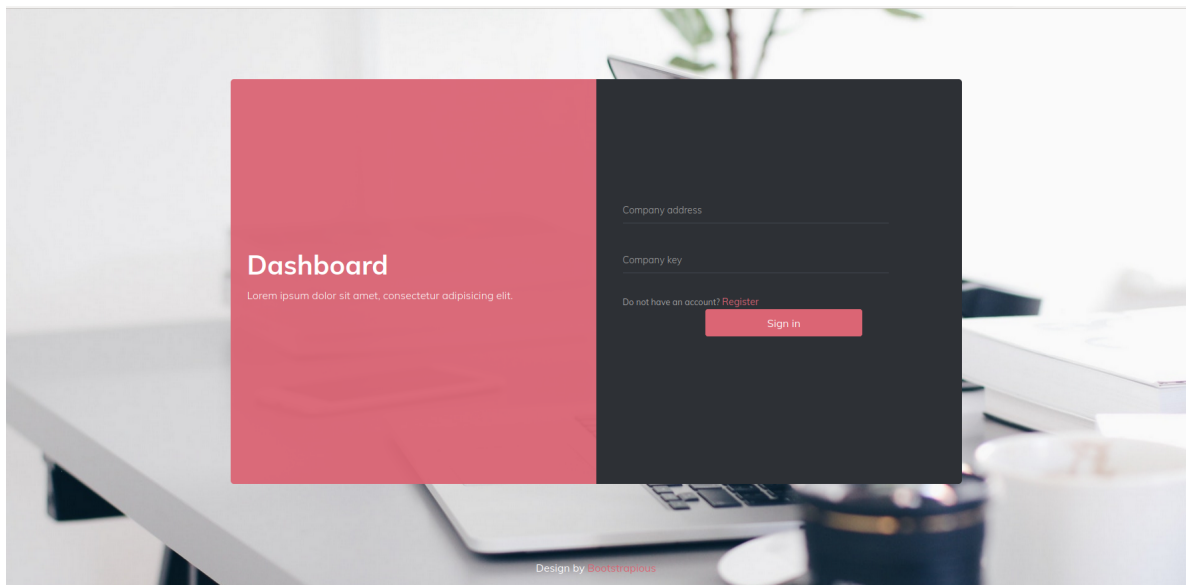
```

三、界面展示

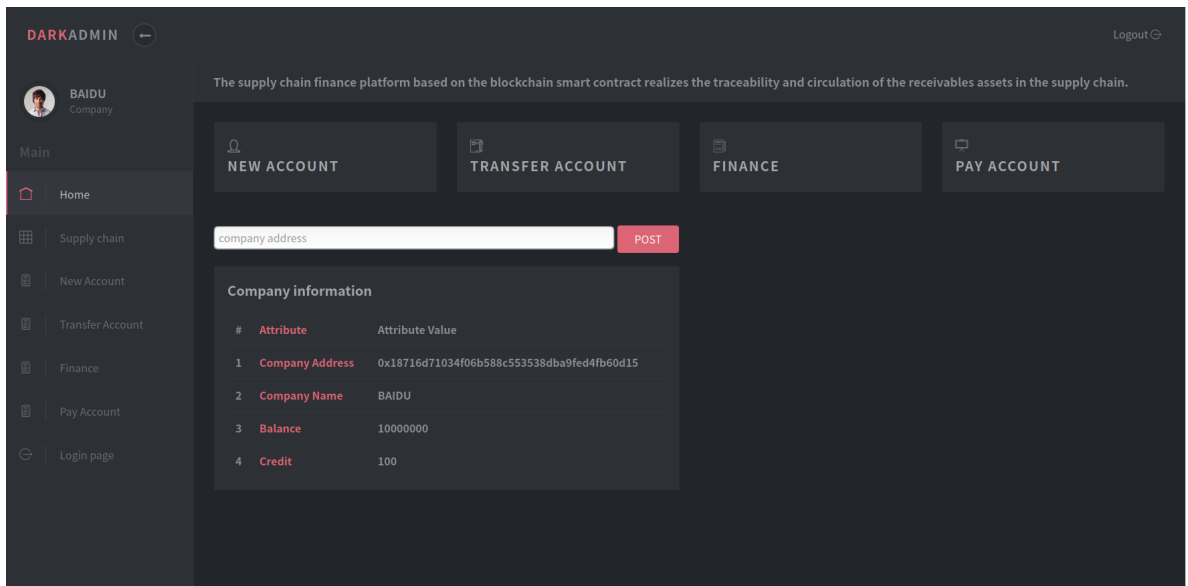
1、注册界面



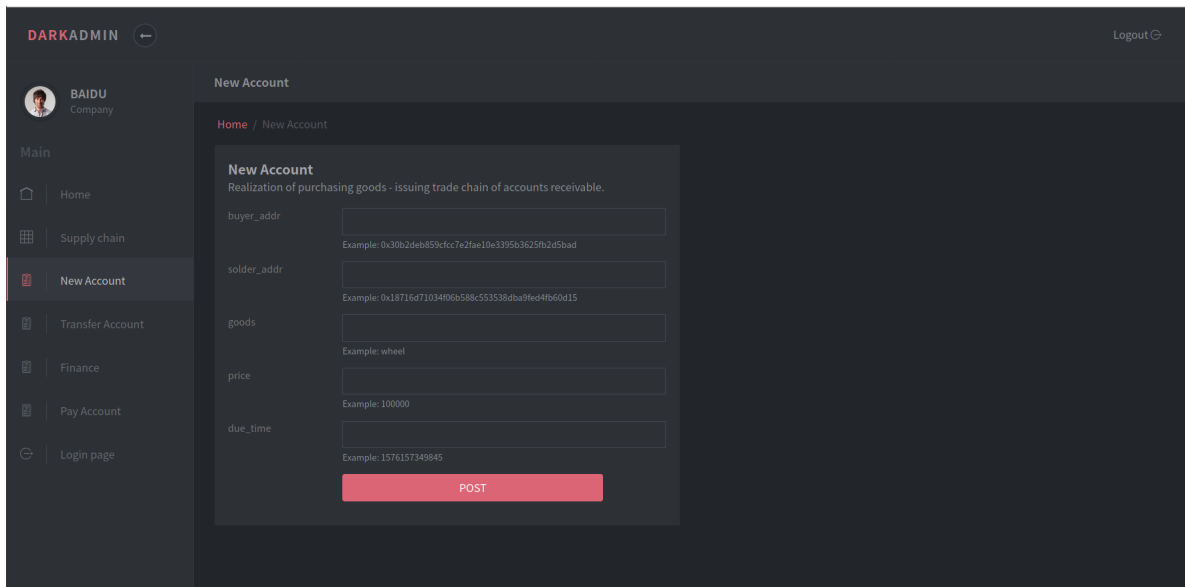
2、登陆界面



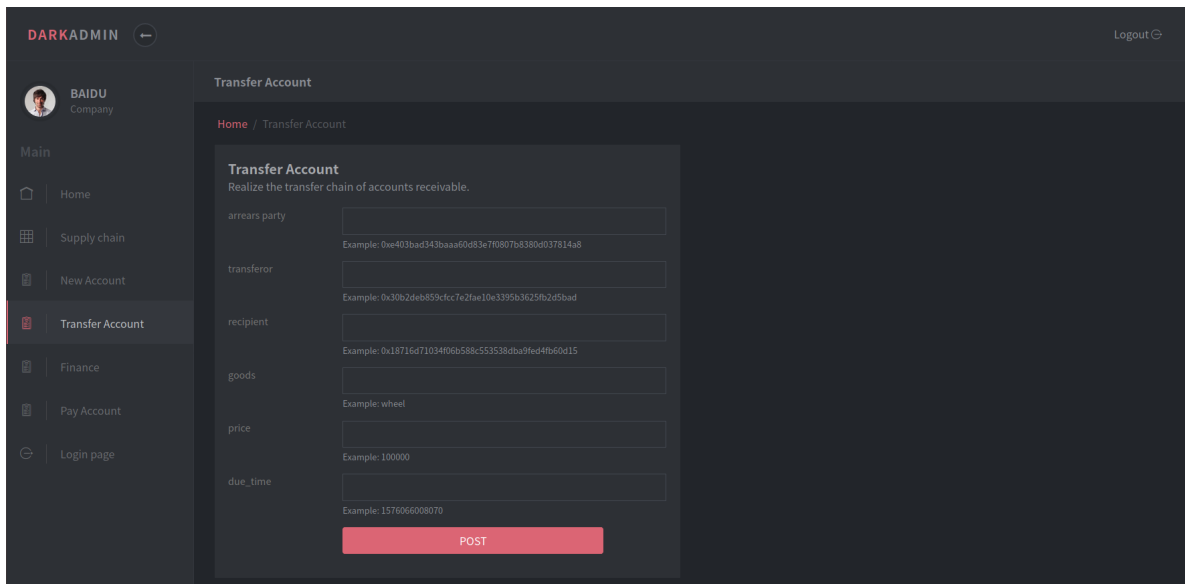
3、HOME界面



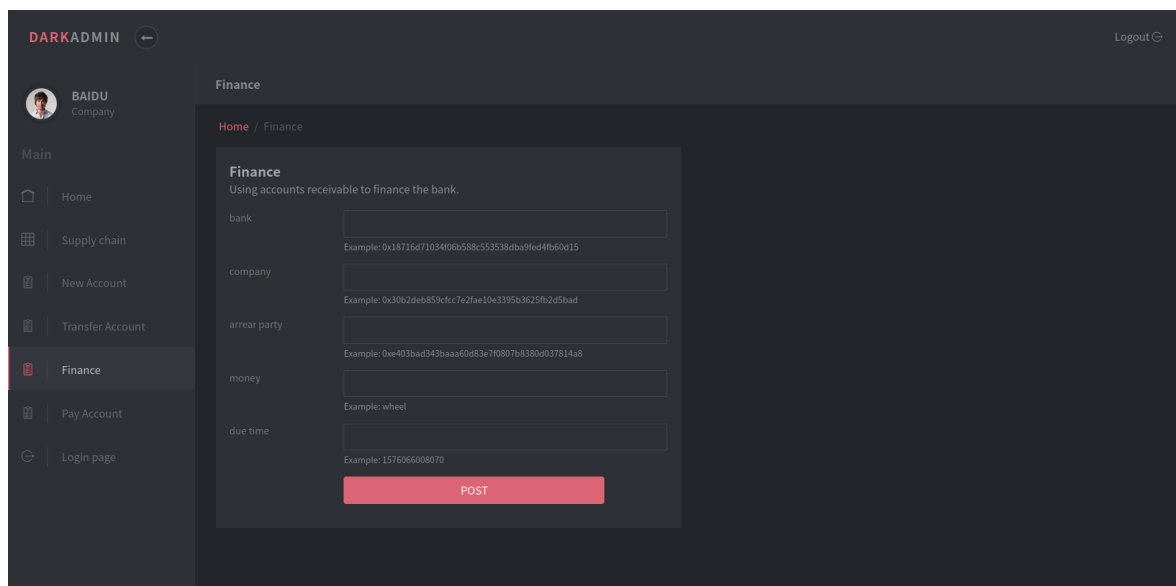
4、发布账款界面



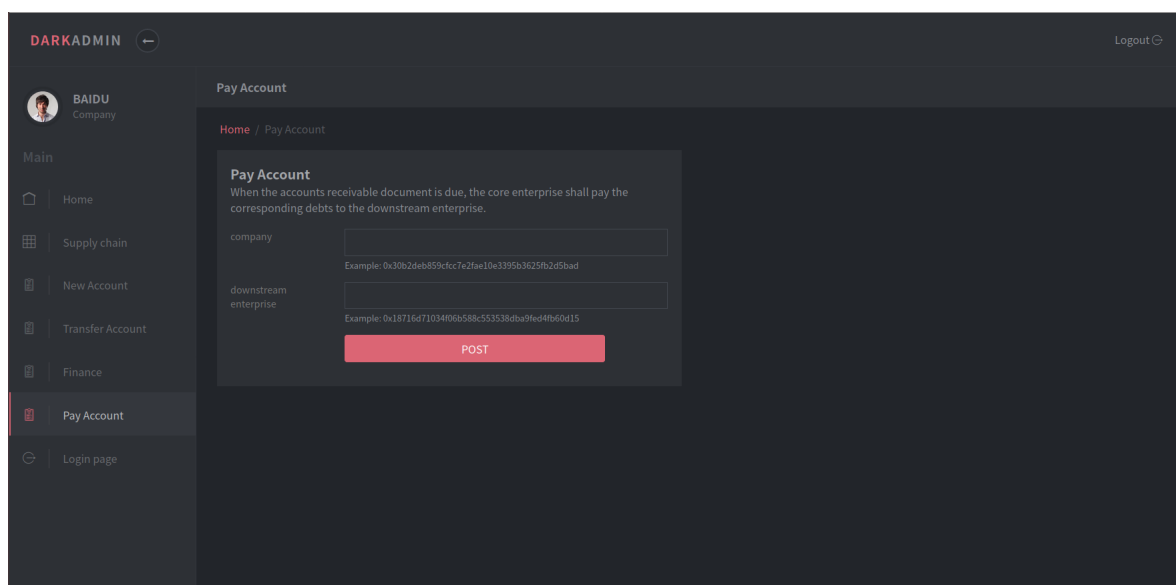
5、账款转让界面



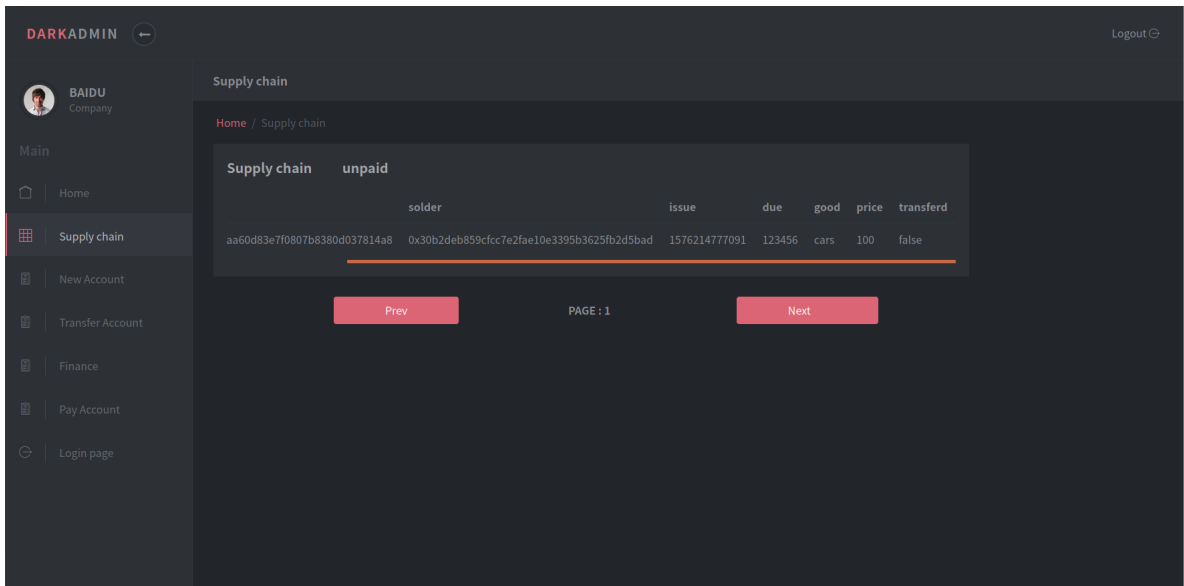
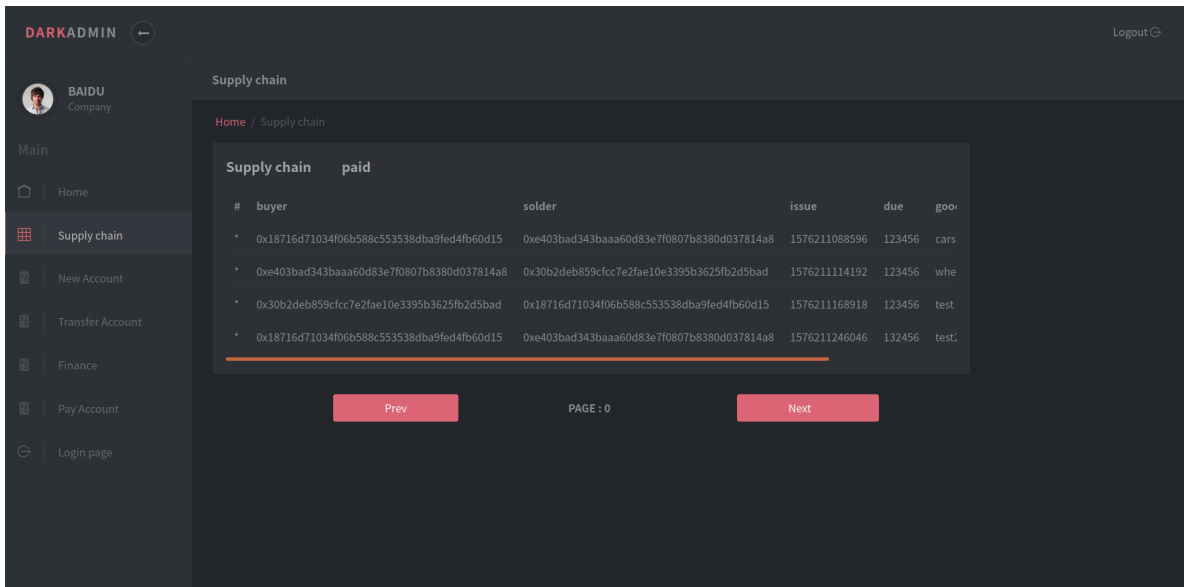
6、融资界面



7、支付账款界面



8、供应链信息界面



四、功能测试

1、注册

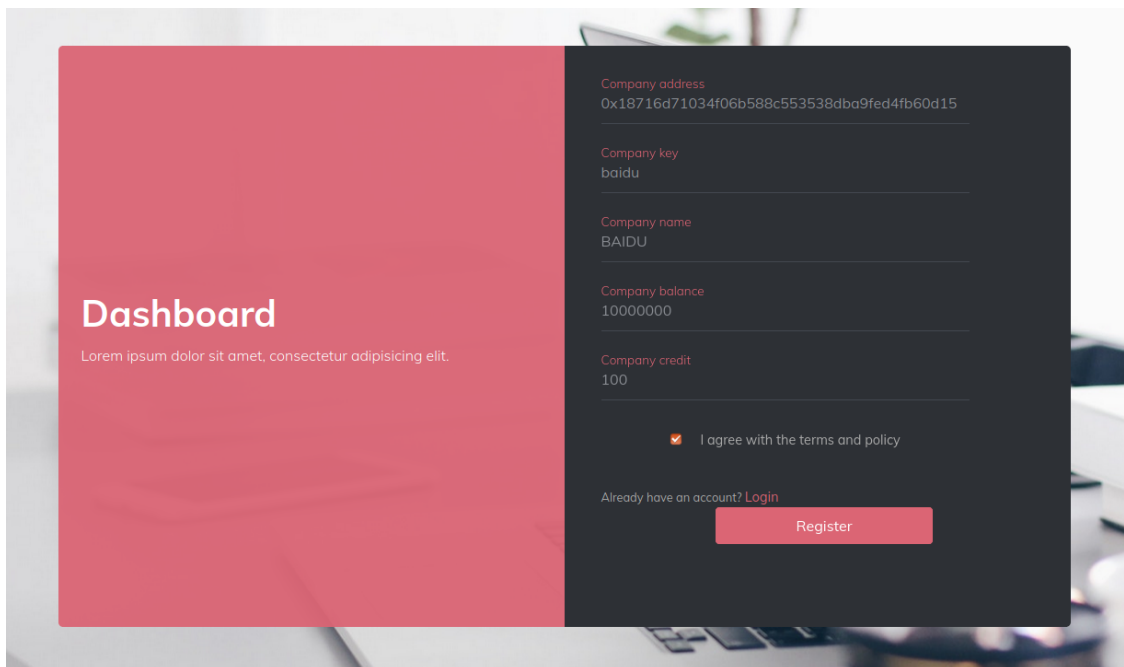
- 目标功能

未注册公司在供应链上进行注册

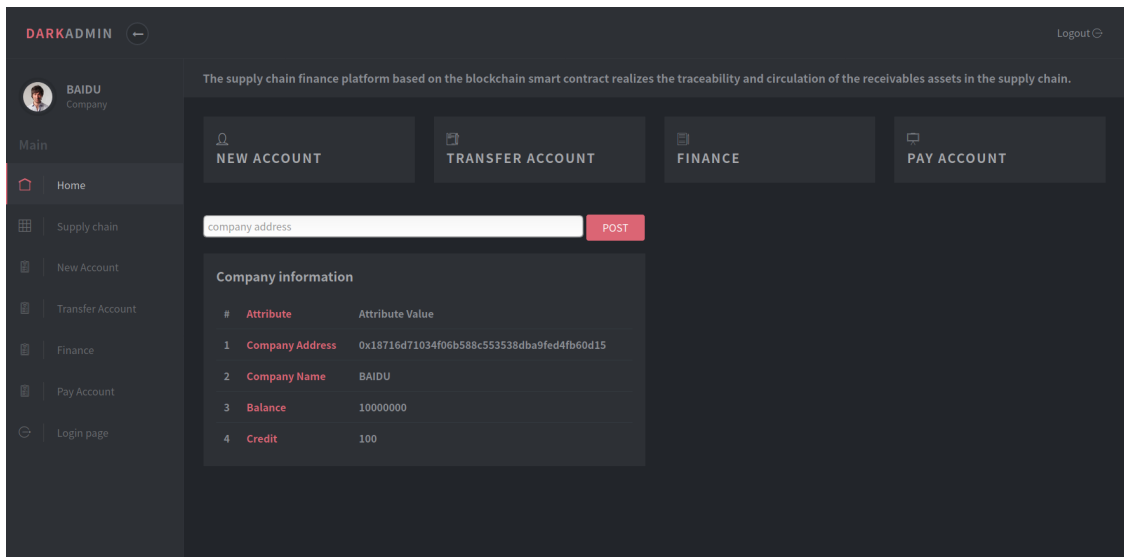
- 测试情景

公司A分别填写公司地址，登录密码，公司名称，公司资产，公司信用进行注册

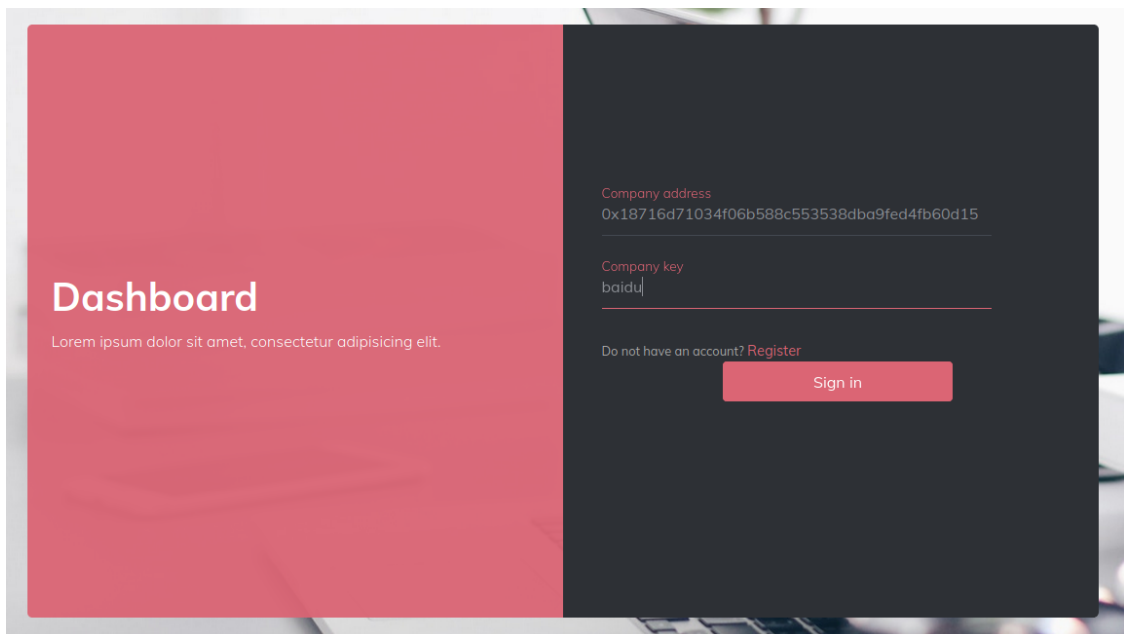
- 公司A填写注册表单进行注册



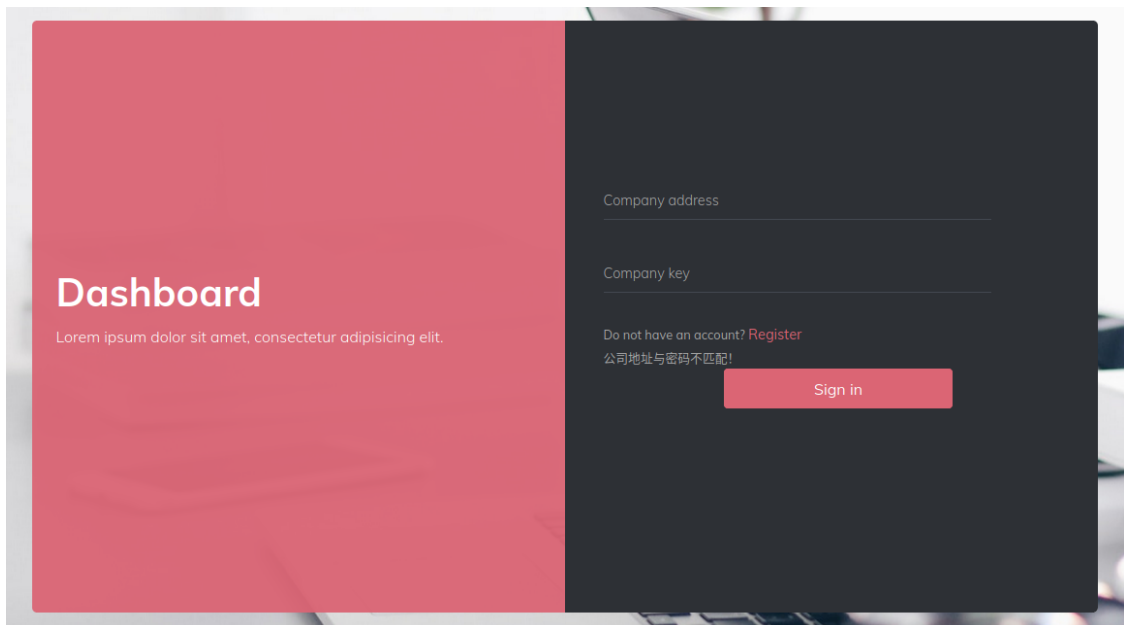
- 成功后自动跳转进入HOME界面



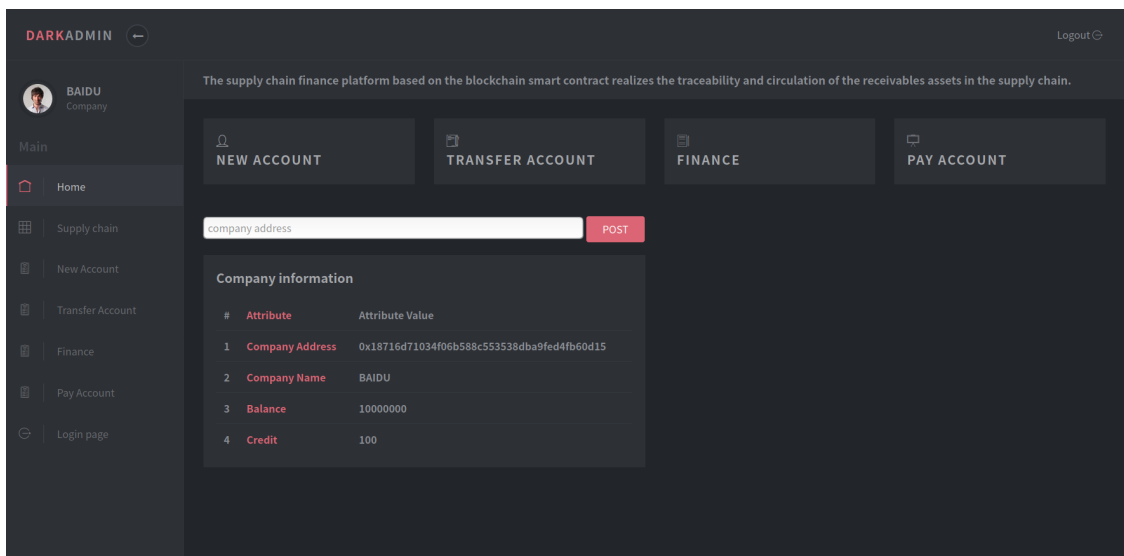
- 目标功能
 - 已注册公司通过公司地址和密码进行供应链应用登录
- 测试情景
 - 公司A分别填写公司地址，登录密码进行登录
- 公司A填写登录表单进行登录



- 地址与密码不匹配会显示错误信息



- 成功后自动跳转进入HOME界面



- 后续测试中需要使用到的公司、银行信息

地址	名称	资产	信用
0x18716d71034f06b588c553538dba9fed4fb60d15	BAIDU	10000000	100
0xe403bad343baaa60d83e7f0807b8380d037814a8	GOOGLE	30000000	200
0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad	WEBase	90000000	300
0x602c6fa929a24d28c6858824e34211e83f518911	BoC		

3、发布账款

- 目标功能

实现采购商品 — 签发应收账款 交易上链。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。

- 测试情景

公司A向公司B购买一批轮胎（tyre），总金额100000\$，截止日期为1576214777010

- 公司A填写并发布表单

New Account

[Home](#) / [New Account](#)

New Account

Realization of purchasing goods - issuing trade chain of accounts receivable.

buyer_addr

0x18716d71034f06b588c553538dba9fed4fb60d15

Example: 0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

solder_addr

0xe403bad343baaa60d83e7f0807b8380d037814a8

Example: 0x18716d71034f06b588c553538dba9fed4fb60d15

goods

tyre

Example: wheel

price

100000

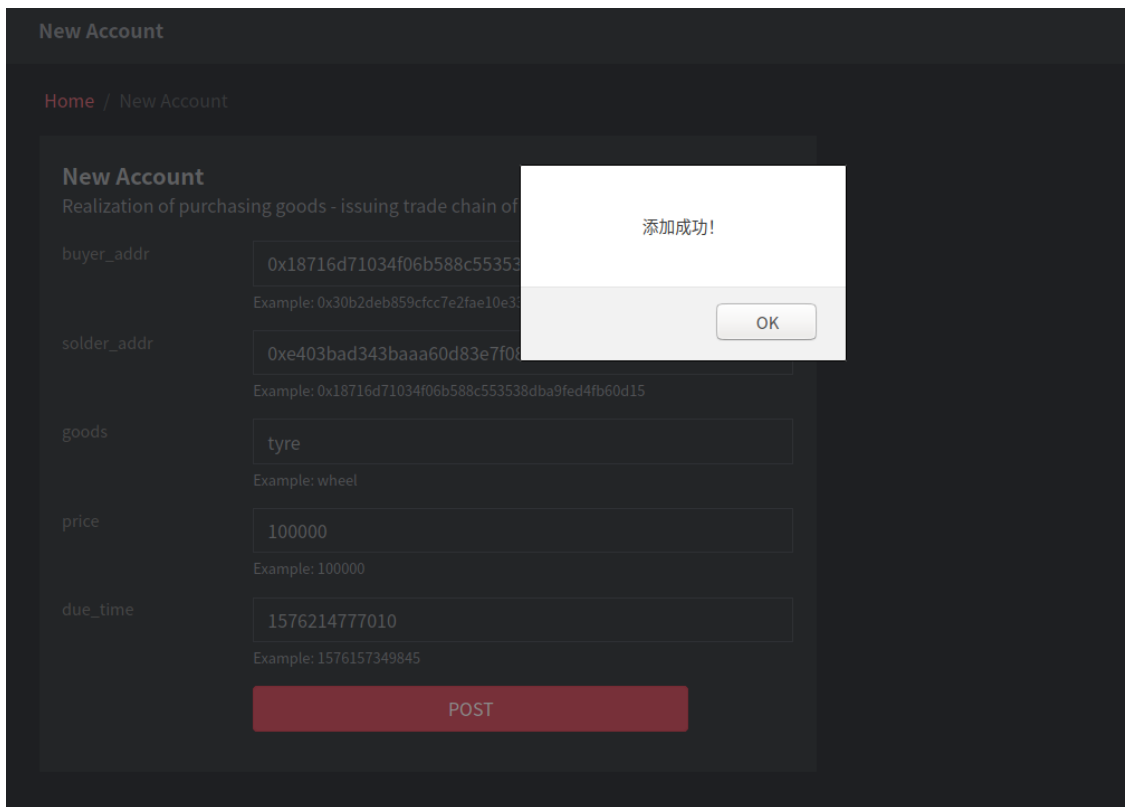
Example: 100000

due_time

1576214777010

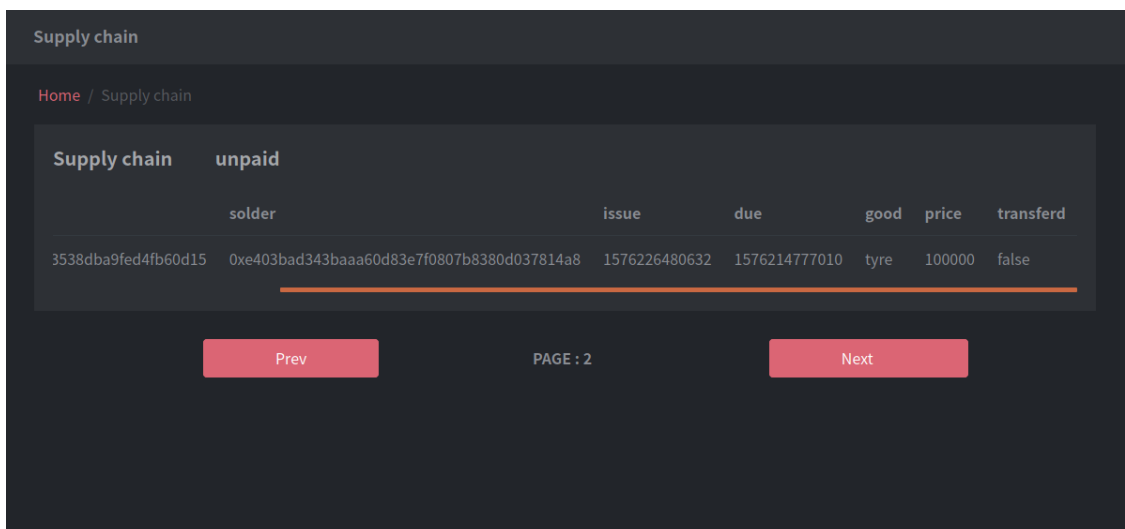
Example: 1576157349845

POST



- 成功后自动跳转，进入供应链界面

可以看到刚刚发布的账款信息已经被写入供应链上



4、账款转让

- 目标功能

实现应受账款的转让上链。比如轮胎公司从轮毂公司购买一笔轮毂，便将于车企的应收账款单据部分转让给轮毂公司。轮毂公司可以利用这个新的单据去融资或者要求车企到期时归还钱款。

- 测试情景

公司B使用公司A的账款向公司C购买一批轮胎（Hub），总金额50000\$，截止日期为1576214777100

- 公司B填写并发布表单

Transfer Account

[Home](#) / [Transfer Account](#)

Transfer Account

Realize the transfer chain of accounts receivable.

arrears party

0x18716d71034f06b588c553538dba9fed4fb60d15

Example: 0xe403bad343baaa60d83e7f0807b8380d037814a8

transferor

0xe403bad343baaa60d83e7f0807b8380d037814a8

Example: 0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

recipient

0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

Example: 0x18716d71034f06b588c553538dba9fed4fb60d15

goods

Hub

Example: wheel

price

50000

Example: 100000

due_time

1576214777100

Example: 1576066008070

POST

Transfer Account

[Home](#) / [Transfer Account](#)

Transfer Account

Realize the transfer chain of accounts receivable.

arrears party

0x18716d71034f06b588c553538dba9fed4fb60d15

Example: 0xe403bad343baaa60d83e7f0807b8380d037814a8

transferor

0xe403bad343baaa60d83e7f0807b8380d037814a8

Example: 0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

recipient

0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

Example: 0x18716d71034f06b588c553538dba9fed4fb60d15

goods

Hub

Example: wheel

price

50000

Example: 100000

due_time

1576214777100

Example: 1576066008070

POST

转让成功!

OK

- 成功后自动跳转，进入供应链界面

刚刚发布的转让账款信息已经被写入供应链上

Supply chain

Home / Supply chain

Supply chain

unpaid

	solder	issue	due	good	price	transferd
0807b8380d037814a8	0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad	1576227086498	1576214777100	Hub	50000	false
538dba9fed4fb60d15	0xe403bad343baaa60d83e7f0807b8380d037814a8	1576226480632	1576214777010	tyre	100000	true

Prev

PAGE : 2

Next

5、账款融资

- 目标功能

利用应收账款向银行融资上链，供应链上所有可以利用应收账款单据向银行申请融资。

- 测试情景

公司C使用从公司B转让过来的账款向银行（BoC）融资50000\$，截止日期为1576214777100

- 公司C填写并发布表单

Finance

Home / Finance

Finance

Using accounts receivable to finance the bank.

bank

0x602c6fa929a24d28c6858824e34211e83f518911

Example: 0x18716d71034f06b588c553538dba9fed4fb60d15

company

0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

Example: 0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

arrear party

0xe403bad343baaa60d83e7f0807b8380d037814a8

Example: 0xe403bad343baaa60d83e7f0807b8380d037814a8

money

50000

Example: wheel

due time

1576214777100

Example: 1576066008070

POST

Finance

Home / Finance

Finance

Using accounts receivable to finance the bank.

bank

0x602c6fa929a24d28c6858824

Example: 0x18716d71034f06b588c55353

company

0x30b2deb859cfcc7e2fae10e3

Example: 0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

arrear party

0xe403bad343baaa60d83e7f0807b8380d037814a8

Example: 0xe403bad343baaa60d83e7f0807b8380d037814a8

money

50000

Example: wheel

due time

1576214777100

Example: 1576066008070

POST

贷款成功!

OK

- 成功后自动跳转，进入HOME界面
此时公司C的账户上多了贷款的金额，说明贷款成功

The supply chain finance platform based on the blockchain smart contract realizes the traceability and

NEW ACCOUNT

TRANSFER ACCOUNT

FINANCE

company address

POST

Company information

#	Attribute	Attribute Value
1	Company Address	0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad
2	Company Name	WEBASE
3	Balance	90050000
4	Credit	300

- 查看供应链
现在这条供应链上的两条记录都被标记为了"transferd"，因为最后一个账款记录在设计逻辑里相当于被转让给了银行，这条账款不能够再被下游企业重复使用

Supply chain

Home / Supply chain

Supply chain

unpaid

	solder	issue	due	good	price	transferd
538dba9fed4fb60d15	0xe403bad343baaa60d83e7f0807b8380d037814a8	1576226480632	1576214777010	tyre	100000	true
0807b8380d037814a8	0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad	1576227086498	1576214777100	Hub	50000	true

Prev

PAGE : 2

Next

6、支付账款

- 目标功能

应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款。

- 测试情景

公司A向下游企业支付相应的金额

- 公司A填写并发布支付表单

Pay Account

Home / Pay Account

Pay Account

When the accounts receivable document is due, the core enterprise shall pay the corresponding debts to the downstream enterprise.

company

0x18716d71034f06b588c553538dba9fed4fb60d15

Example: 0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad

downstream enterprise

0xe403bad343baaa60d83e7f0807b8380d037814a8

Example: 0x18716d71034f06b588c553538dba9fed4fb60d15

POST

Pay Account

Home / Pay Account

Pay Account

When the accounts receivable document is due, the company pays the corresponding debts to the downstream enterprise.

company

0x18716d71034f06b588c553538dba9fed4fb60d15

Example: 0x30b2deb859cfcc7e2fae10e3...

downstream enterprise

0xe403bad343baaa60d83e7f0807b8380d037814a8

Example: 0x18716d71034f06b588c553538dba9fed4fb60d15

POST

支付成功!

OK

- 成功后自动跳转，进入HOME界面

- 查看公司A账户信息

公司A支付了账款的全部金额，总资产为 $10000000-100000=9900000$

POST

Company information

#	Attribute	Attribute Value
1	Company Address	0x18716d71034f06b588c553538dba9fed4fb60d15
2	Company Name	BAIDU
3	Balance	9900000
4	Credit	100

- 查看公司B账户信息

公司B获得了部分账款金额，总资产为 $30000000+50000=30050000$

POST

Company information

#	Attribute	Attribute Value
1	Company Address	0xe403bad343baaa60d83e7f0807b8380d037814a8
2	Company Name	GOOGLE
3	Balance	30050000
4	Credit	200

- 查看公司C账户信息

公司B获得了部分账款金额，并且有银行的贷款，总资产为 $90000000+100000=90100000$

Company information

#	Attribute	Attribute Value
1	Company Address	0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad
2	Company Name	WEBASE
3	Balance	90100000
4	Credit	300

可以看到三个公司的资产全部正确，支付操作成功

- 查看供应链

现在这条供应链上的属性变成了“paid”，说明已经被支付

Supply chain

Home / Supply chain

Supply chain	paid						
	solder	issue	due	good	price	transferd	
538dba9fed4fb60d15	0xe403bad343baaa60d83e7f0807b8380d037814a8	1576226480632	1576214777010	tyre	100000	true	
0807b8380d037814a8	0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad	1576227086498	1576214777100	Hub	50000	true	

PAGE : 2

五、项目加分项

功能

- 对链上企业信息的查询

```

1 function get_user(address addr)public view returns(string name, uint
  balance, uint credit){
2     return (companies[addr].name, balances[addr],
  companies[addr].credit);
3 }
```

Company information

#	Attribute	Attribute Value
1	Company Address	0x18716d71034f06b588c553538dba9fed4fb60d15
2	Company Name	BAIDU
3	Balance	9900000
4	Credit	100

- 查看链上供应链信息

```

1 //获取供应链
2 //获得供应链链表长度
3 function get_reciept_chain_len()public view returns (uint){
4     return reciept_chains.length;
5 }
6 //获得供应链长度
7 function get_reciepts_len(uint chain_index)public view returns (uint){
8     return reciept_chains[chain_index].reciepts.length;
9 }
10 //获得单个账款长度
11 function get_reciept(uint chain_index, uint reciept_index)public view
12 returns(address,address,uint,uint,string,uint,bool){
13     Receipt memory tmp =
14     reciept_chains[chain_index].reciepts[reciept_index];
15     return (tmp.buyer_addr, tmp.solder_addr, tmp.issure_time,
16     tmp.due_time, tmp.good, tmp.price, tmp.transferred);
17 }
18 //获得供应链支付状态
19 function get_chain_status(uint chain_index)public view returns (bool){
20     return reciept_chains[chain_index].paid;
21 }

```

Supply chain

Home / Supply chain

Supply chain	unpaid						
	solder	issue	due	good	price	transferd	
538dba9fed4fb60d15	0xe403bad343baaa60d83e7f0807b8380d037814a8	1576226480632	1576214777010	tyre	100000	true	
0807b8380d037814a8	0x30b2deb859cfcc7e2fae10e3395b3625fb2d5bad	1576227086498	1576214777100	Hub	50000	true	

Prev

PAGE : 2

Next

- 增加了供应链上的信用属性

企业包括信用属性，当进行账款操作时其他企业可以通过HOME界面的搜索框查询交易企业的信用值

Company information

#	Attribute	Attribute Value
1	Company Address	0x30b2deb859cfc7e2fae10e3395b3625fb2d5bad
2	Company Name	WEBASE
3	Balance	90100000
4	Credit	300

- 后端对用户信息进行加密

后端通过MD5算法对企业的密码进行了加密后再上链，保证了密码的安全性

```
1 function getMD5Password(content) {  
2     var md5 = crypto.createHash('md5');//md5加密方式;  
3     md5.update(content);  
4     var d = md5.digest('hex'); //加密后的值  
5     return d;  
6 }
```

友好的用户界面

- 在HOME界面可以通过搜索框查询企业信息，信息会直接同步到界面上的表格中

DARKADMIN

BAIDU
Company

Home

Supply chain

New Account

Transfer Account

Finance

Pay Account

Login page

The supply chain finance platform based on the blockchain smart contract realizes the traceability and circulation of the receivables assets in the supply chain.

NEW ACCOUNT

TRANSFER ACCOUNT

FINANCE

PAY ACCOUNT

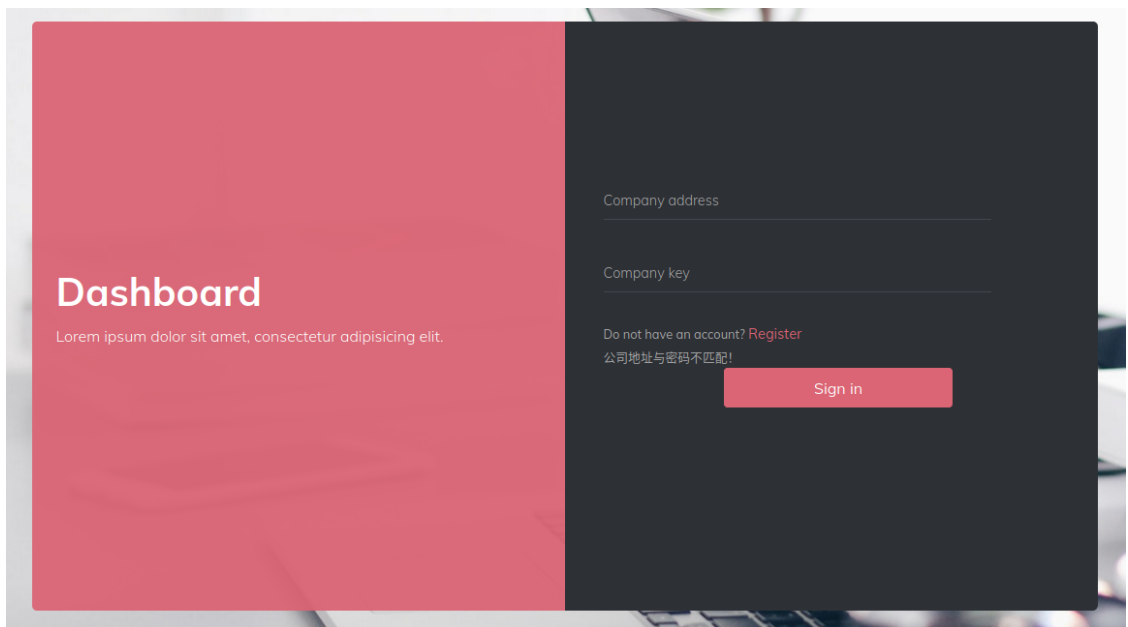
Company information

#	Attribute	Attribute Value
1	Company Address	0x18716d71034f06b588c553538dba9fed4fb60d15
2	Company Name	BAIDU
3	Balance	10000000
4	Credit	100

- 在用户提交表单后系统自动跳转到对应页面

具体详见主要功能展示部分，账款操作结束后系统会自动跳转到对应的HOME或供应链界面，便于用户检查操作结果

- 检测到错误时部分功能输出错误原因



六、 心得体会

本次大作业还是挺有难度的，毕竟链段、后端、前端都需要一个人完成，而且用到了区块链，nodejs，html的好几方面知识，大一时在web课上学到的东西终于有了拾起来的机会。通过这次作业我不仅了解到了区块链如何与后端进行交互，还学到了很多前后端的框架，比如express和vue的用法，着实获益匪浅。此外，由于需要实现与区块链交互的部分，所以我到部署的文件下读了很多源码，觉得对于fisco的了解比之前上课学到的理论印象要深刻许多，收获良多。