Building a Payment Splitter iOS Application Based on Blockchain

Yan-Da Chen / Po-Hsuan Chen / Che-Yu Chang
Ling-Yuan Chen / Wei-Ting Ho / Yan-Yu Huang

UCDAVIS
COMPUTER SCIENCE

Outline

- Introduction
- Implementation
  - Frontend Development
  - Backend Development
- Risks and Difficulties
- Future Work
- Demo

Introduction

- Payment splitting applications to keep track of money
- Split Ether
- Cryptocurrency payment splitter application

Introduction

- Payment splitting applications to keep track of money
- Split Ether
- Cryptocurrency payment splitter application

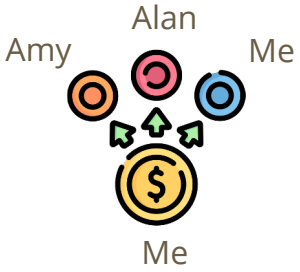
Introduction - Payment Splitter Flow

Create Participants

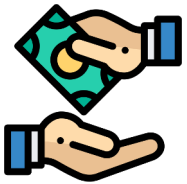
Set Agreement

Check Balance

Create Expense



Create Payment



Implementation

Front-end

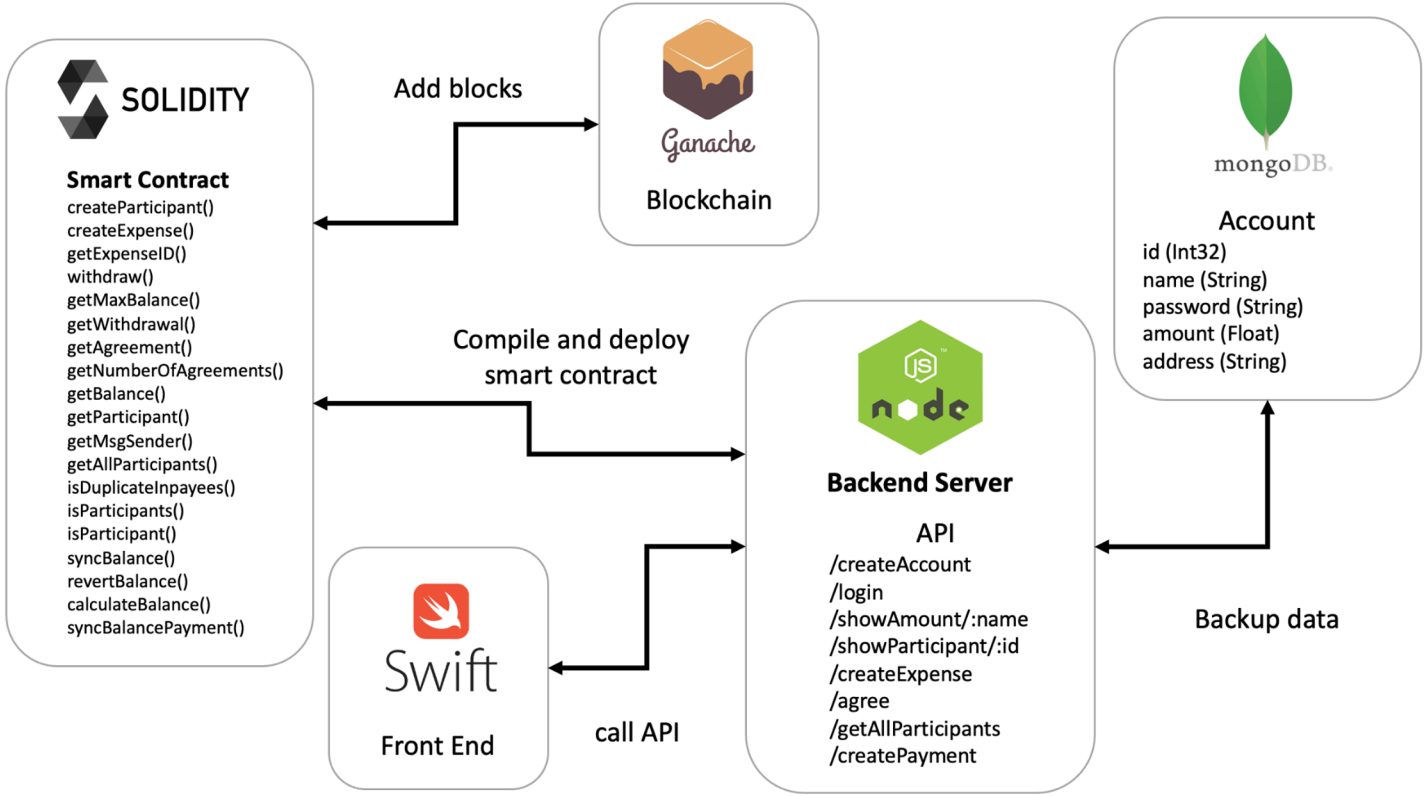
Back-end

Implementation - Blockchain

Smart Contract

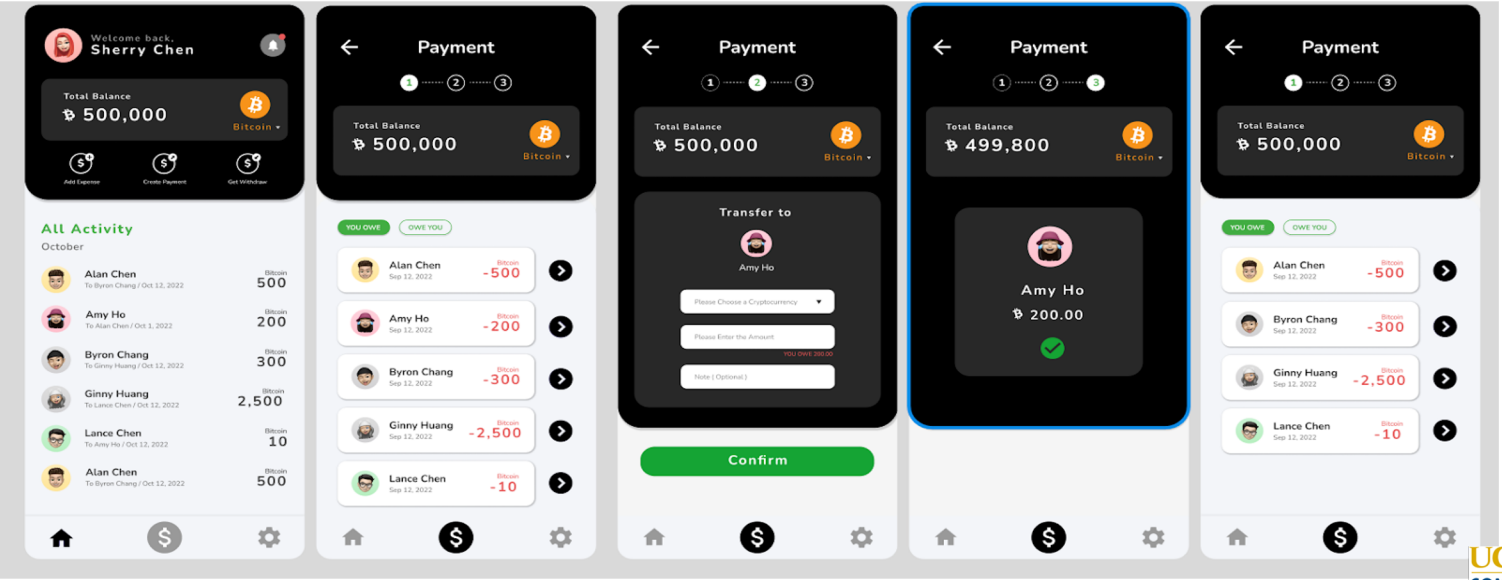
Development Tool

Flowchart


SOLIDITY
Smart Contract
createParticipant()
createExpense()
getExpenseID()
withdraw()
getMaxBalance()
getWithdrawal()
getAgreement()
getNumberOfAgreements()
getBalance()
getParticipant()
getMsgSender()
getAllParticipants()
isDuplicateInpayees()
isParticipants()
isParticipant()
syncBalance()
revertBalance()
calculateBalance()
syncBalancePayment()
Add blocks
Ganache
Blockchain
mongoDB.
Account
id (Int32)
name (String)
password (String)
amount (Float)
address (String)
Compile and deploy
smart contract
Backend Server
API
/createAccount
/login
/showAmount/:name
/showParticipant/:id
/createExpense
/agree
/getAllParticipants
/createPayment
Backup data
Swift
Front End
call API

Frontend Development

- Swift UI
  - User interface toolkit
- Swift
  - iOS develop language



Backend Development - API

API	REST Verb	Request Type	Request Structure	Response Structure	Request Description

/createAccount	POST	JSON	{
			"name": "alan",
			"password": "alan5411"
			}
				[
				{
				"info": "Create Account!"
				}
				]
					Register a new user and create the password.

/login	POST	JSON	{
			"name": "alan",
			"password": "alan5411"
			}
				{
				"info": "Login Successful!"
				}
				or
				{
				"info": "Wrong Password!"
				}
					Login a user and get the login status.

/showAmount/:name	GET		/showAccount/alan	[
				{
				"amount": 99.43462224
				}
				]
					Enter the name to get the amount of that account.

Backend Development - API

/getAllParticipants	GET			{ "info": "check all participants successful", "balanceInfo": [{ "name": "amy", "balance": 175000000000000000 }, { "name": "byron", "balance": -75000000000000000 }, { "name": "sherry", "balance": 125000000000000000 }, { "name": "alan", "balance": -75000000000000000 }, { "name": "ginny", "balance": -150000000000000000 }, {	Get all participants' names and balance.

Backend Development - API

/showParticipant/:id	GET		/showParticipant/1	[{ "Name": "alan", "Balance": 0, "Address": "0xE1a62eF686fe820DE76191e88fBa648A37dE696d" }]	Enter the id to get the name, balance, address from that account.
/createExpense	POST	JSON	{ "title": "food", "amount": 500000, "date": 2020, "payer": "Amy", "payee": "Sherry" }		Create an expense, including the title, amount, data, payer and payee.

Backend Development - API

/createPayment	POST	JSON	{
			"payer": "Amy",
			"payee": "Sherry",
			"amount": 500000
			}
				[
				{
				"sources": "200"
				}
				]
					Make a payment from one account to another account.

/agree	POST	JSON	{
			"name": "amy",
			"expenseID": 2
			}
				{
				"info": "amy agree expense 2"
				}
					Set agreement to expense.

Risks and Difficulties

Risks

- **High latency** of blockchain

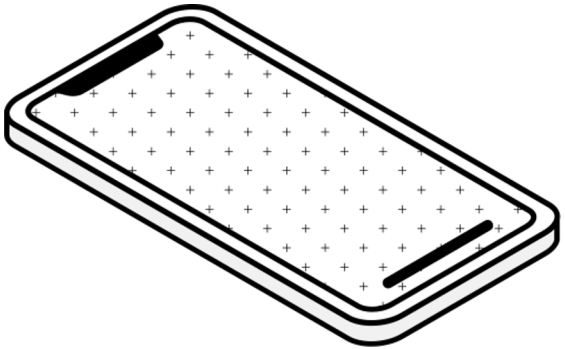
Difficulties

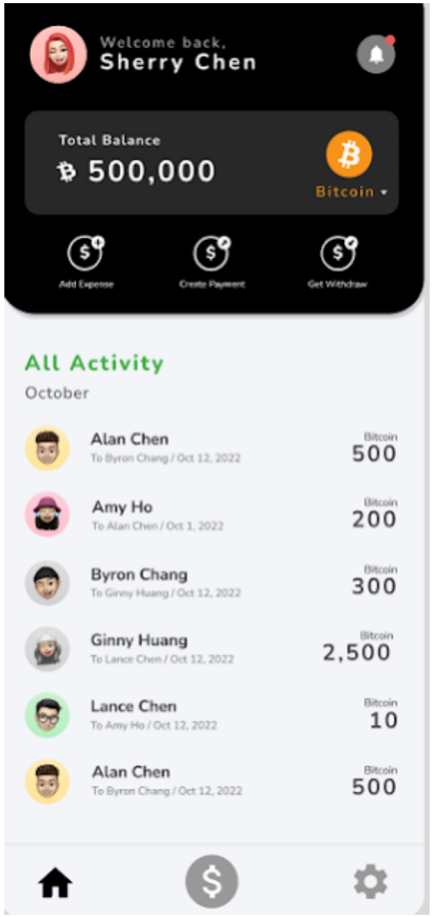
- New development environment
- Integration of backend and smart contract

Future Work

- **Add different splitting actions**
  - split by the **number of shares**
  - split by the **percentage**
  - share by **exact amounts**
- **Combine debts**
  - Ex. A owes B $10 and B owes C $10 = A owes C $10
  - **avoid extra payments**
  - **reduce transaction gas**

Demo




Welcome back,
Sherry Chen
Total Balance
฿ 500,000
Bitcoin
Add Expense
Create Payment
Get Withdraw
All Activity
October
Alan Chen
To Byron Chang / Oct 12, 2022
Bitcoin
500
Amy Ho
To Alan Chen / Oct 1, 2022
Bitcoin
200
Byron Chang
To Ginny Huang / Oct 12, 2022
Bitcoin
300
Ginny Huang
To Lance Chen / Oct 12, 2022
Bitcoin
2,500
Lance Chen
To Amy Ho / Oct 12, 2022
Bitcoin
10
Alan Chen
To Byron Chang / Oct 12, 2022
Bitcoin
500

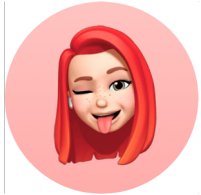
Team Work

Yan-Yu Huang

- User interface design
- Front end development

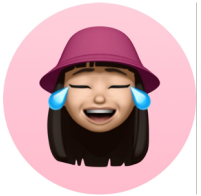
Yan-Da Chen

- Smart contract test
- Node.js API test



Po-Hsuan Chen

- Smart contract design
- Node.js API test



Wei-Ting Ho

- Smart contract test
- Node.js API design

Che-Yu Chang

- Front back end integration
- Node.js API design

Ling-Yuan Chen

- Front back end integration
- Smart contract design

UCDAVIS COMPUTER SCIENCE

Thank You