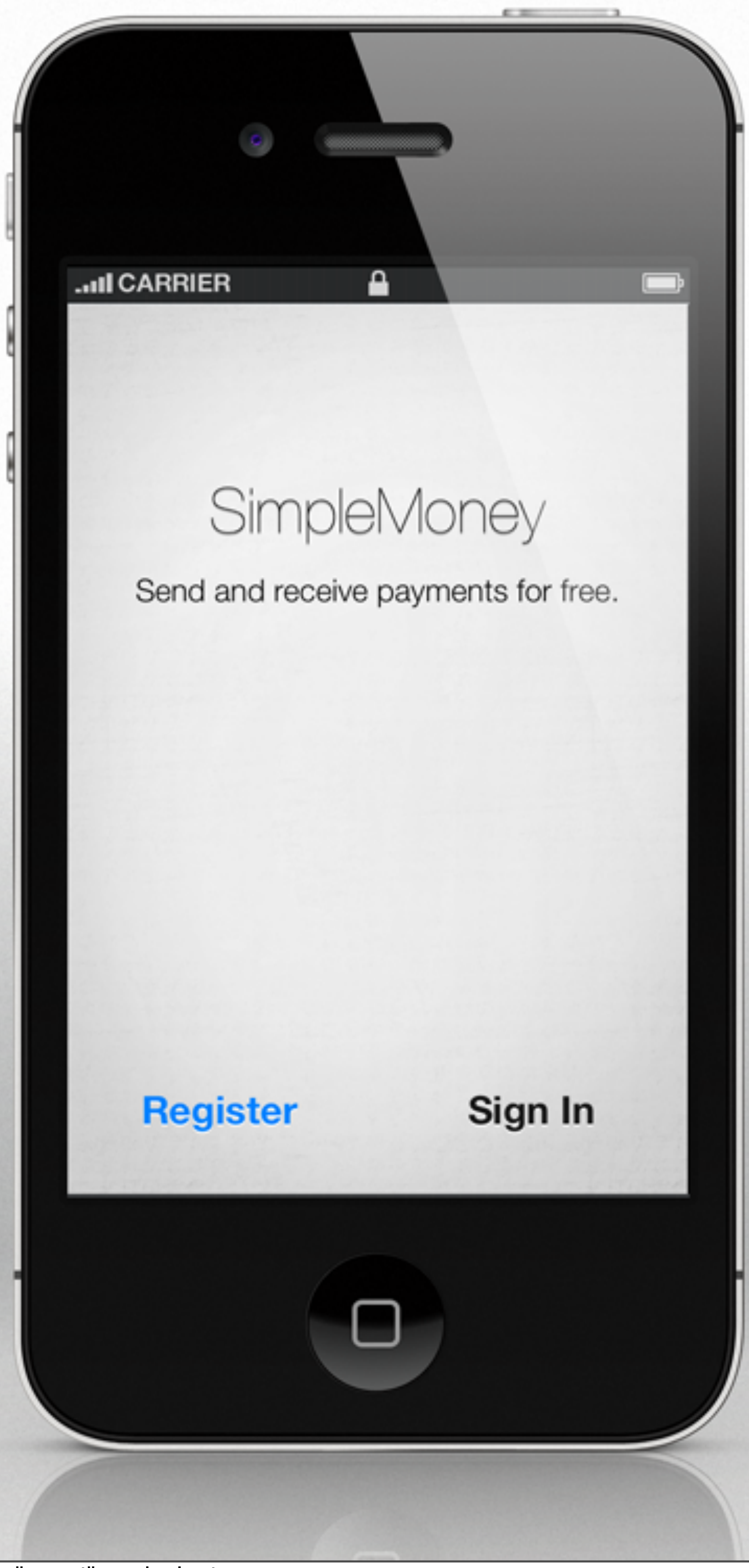
Making Money Simple

Client: Joshua Cross

Team Saon
Arthur Pang - Joshua Conner - Nicholas Pallares
April 5 2012



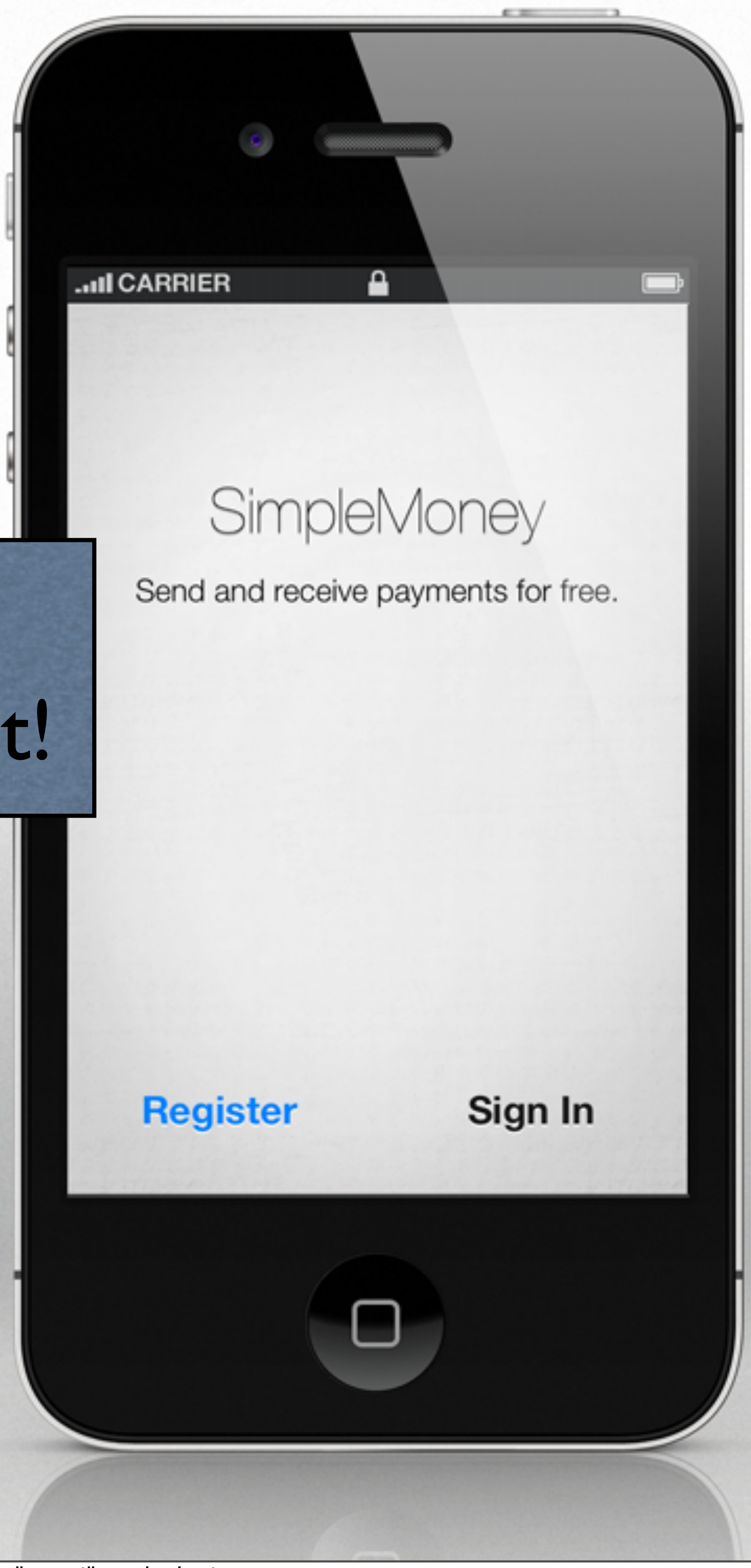
Smartphones: revolutionary because they replace "dumb" information sources with context-aware, or "smart" equivalents...

Making Money Simple

and smart!

Client: Joshua Cross

Team Saon
Arthur Pang - Joshua Conner - Nicholas Pallares
April 5 2012



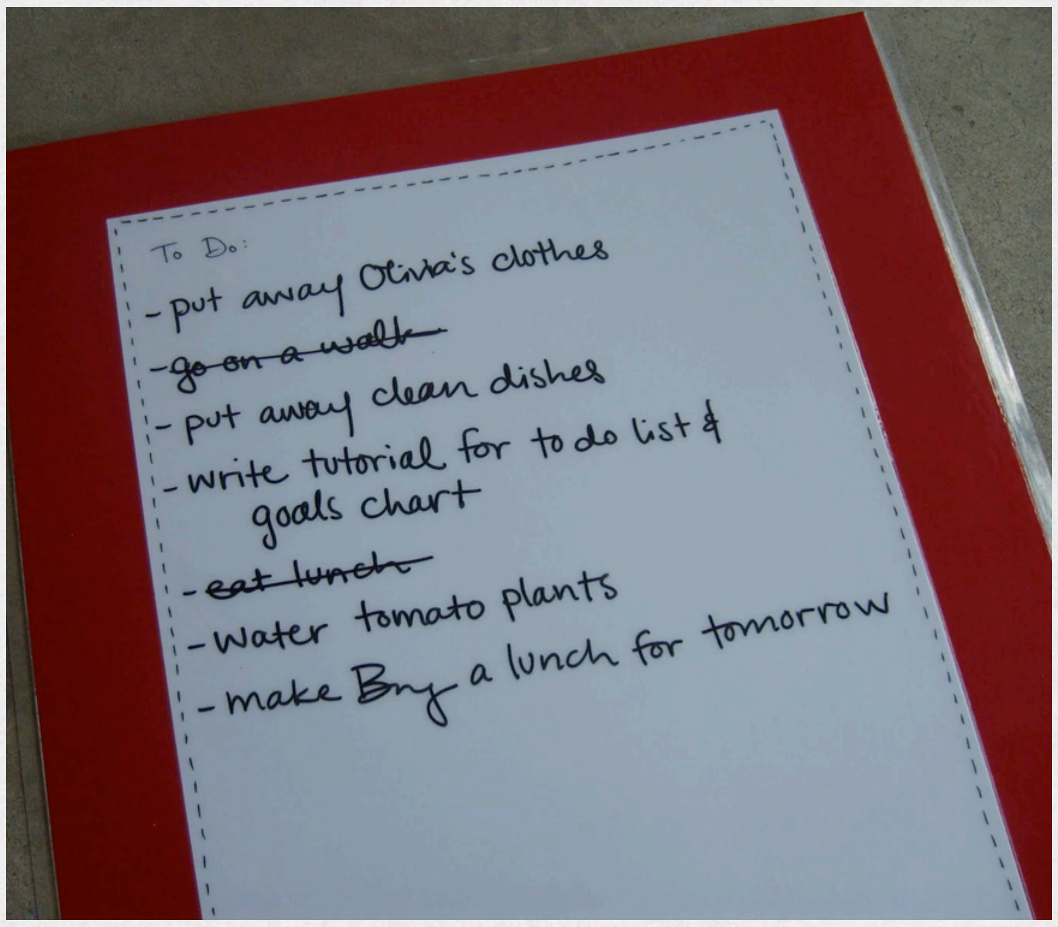
Smartphones: revolutionary because they replace "dumb" information sources with context-aware, or "smart" equivalents...

Smart maps

vs.

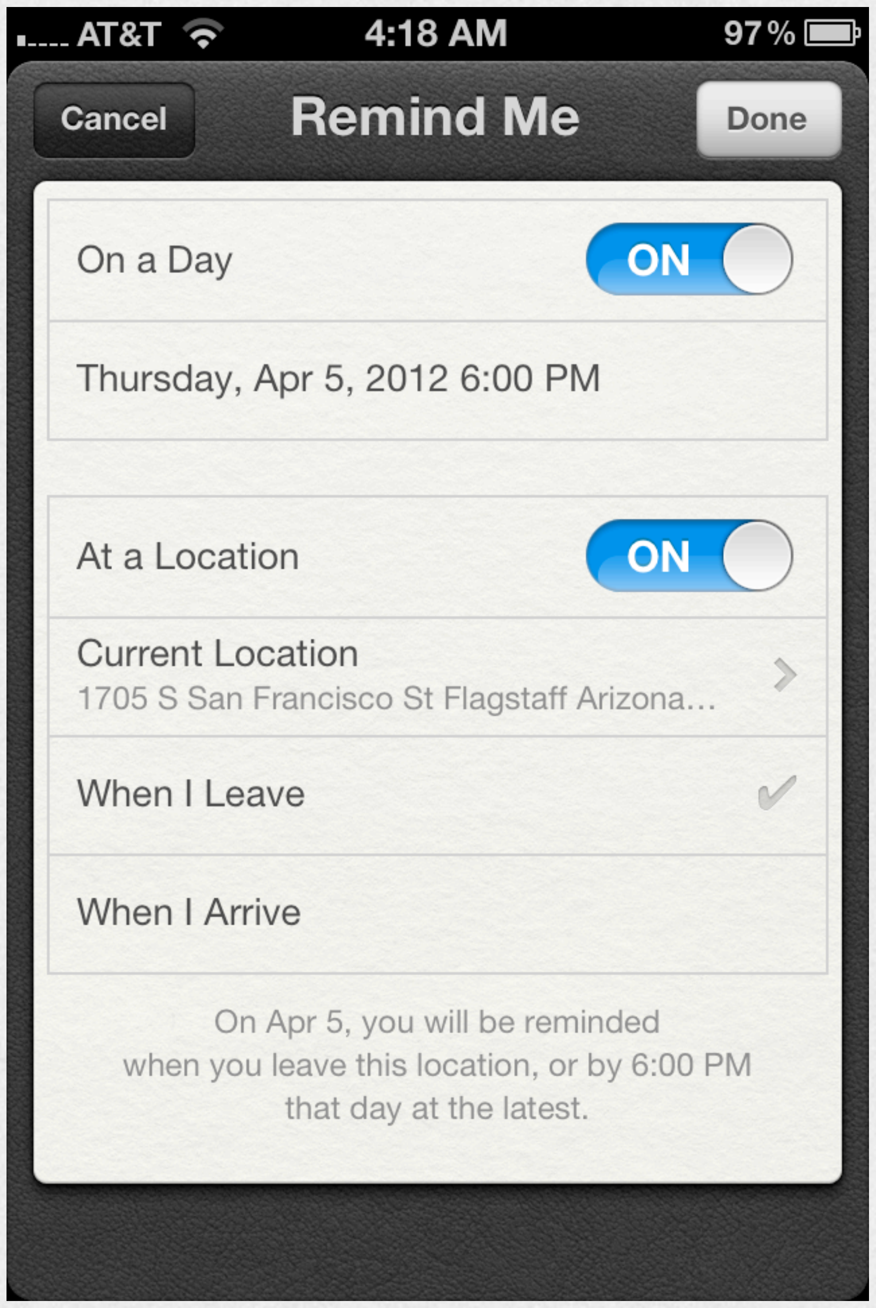
Dumb map vs. iPhone or Android "Maps" app
- uses GPS chip to figure out where you are and give you turn-by-turn directions
- directions take into account the traffic on the roads you'd take to give you the fastest route at that exact moment in time.
- don't need to know an address at all! can type in “target” to get nearest “Target” store

Smart lists



To Do:
- put away Olivia's clothes
- ~~go on a walk~~
- put away clean dishes
- write tutorial for to do list & goals chart
- ~~eat lunch~~
- water tomato plants
- make ~~Bry~~ a lunch for tomorrow

vs.



AT&T 4:18 AM 97 %

Cancel **Remind Me** Done

On a Day: ON

Thursday, Apr 5, 2012 6:00 PM

At a Location: ON

Current Location
1705 S San Francisco St Flagstaff Arizona...

When I Leave ✓

When I Arrive

On Apr 5, you will be reminded when you leave this location, or by 6:00 PM that day at the latest.

Todo list vs. iPhone “Reminders” app”
- can not only set off reminder at particular time
- but at particular PLACE

Smart cards?

- "Dumb" - no context
- Fees and interest
- Tied to hardware: useless if lost

- CC's are "dumb" – can't even simple things like checking balance from card
- Merchants paid $48 billion in swipe fees in 2011
- Losing CC on vacation

There's an app for that?

Credit card-based:

- Square, Paypal Here
- Google Wallet

Square: Fees! Still CC based
 - Great for small merchants who wouldn't otherwise be able to accept CC's
 - but little value-added for consumers

Google Wallet: blocked by Verizon, who is only carrier of only phone that can use GW.

There's an app for that?

Smartphone-based:

- "Vanilla" Paypal
- Dwolla

These are closer:
- Less fees if using ACH
- Not hardware-based: can use from any smartphone
- BUT can't do consumer-to-merchant payments

We can do better

(requirements)

REQUIREMENTS! say “requirements”!

We can do better
(requirements)

- Painless peer-to-peer AND merchant payments

REQUIREMENTS! say “requirements”!

We can do better
(requirements)

- Painless peer-to-peer AND merchant payments
- View balance and transaction history

REQUIREMENTS! say “requirements”!

We can do better
(requirements)

- Painless peer-to-peer AND merchant payments
- View balance and transaction history
- Crazy fast and easy: scan a QR code, pay in seconds

  (or peer-to-peer pay w/Address Book integration)

REQUIREMENTS! say “requirements”!

We can do better
(requirements)

- Painless peer-to-peer AND merchant payments
- View balance and transaction history
- Crazy fast and easy: scan a QR code, pay in seconds

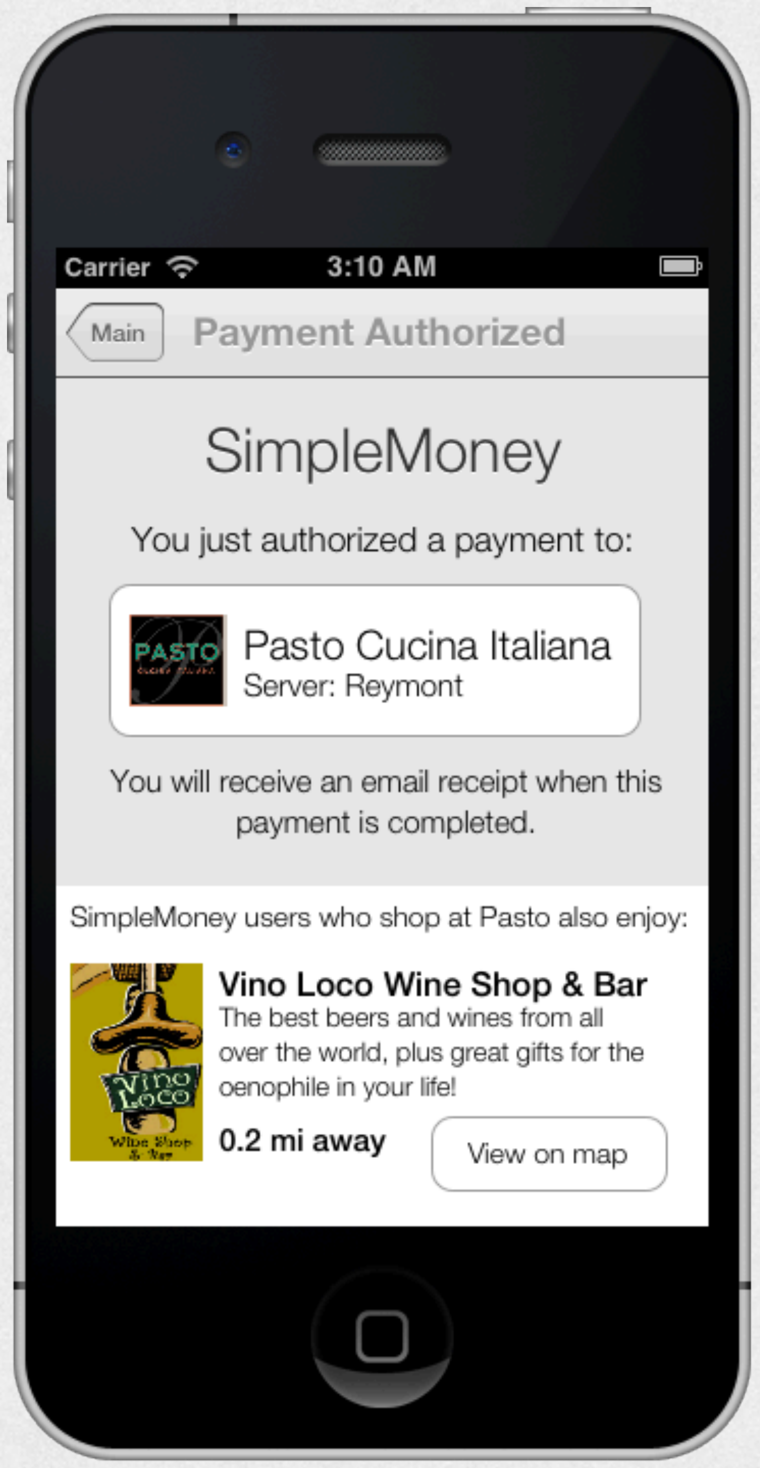
  (or peer-to-peer pay w/Address Book integration)

- Be the “smartest” smart money app

REQUIREMENTS! say “requirements”!

Recommendations

- Great value-add for merchants AND consumers
- Location-aware: encourages users to “shop local”


Carrier 3:10 AM

Main Payment Authorized

SimpleMoney

You just authorized a payment to:

Pasto Cucina Italiana
Server: Reymont

You will receive an email receipt when this payment is completed.

SimpleMoney users who shop at Pasto also enjoy:

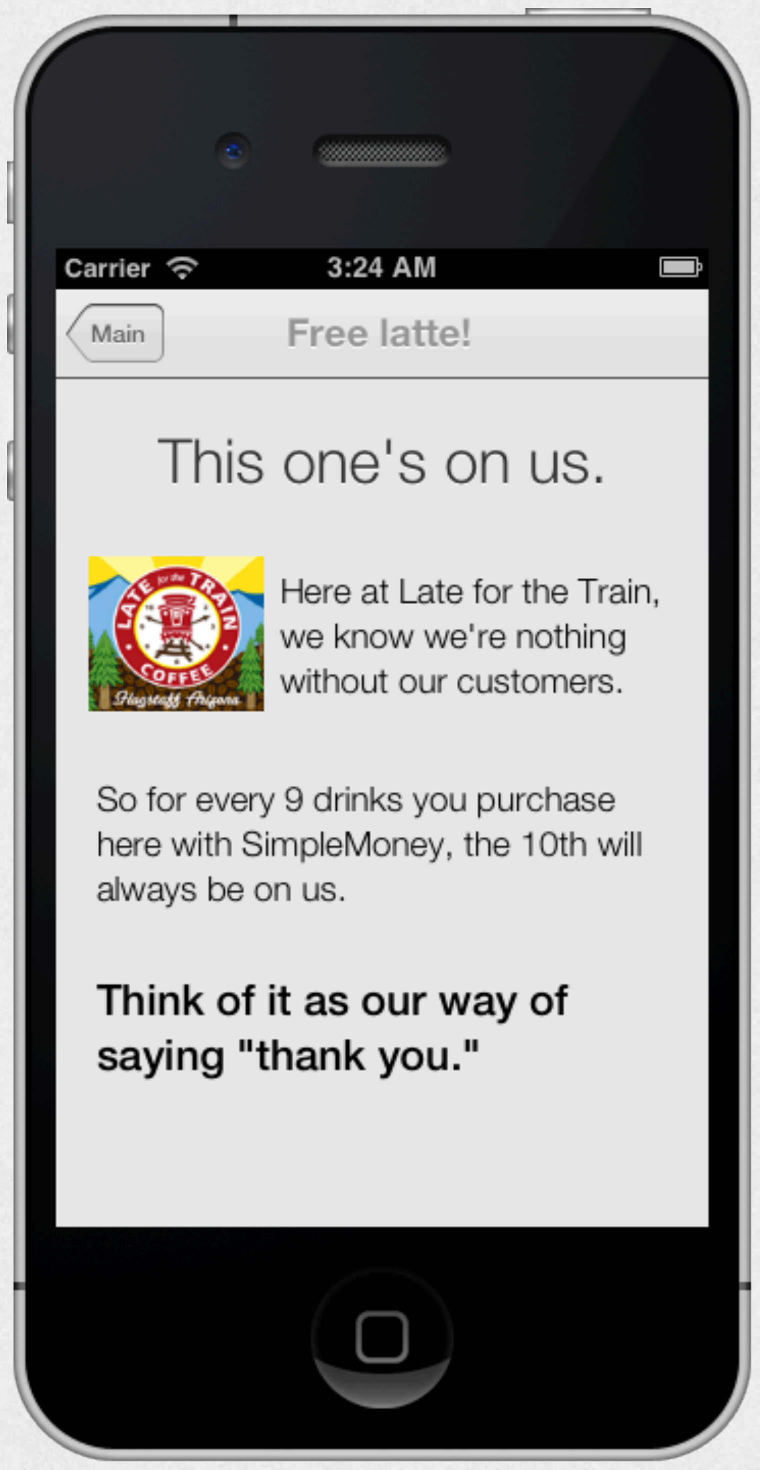
Vino Loco Wine Shop & Bar
The best beers and wines from all over the world, plus great gifts for the oenophile in your life!

0.2 mi away

- big money in online shopping
- location aware: shows distance, and has “view on map” button

Loyalty Programs

- Normally require expensive POS or tracking systems
- Encourages user adoption and customer loyalty



This one's on us.

Here at Late for the Train, we know we're nothing without our customers.

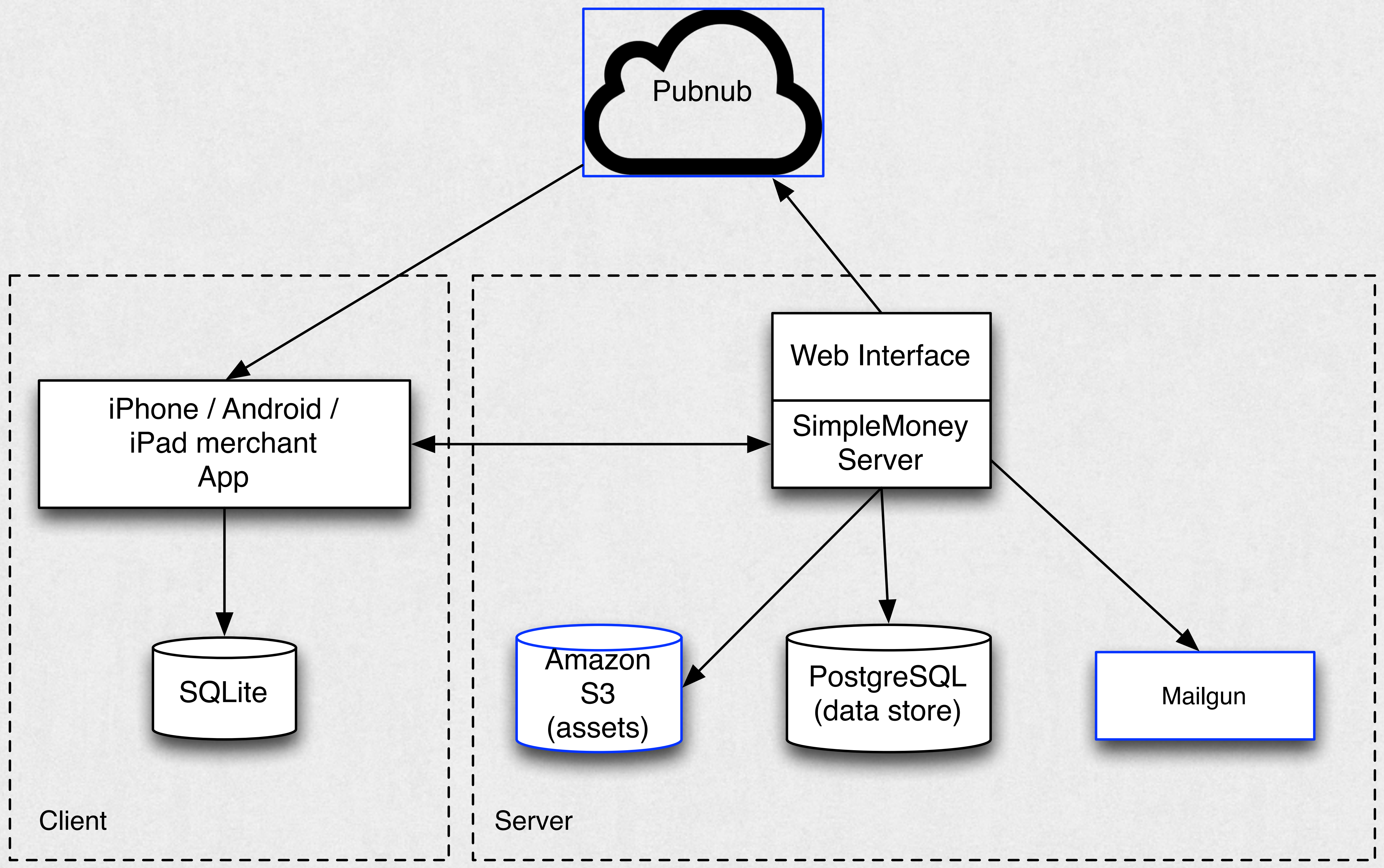
So for every 9 drinks you purchase here with SimpleMoney, the 10th will always be on us.

Think of it as our way of saying "thank you."

instead of carrying around punch card, what if it were automatic?

We think we've got a great concept, and some great architecture to back it up. Arthur's going to tell you all about it.

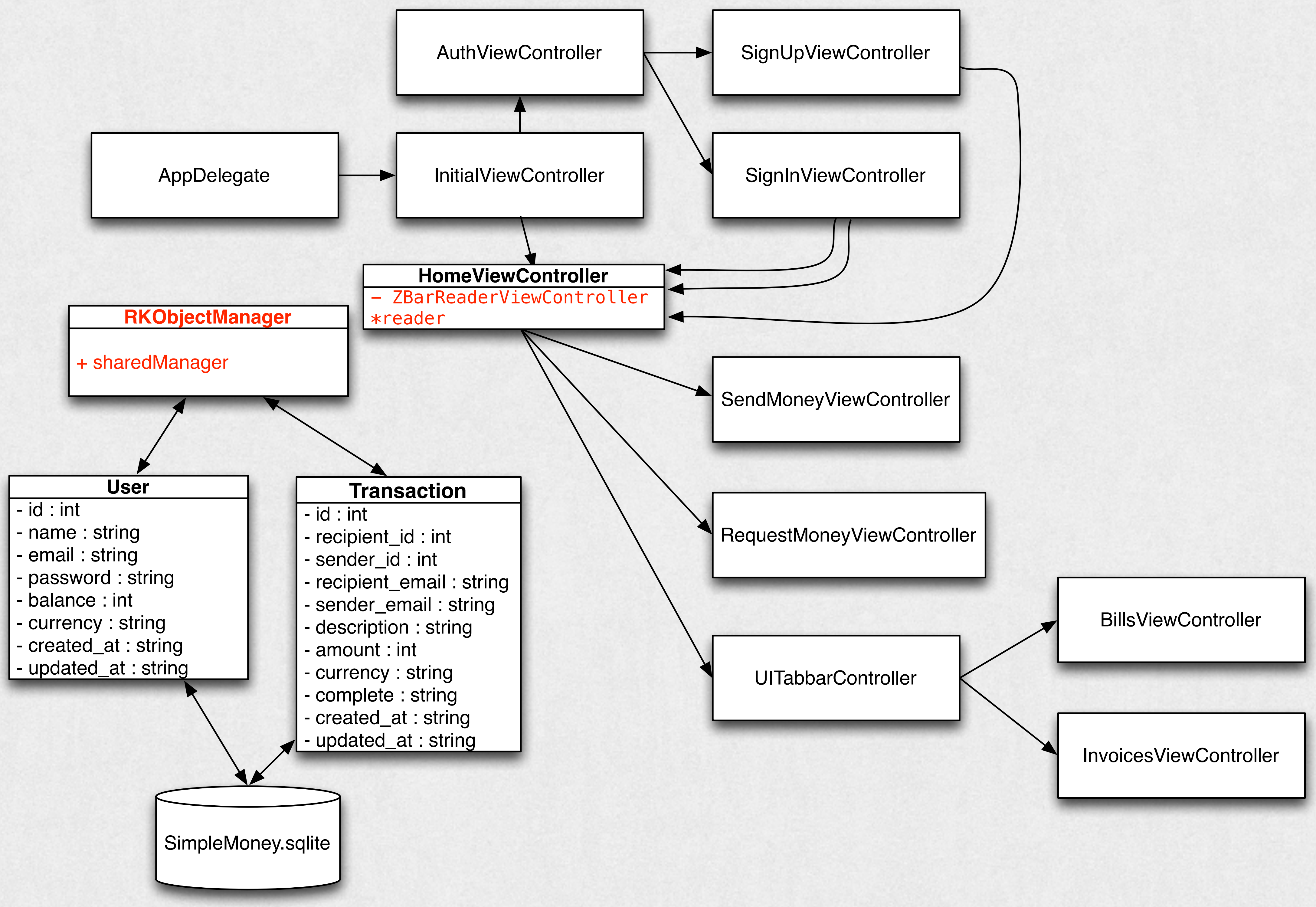
## Architecture



# Architecture

Here's a high level overview of our system.
- The iPhone and Android apps fetch data through a REST API for users, sessions and transactions, and replicate that data in a local SQLite database so transactions can be viewed offline.
- The server is built on Ruby on Rails, a great open source framework that allows us to iterate quickly.
- We are using several web services such as Amazon S3, Pubnub, and Mailgun.
- Amazon S3 is used to store our image assets, such as User avatars,
- Pubnub is used to send push notifications between our server and client applications when transactions are posted or updated so our users have up-to-date data.
- Lastly Mailgun is a service we use to send out confirmation emails or receipts.

# iPhone Architecture



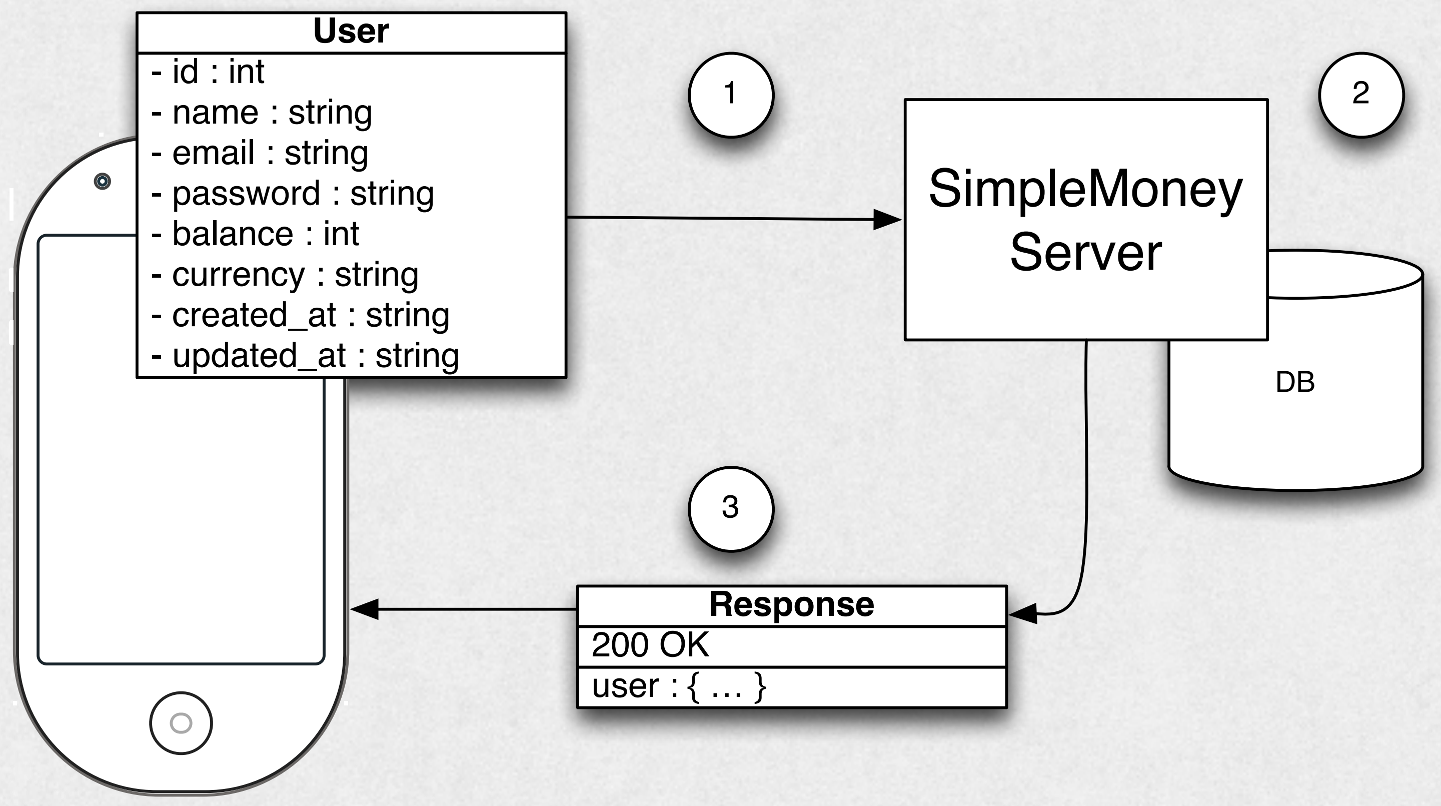
# iPhone Architecture

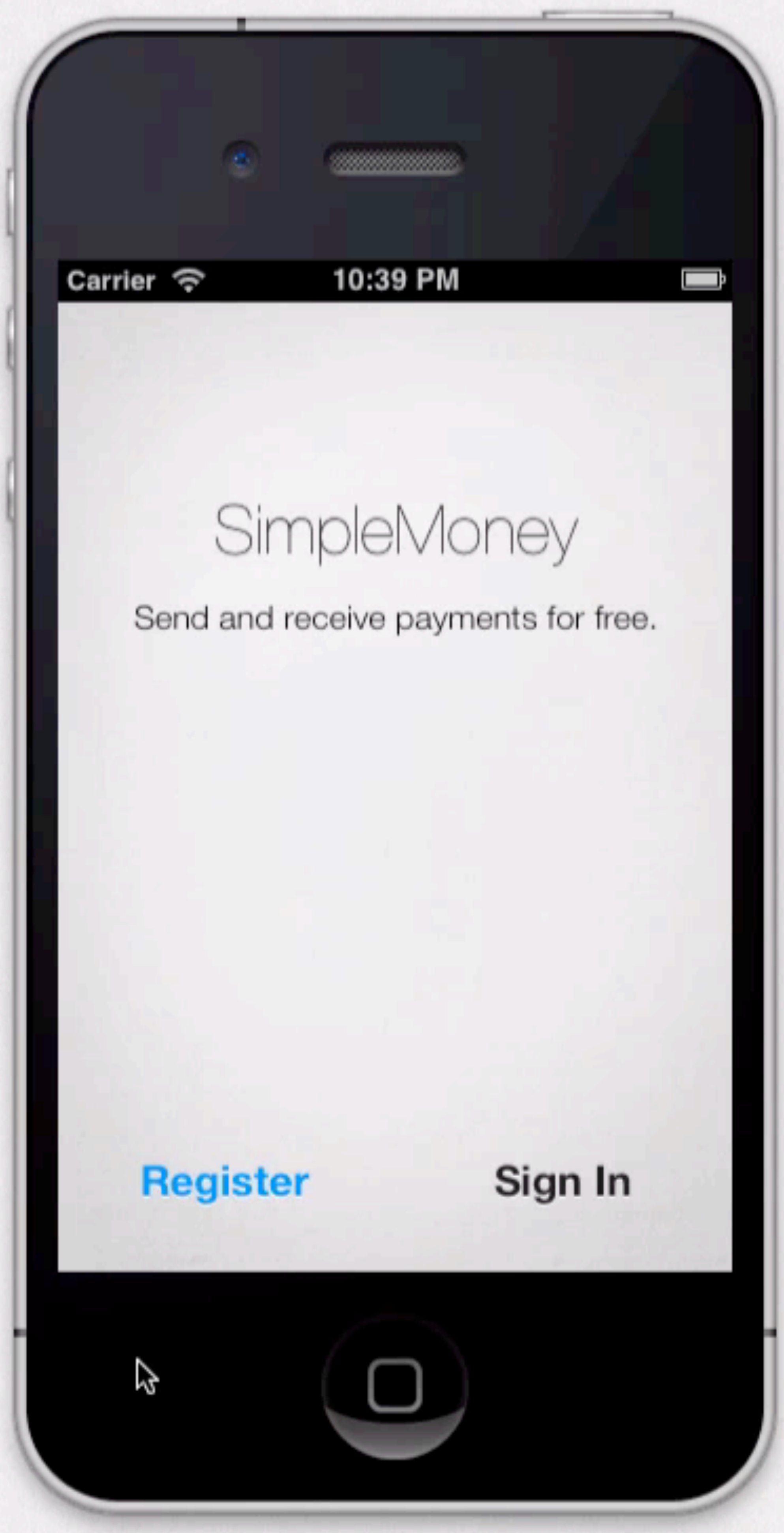
- Let's drill down into the iOS client application. We're using the iOS 5.1 SDK and two third party libraries marked in red: RESTKit and ZBar.
- RESTKit is a framework that allows us to interact with our server through a request and response API, and it maps our JSON objects from our server, to objective-c objects that can be stored and managed in a local SQLite database.
- ZBar is a small library that reads QR Codes.

- The app is quite simple, the initialViewController checks the iOS keychain for an existing username and password, if we have existing credentials we use RKObjectManager to send a POST request to our server's sessions resource. Otherwise, we allow the user to sign up or sign in.

- I won't walk you through the entire app here. Instead we will walk though the views that we have implemented to help illustrate how the app works in more detail.

## Sign Up





# Sign Up
- Let's look at the sign up view.
- First we populate the necessary parameters such as the email address and password, along with any optional ones like a user avatar. Users can take a photo with their camera, or choose an existing one from their library.
- When we're done filling out the form, we send a POST request to our server's USER resource. Our server will validate the format of the email address and also validate that the email address is unique. If the user model validates and saves to the database, the server sends a 200 response along with a JSON representation of the newly created user.
- Once our client app receives the response and user object, we display a successful confirmation dialog, and push the homeViewController.

Home Screen

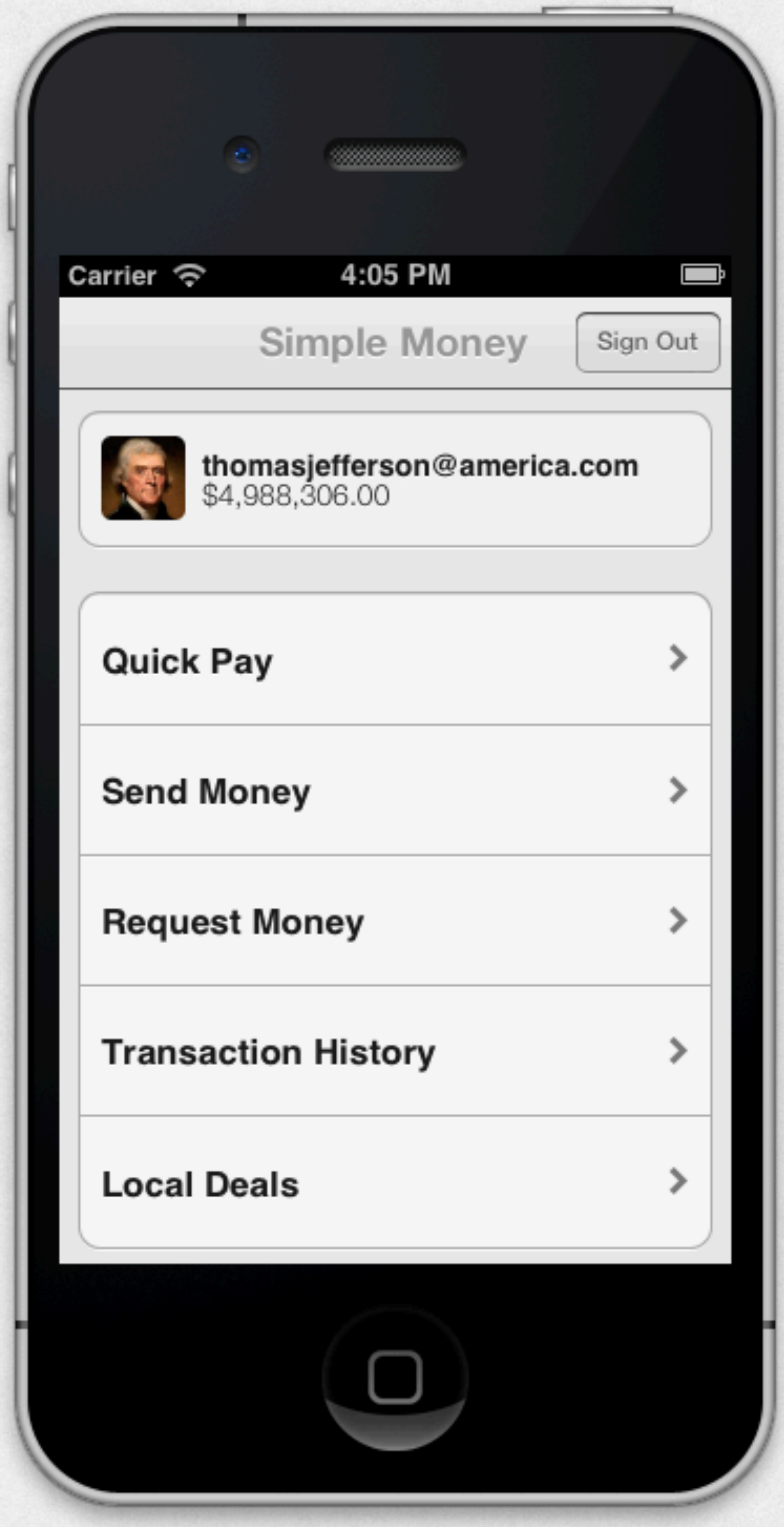
View account balance

Pay by scanning a QR code

Send and request money

View transactions

View local deals


Carrier 4:05 PM

Simple Money

Sign Out

thomasjefferson@america.com
$4,988,306.00

Quick Pay

Send Money

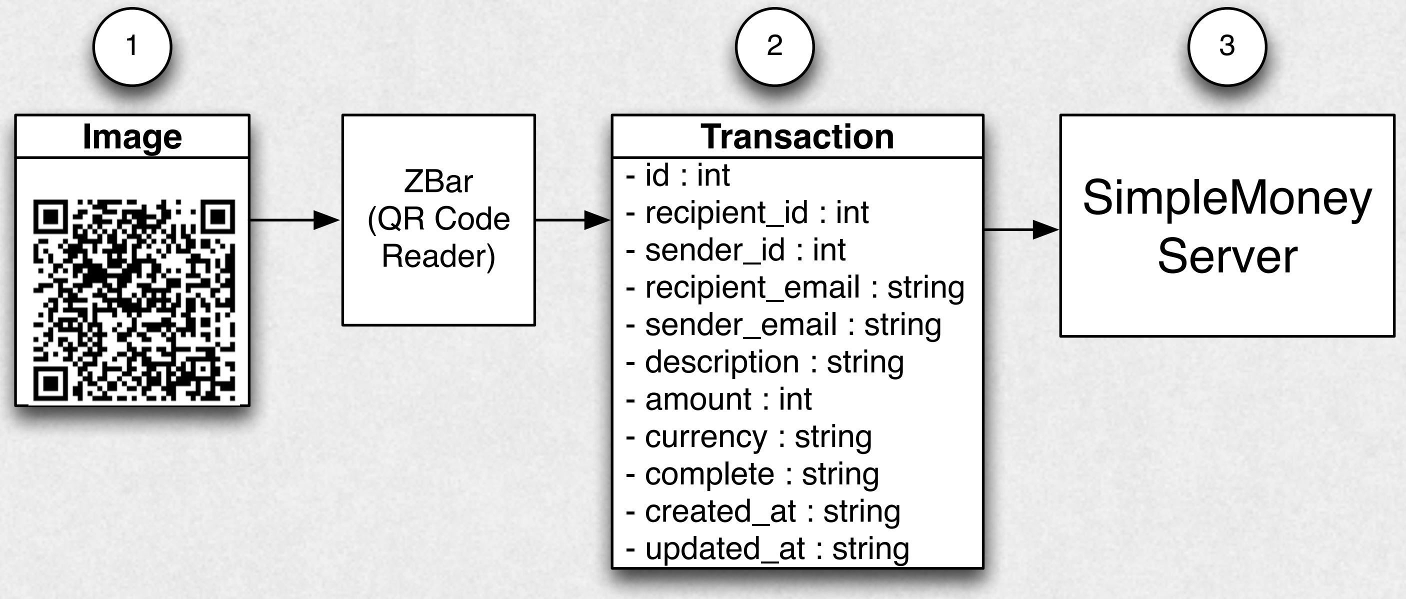
Request Money

Transaction History

Local Deals

Once signed in, a user can pay for a transaction by scanning a QR code, send and request money from other users, view transaction history, or view local deals.

## Quick Pay



1. QR Code is scanned

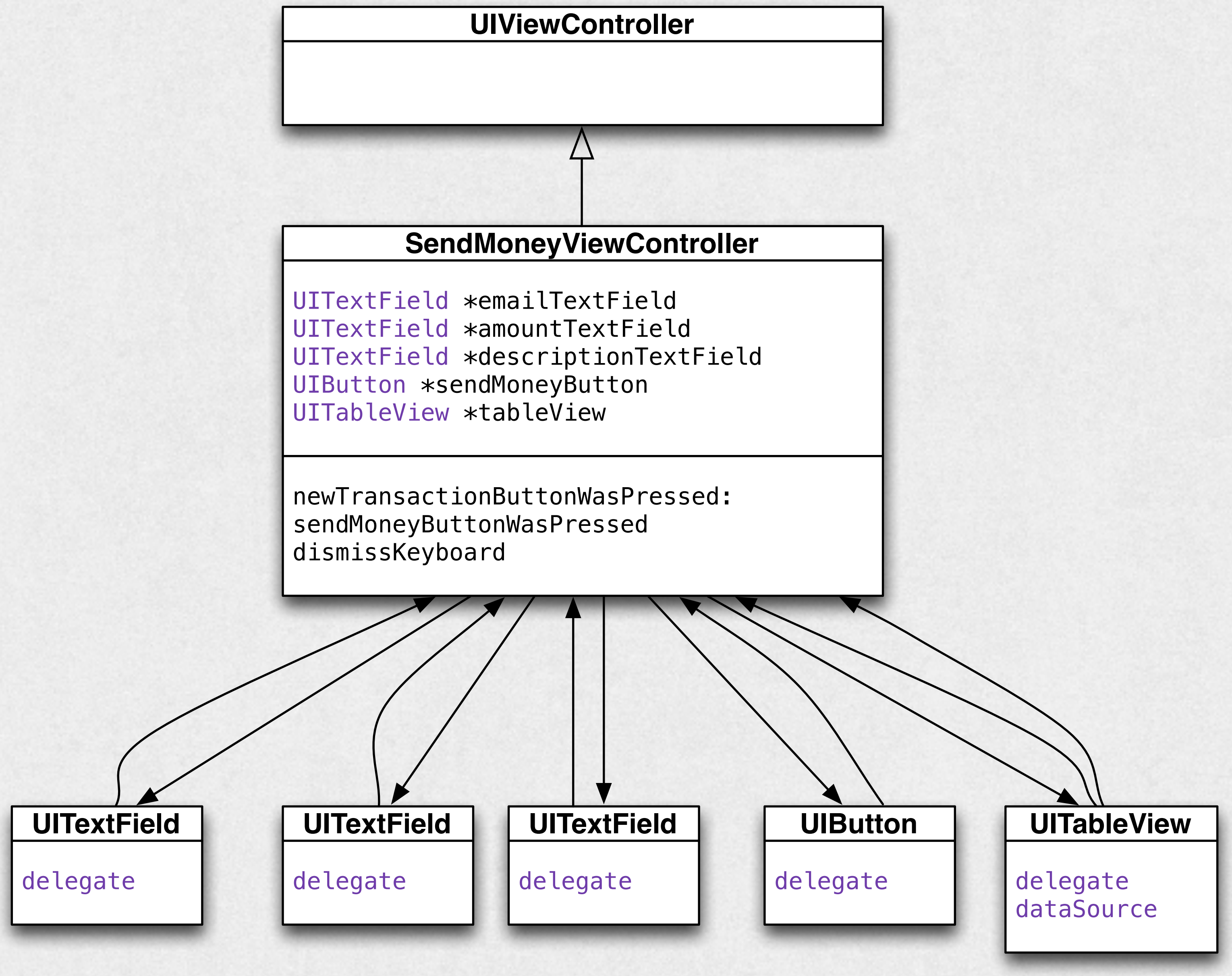
2. App grabs the merchant id and creates a transaction

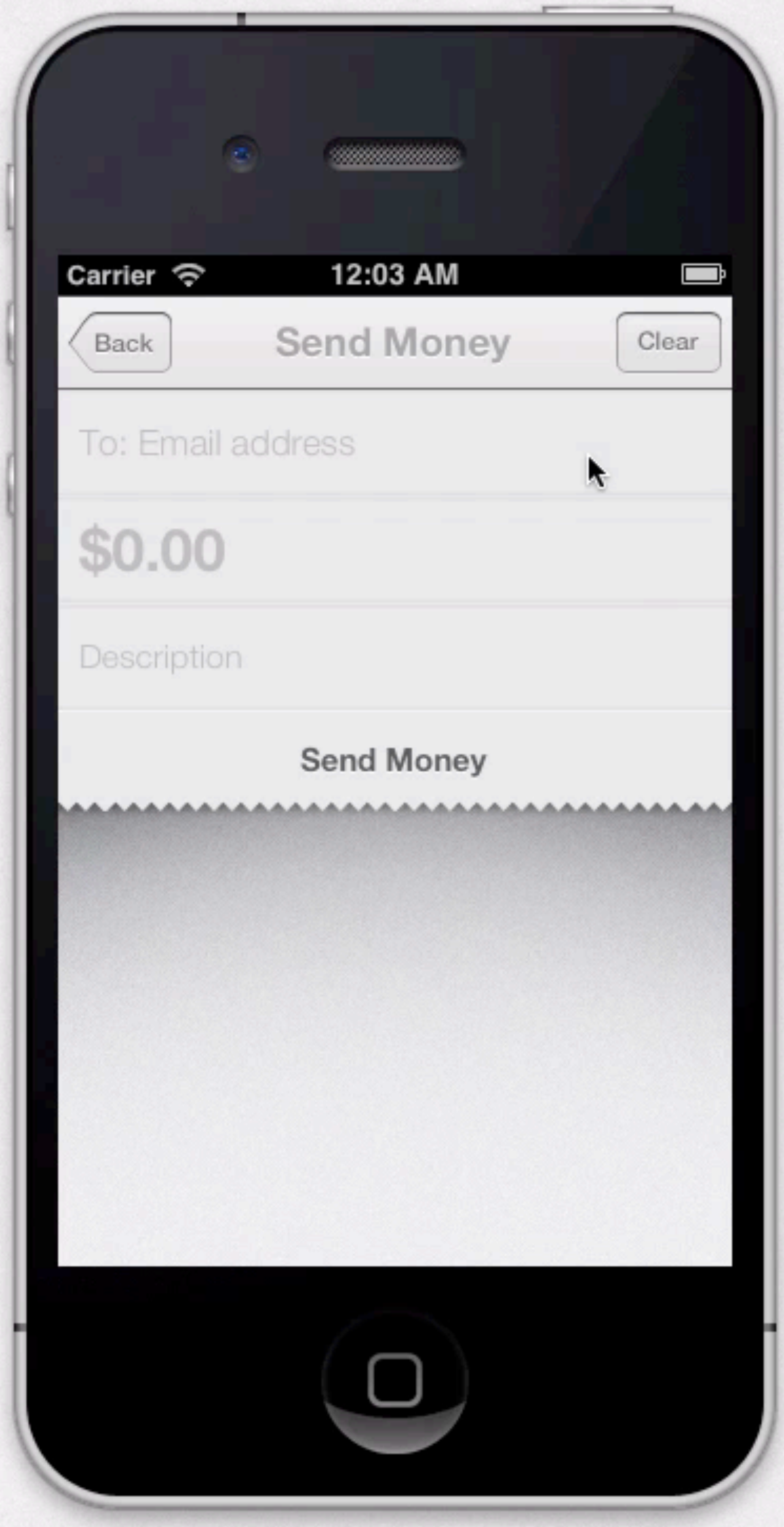
3. Sends a POST request to simplemoney.dev/transactions/

# QuickPay
- If QuickPay is selected, a ZBar camera controller will be activated and we can scan a QR Code that contains a merchant id.
- Once a QR Code can be recognized,
- the camera controller automatically dismisses itself
- our app grabs the merchant id, builds a new transaction object,
- and sends that object as JSON to the transactions resource to create a new incomplete transaction.
- Our transaction model has a boolean flag that marks whether or not the transaction is complete.

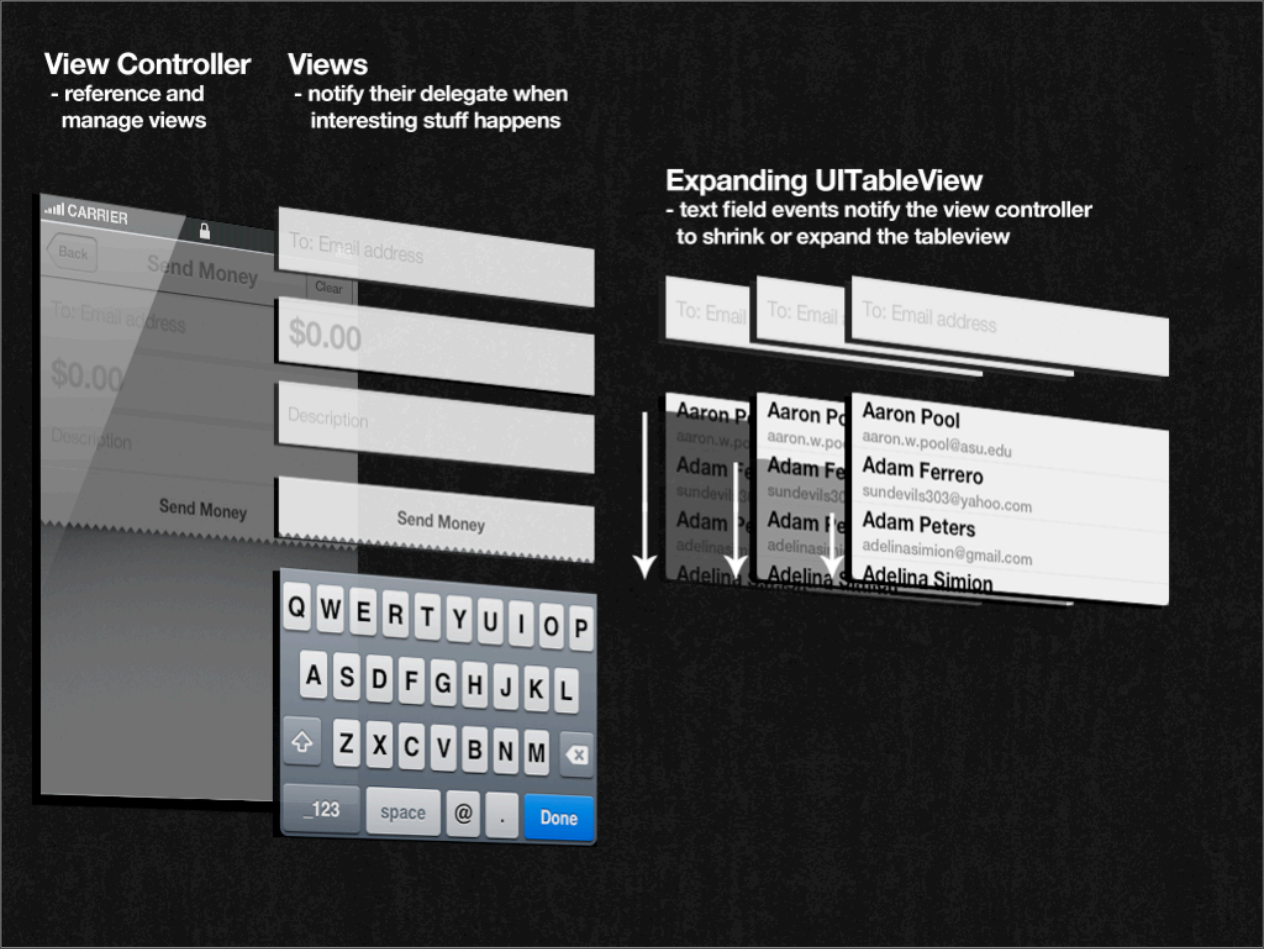
- Similar to the process of authorizing a charge on your credit card when you rent a car or a hotel room, a user can simply scan a QR Code at checkout to authorize the merchant to charge their account.
- Once the QR Code is scanned, the user is free to go, and the merchant can update the transaction with the proper amount and mark the transaction complete.

Send & Request Money



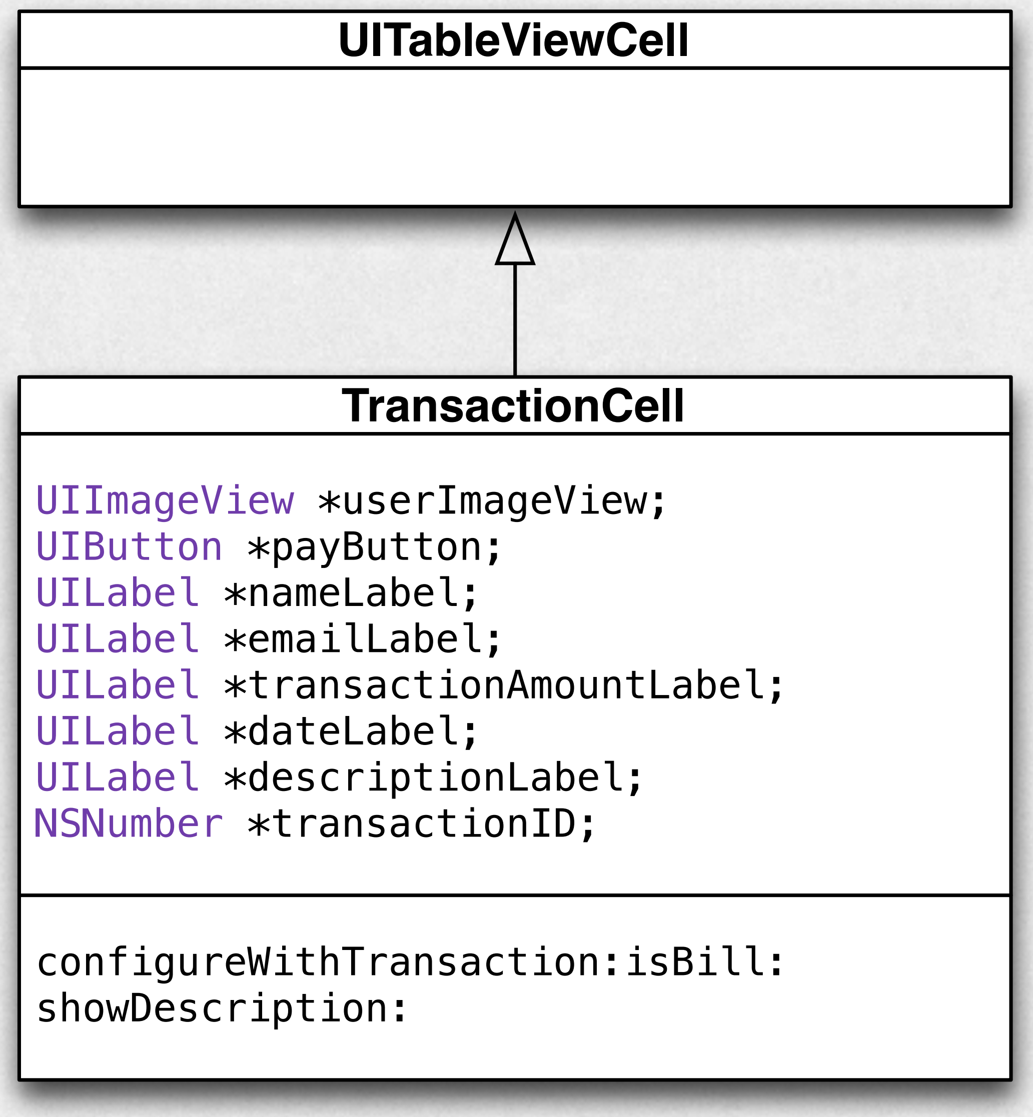


# Send and Request Money
- Our app makes it easy to send or request money from your friends by reading from the phone's address book.
- This view allows you to search through your contacts by name or email address
- The table view that lists your contacts appears when you are entering a recipient email address, and gracefully disappears otherwise.



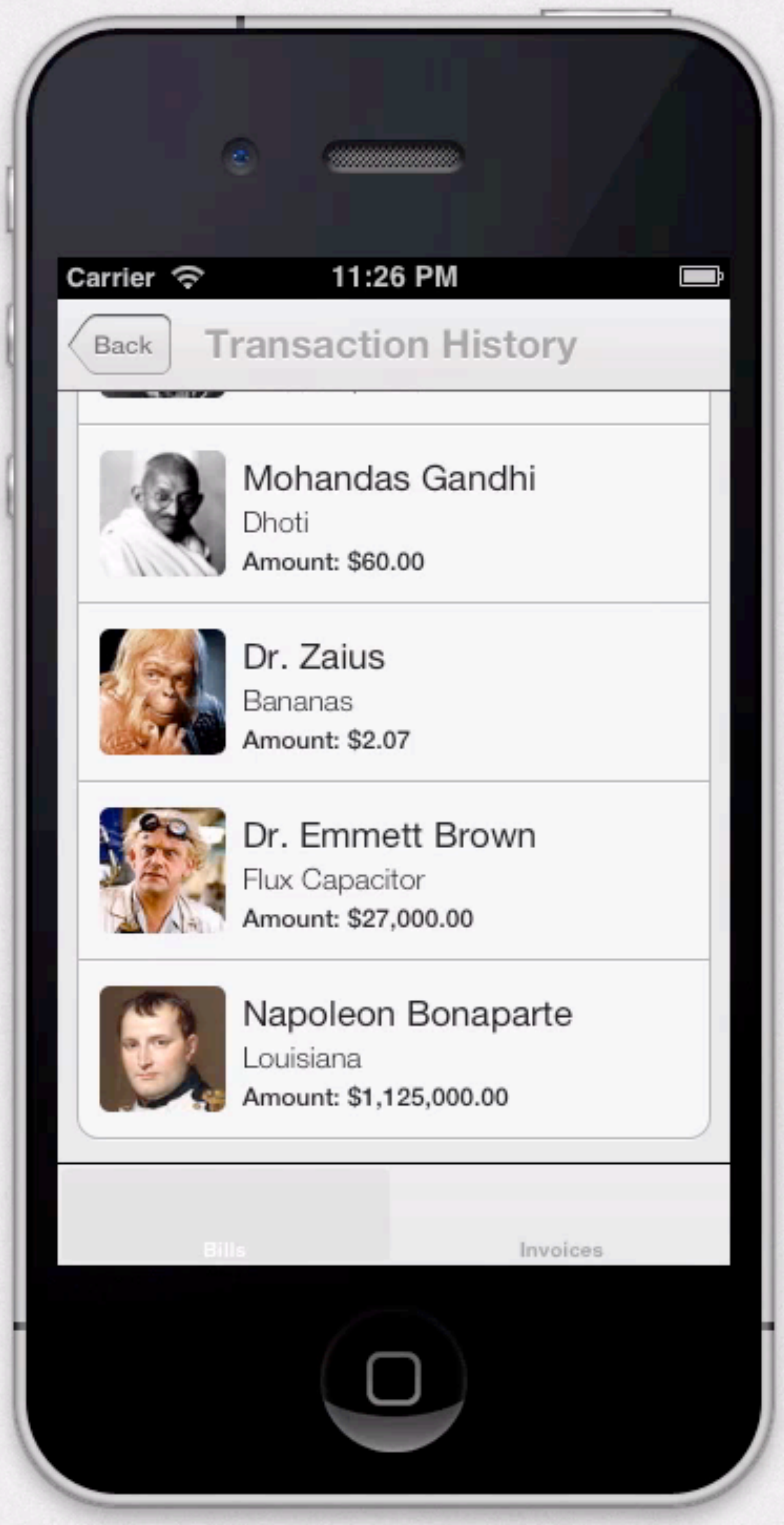
# SendMoney diagram
- Here's a diagram that shows the delegate pattern in action.
- The view controller references and manages several views, seen here in the middle.
- Views notify their delegate, or the view controller when something interesting happens.
- This pattern makes to easy to manage views and create unique user interactions outside of the iOS SDK.
- For example, the expanding TableView that we saw in the last video takes advantage of the delegate pattern quite well. When the email address text field receives a touch event, the view controller expands the tableview and hides the other text fields. When a contact is selected from the tableview, it notifies the view controller and the tableview is dismissed, and the other text fields are faded in.

## Transaction Cell



```
UIImageView *userImageView;
UIButton *payButton;
UILabel *nameLabel;
UILabel *emailLabel;
UILabel *transactionAmountLabel;
UILabel *dateLabel;
UILabel *descriptionLabel;
NSNumber *transactionID;

configureWithTransaction:isBill:
showDescription:
```



Mohandas Gandhi
Dhoti
Amount: $60.00

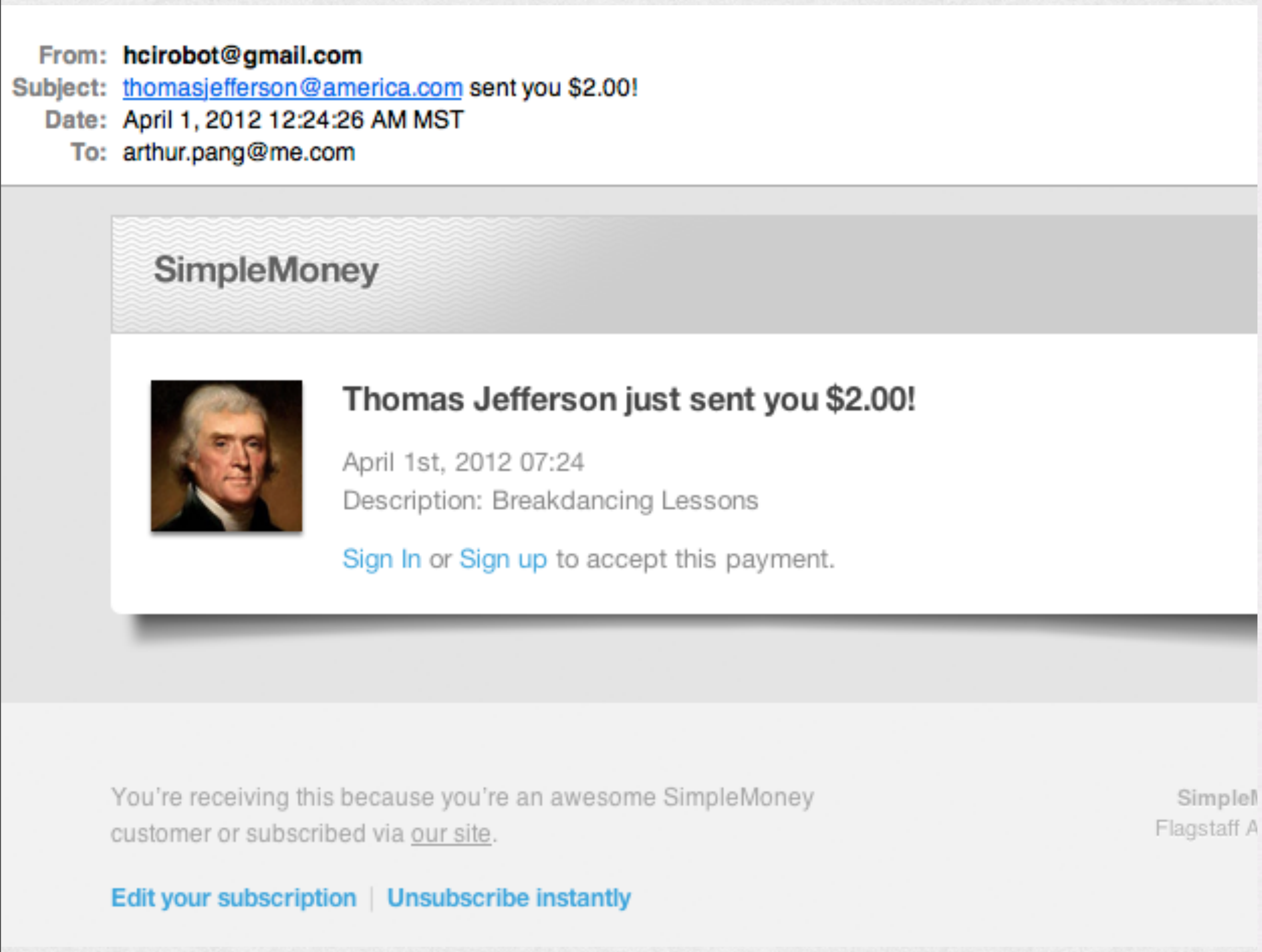
Dr. Zaius
Bananas
Amount: $2.07

Dr. Emmett Brown
Flux Capacitor
Amount: $27,000.00

Napoleon Bonaparte
Louisiana
Amount: $1,125,000.00

# Transaction Cell
- Another UI component we built was the Transaction cell.
- We subclassed a TableView cell to accept a transaction object as a parameter and expand when it receives a touch event.

## Pay a Bill



From: hcirobot@gmail.com
Subject: thomasjefferson@america.com sent you $2.00!
Date: April 1, 2012 12:24:26 AM MST
To: arthur.pang@me.com

SimpleMoney

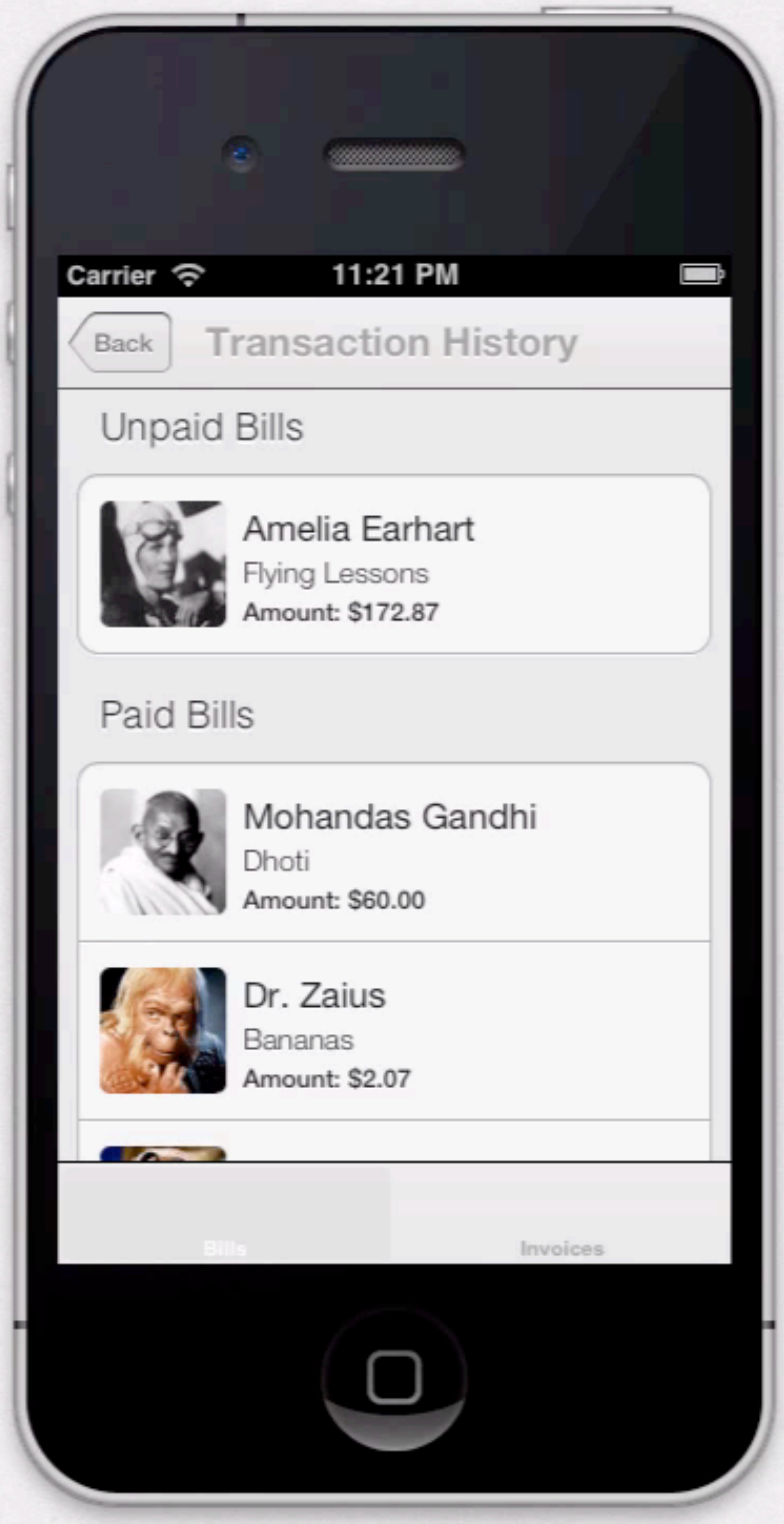
Thomas Jefferson just sent you $2.00!

April 1st, 2012 07:24
Description: Breakdancing Lessons

Sign In or Sign up to accept this payment.

You're receiving this because you're an awesome SimpleMoney customer or subscribed via our site.

Edit your subscription | Unsubscribe instantly



Carrier 11:21 PM

Back Transaction History

Unpaid Bills

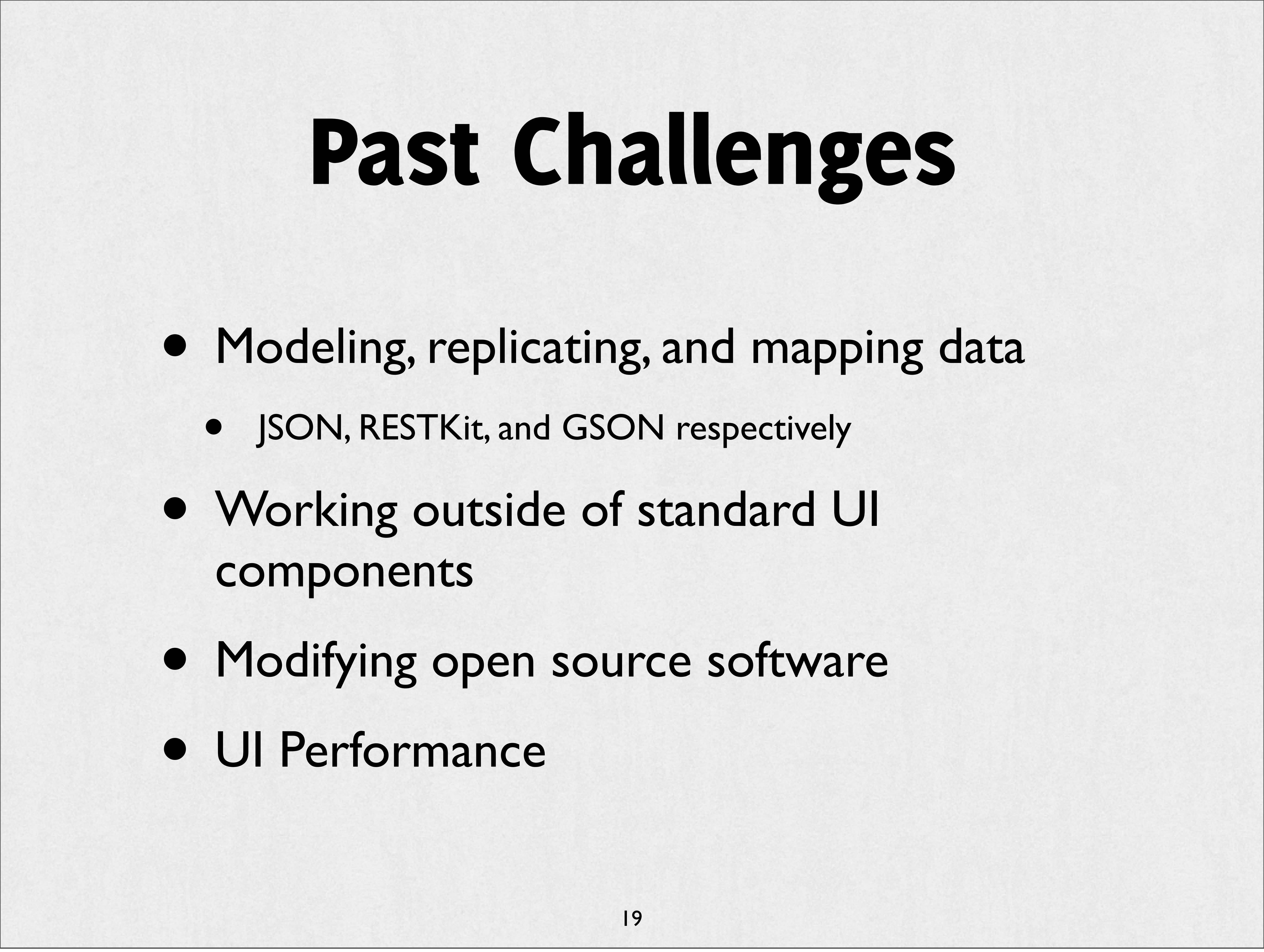
Amelia Earhart
Flying Lessons
Amount: $172.87

Paid Bills

Mohandas Gandhi
Dhoti
Amount: $60.00

Dr. Zaius
Bananas
Amount: $2.07

# Pay a Bill
- Our app also allows users to pay bills by selecting the unpaid bill, and tapping on the pay button.
- The app simply updates the transaction locally and sends a PUT request to the server, and the server moves the money from the sender's account to the recipient's account. When the money is sent and the transaction is updated, our server uses MailGun to email a receipt to the recipient.



Past Challenges

- Modeling, replicating, and mapping data
  - JSON, RESTKit, and GSON respectively
- Working outside of standard UI components
- Modifying open source software
- UI Performance

modeling data - stupid mistake: representing money as floats instead of ints
- setting up relatonships between users, transactions, items

replicating data - user and transaction models

apple's UI components abstracts a lot of difficulties away - custom UI requires you to dive deeper
o

adding ARC support to pubnub wrapper

Current Challenges

- What is electronic money?
- Matching Android UI to iPhone UI
- Local coupons
  - Location data

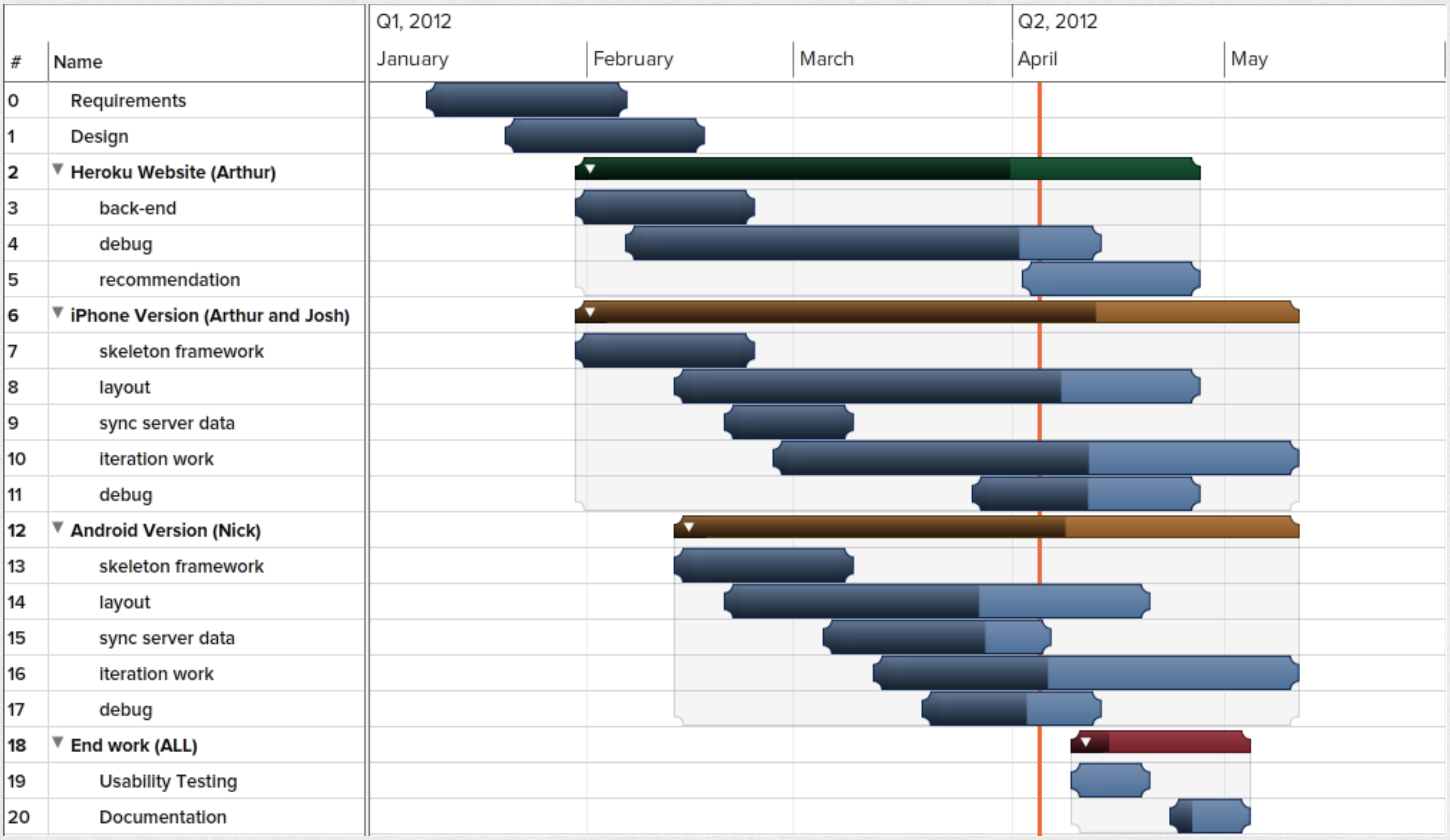
Adding merchant features –
to date, we have backend support for adding purchase items and their associated data, such as images.

Not using ACH – automated clearing house – API to transfer money between accounts
Currently, our app is using play money

Matching Andriod UI – we might have to build A LOT of custom UI to match apple’s ui components
tableviews?

Local coupons – we don’t have an data for local businesses

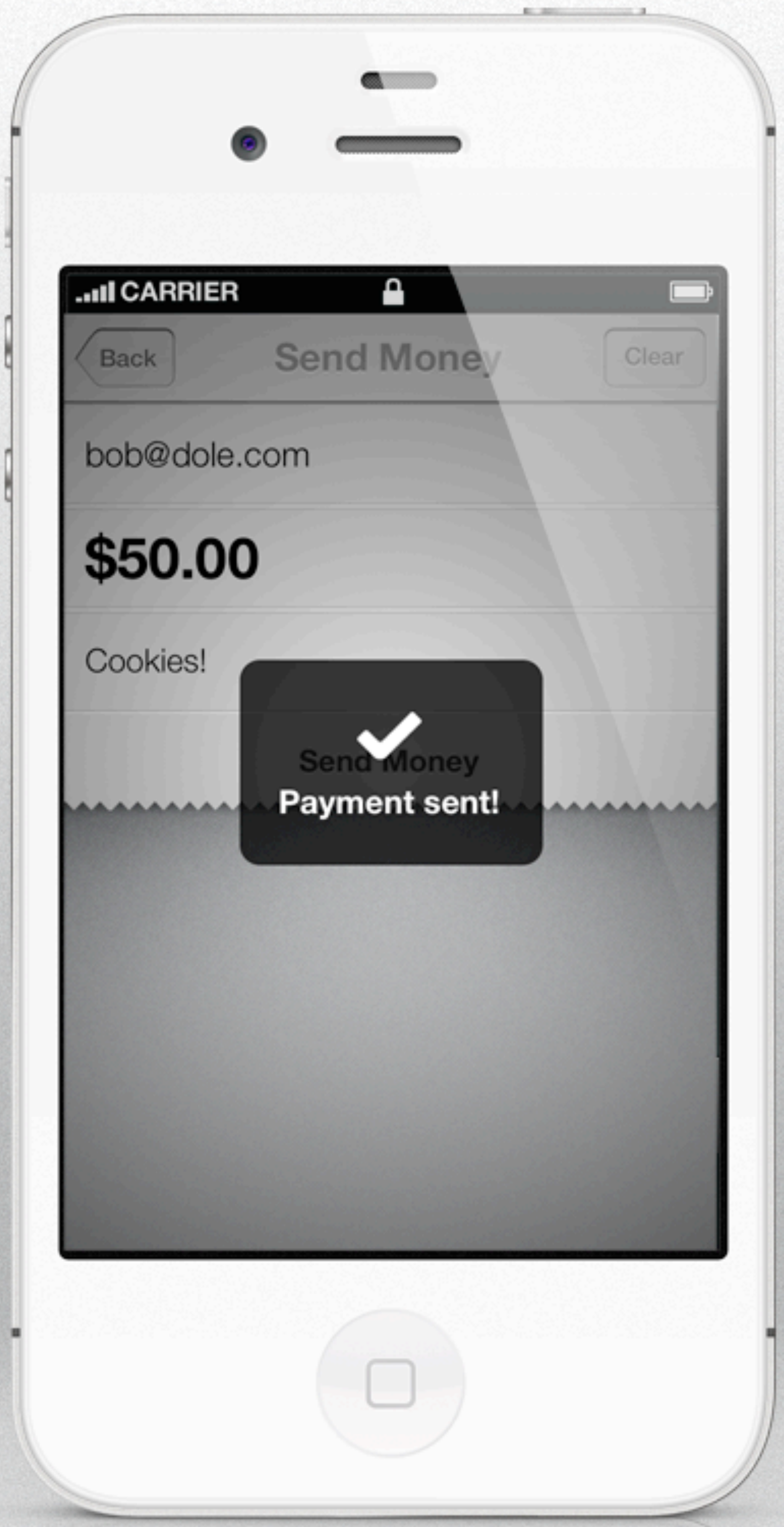
Schedule



#	Name
0	Requirements
1	Design
2	▼ Heroku Website (Arthur)
3	back-end
4	debug
5	recommendation
6	▼ iPhone Version (Arthur and Josh)
7	skeleton framework
8	layout
9	sync server data
10	iteration work
11	debug
12	▼ Android Version (Nick)
13	skeleton framework
14	layout
15	sync server data
16	iteration work
17	debug
18	▼ End work (ALL)
19	Usability Testing
20	Documentation

Conclusion

- Simple, flexible and powerful payment solution
- Replace the credit card
- $48 billion waste!!!



Money is tied to hardware.
Credit cards are stupid - they don’t tell you your balance or transaction history
banks offer apps that let you check your balance, why not take a step further?