

# Not Our (Seg)Fault!

Connor Black

Damien King-Acevedo

Edward Skrod

Alex Windelberg

# Presentation overview

- Team Members
- Introduction
- Problem statement
- Project justification
- Similar Applications
- Scope
- Schedule
- Requirements
- Analysis of requirements
- Design
- Conclusions

# Introduction

- Lend.me is a debt management application that will allow users to track incoming and outgoing debts.

It will serve three main purposes:

1. Facilitate the creation and recording of a loan contract
2. Remind all parties of their obligation and due
3. Facilitate repayment

# Team members

- Connor Black – Prototype, Front End, Backend, JSON Documents
- Damien King-Acevedo – Use Case Scenarios, Business Logic, UI Layer
- Edward Skrod – Database, SRS, Backend, UI Layer
- Alex Windelberg – SRS, Backend, Business Logic, UI Layer

# Problem Statement

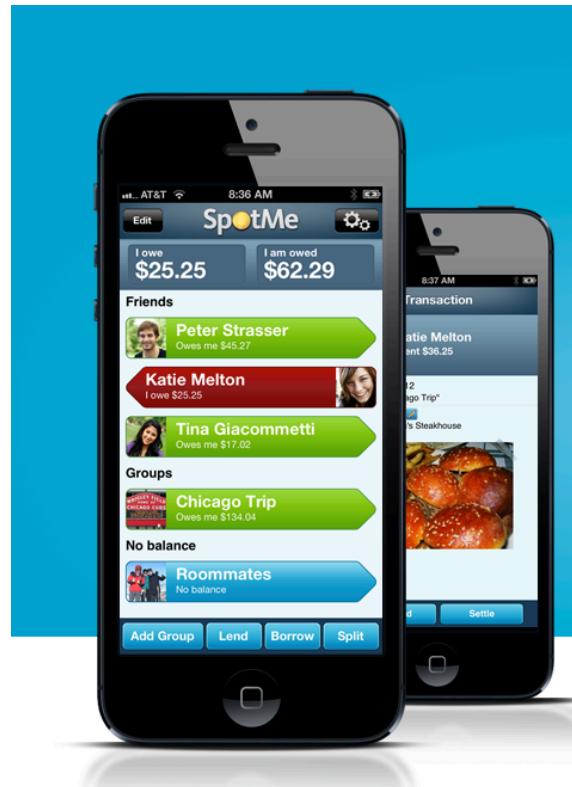
- Lending money to family and friends, even in small amounts, can be a hassle.
- It is easy to forget who owes money to who.
- Asking for repayment can be tedious.
- If repayment never comes, it would be nice to have a paper trail.

# Project justification

- Similar products exist but fail to provide all the functionality we want from an app
- Current applications can cost as much as \$15 to purchase

# Similar applications

- ZimpleMoney
- Venmo
- CashLender
- SpotMe



# Zimplemoney

- Cloud based
- Data is password protected
- Identity verification
- Users can track lenders and borrowers
- Multiple options for lending money
- Multiple types of loans recognized (amortized, interest only, etc.)

The screenshot shows the ZimpleMoney homepage. At the top, there's a navigation bar with 'Register' and 'Login' buttons. Below the navigation, social media links for Facebook, Twitter, and LinkedIn are displayed, along with a 'Like' button showing 110 likes and a 'Follow' button showing 922 followers. A 'Login with Facebook' button is also present. The main content area features a purple banner with the heading 'How It Works' and the subtext 'finance your passions'. To the right of the banner are two green buttons: 'How It Works' and 'Why ZimpleMoney?'. On the left side, there's a sidebar with links to 'How It Works', 'Why ZimpleMoney', 'Who are ZimpleMoney Members?', 'Loan Tools', 'Pricing Plans', and 'Get Started Now!'. Below this sidebar is a small image of a booklet titled 'Lending with a Purpose'. The central part of the page contains three steps: 'Getting Started is as Zimple as 1,2,3'. Step 1: 'Determine how much money you can afford to borrow. Using the Zimple calculator, calculate the amount of the loan, payment plan and terms. >> Use the ZimpleMoney Calculator.' Step 2: 'Sign up to be a Zimple member. All you need is your name and an email address. >> Register' Step 3: 'Invite your family or friends. Your family and friends can review and accept the agreed upon loan terms. >> Sign up now to start a loan'. At the bottom of the page, there's a link 'Get started now' and a 'DOWNLOAD ZIMPLEMONEY'S PRICING PLANS' button.

The screenshot shows the ZimpleMoney Calculator interface. The title 'ZimpleMoney Calculator' is at the top. Below it, there are three input fields: 'Loan Amount' set to '\$ 10,000.00', 'Interest Rate' set to '10 %', and 'Loan Term' set to '3'. Next to the 'Loan Term' field are two radio buttons: one for 'Months' (which is selected) and one for 'Years'. At the bottom of the calculator is a large blue 'Calculate >' button.

\$98.71

**Diana J paid Jack Y** Noms 1m   
Like Comment

**Ryan V charged Tyler V** Dinner 1m   
Like Comment

**Chris R paid Calvin A** Jimmy Johns 1m   
Like Comment

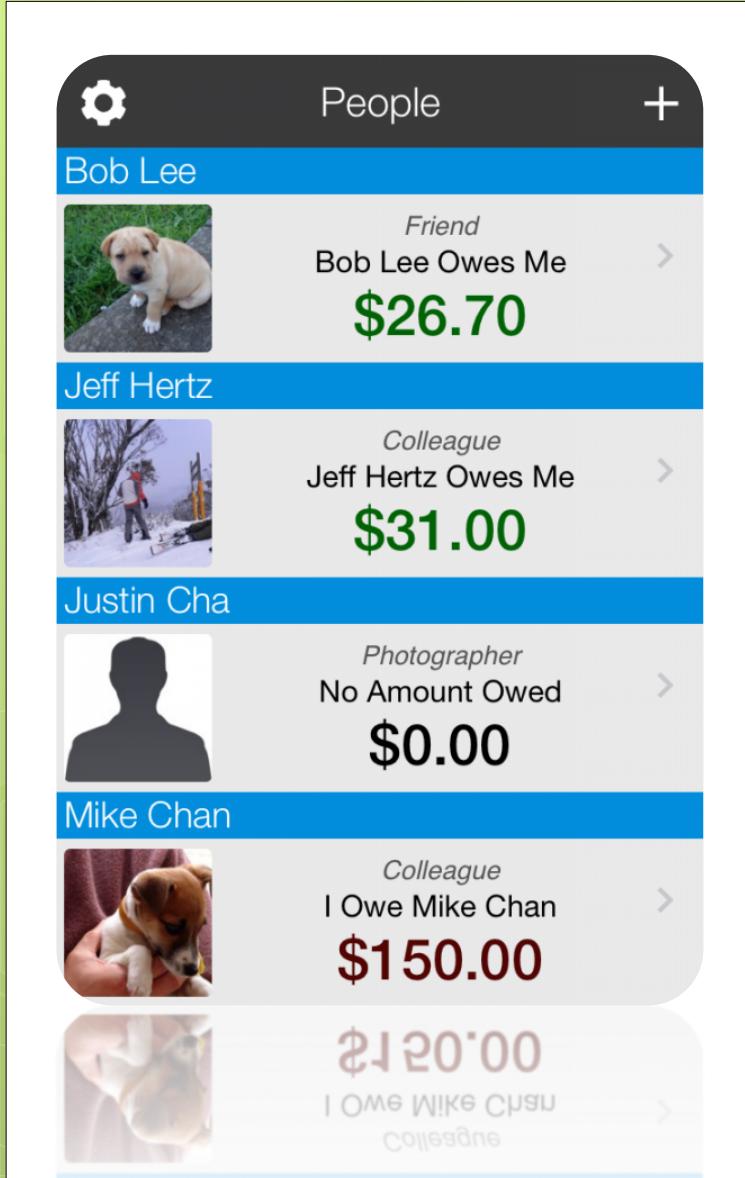
**Sarthak M charged Tyler V** Copious consumption 1m   
Like Comment

**Sudi R paid Biruk B** for u da best! 2m   
Like Comment

**James G paid Erin T** Paint night tip 2m   
Like Comment

## Venmo

- Not a lending app, a payment app
- Log In with Facebook
- Social – Connect with Friends through the Facebook API
- Limited features. No concept of loans or debts.
- No reminders.
- Cost: Free to join. Free to send via Bank. 3% fee if paying with credit cards



## Cash Lender

- iCloud compatible
- Manage multiple people
- Lend and borrowing money
- Passcode protection
- Email debt reminders
- Export history as CSV files
- Set debt due dates with reminders using notifications
- Total sum of debts per person
- Supports partial repayments
- Supports local currency
- **Cost: \$1.99 to \$14.00**



## SpotMe

- Register with Facebook
- Lend to an email address
- Can split bill with “Create Group”
- Can invite friends
- Manually remind borrower by sending an email alert
- Rating: 4+ / 5 Stars on iTunes
- Cost: Free

# How is Lend.me different?

- Free to use. Small charge when payment made within the system
- User will have ability to make payments from a credit card or bank account
- Simple, web 2.0 interface and design

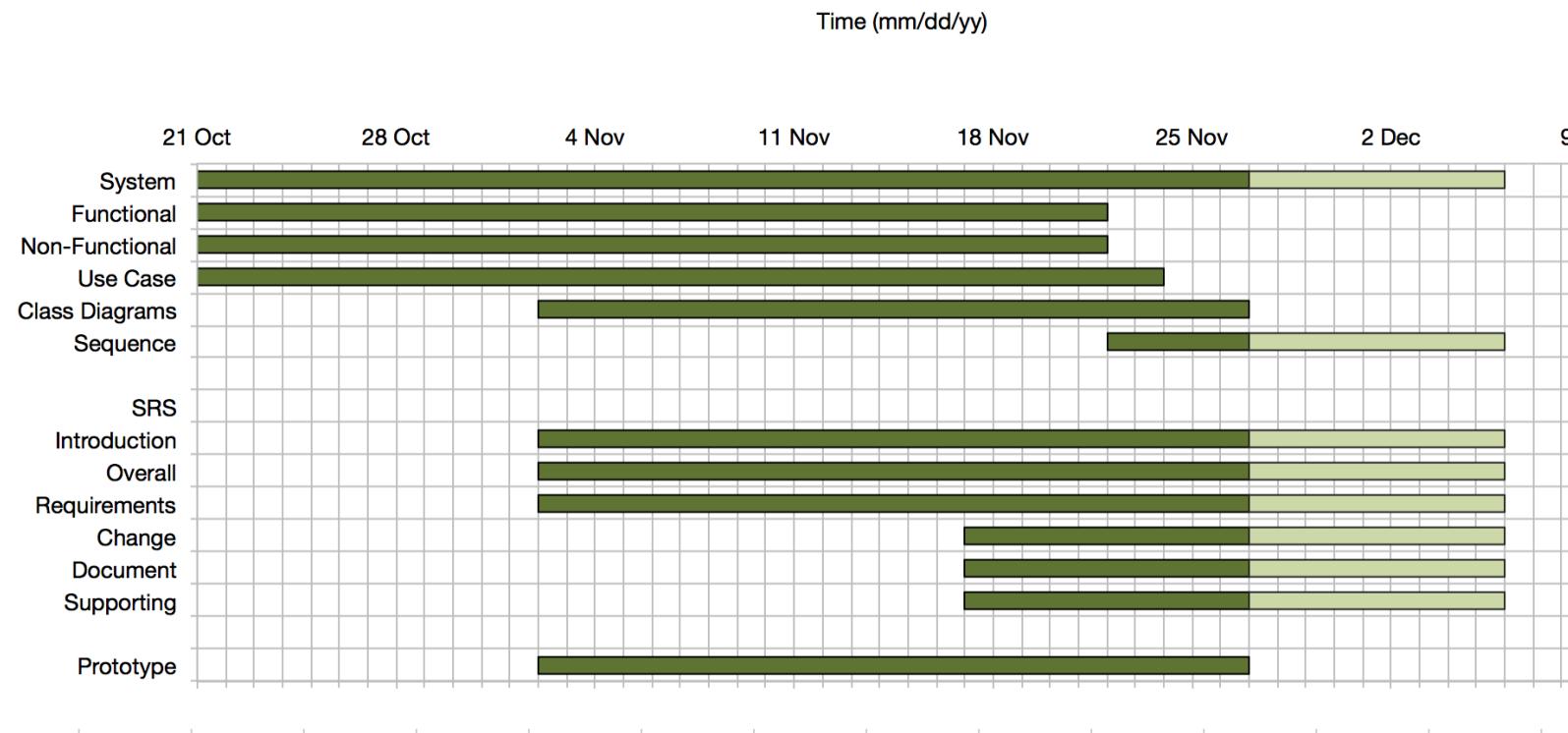
# Analysis of Similar Applications

- Optimally we would like to incorporate features from each of the above apps
  - The loan management from SpotMe
  - The Interface from Cash Lender or SpotMe
  - Payment management from Venmo
  - Zimplemoney served to showcase how not to design a website. It has more features than we would like to implement.

# Scope

- The scope of this application, in the short term, is to produce a working prototype with maximum functionality for the class.
- The project is scalable to the English-speaking world and beyond with minimal further requirements. To take it further, we would need:
  1. A commercially viable name
  2. Multi-lingual support
  3. Support for local currencies
  4. Improved payment system

# Schedule



# Functional Requirements

- Visitor

1. must register for an account to use the system by providing a unique email and password.

- User

1. is able to recover a lost username by entering their email addresses
2. must login to the system using his/her unique username and password.
3. is able to change his/her password.
4. is able to recover a lost password.
5. is able to query for up-to-date transaction information.
6. User is notified by email of a reminder of loan due.
7. can add a friend.

# Functional Requirements

- Lender

1. can start and set terms on a new loan
2. can agree or disagree to new terms of the loan
3. can access a list of loans outstanding
4. can click on loan outstanding to access more information about it
5. can agree or disagree to a loan extension
6. can forgive the loan (clear it out)
7. can send loan money via app or with cash
8. can send a personalized reminder to the borrower

- Borrower

1. can agree or disagree to terms of new loans
2. can set new terms of the loan
3. can ask for a loan extension
4. can ask for loan forgiveness
5. can pay back loan by using the application or cash

# Non-Functional Requirements

- Usability

1. An FAQ should be available to help visitors register for an account
2. An FAQ should be available to help users lend, borrow or make a payment on a loan
3. The system should prevent users from changing terms in a contract with the opposite party's consent

- Security

1. The system should serve an HTTPS server
2. The system should protect API requests by requiring either an email/password or cookie

# Non-Functional Requirements

- System Interface

1. The website should have a modern, uncluttered, Web 2.0 look
2. The system should have the option to register and authenticate with Facebook

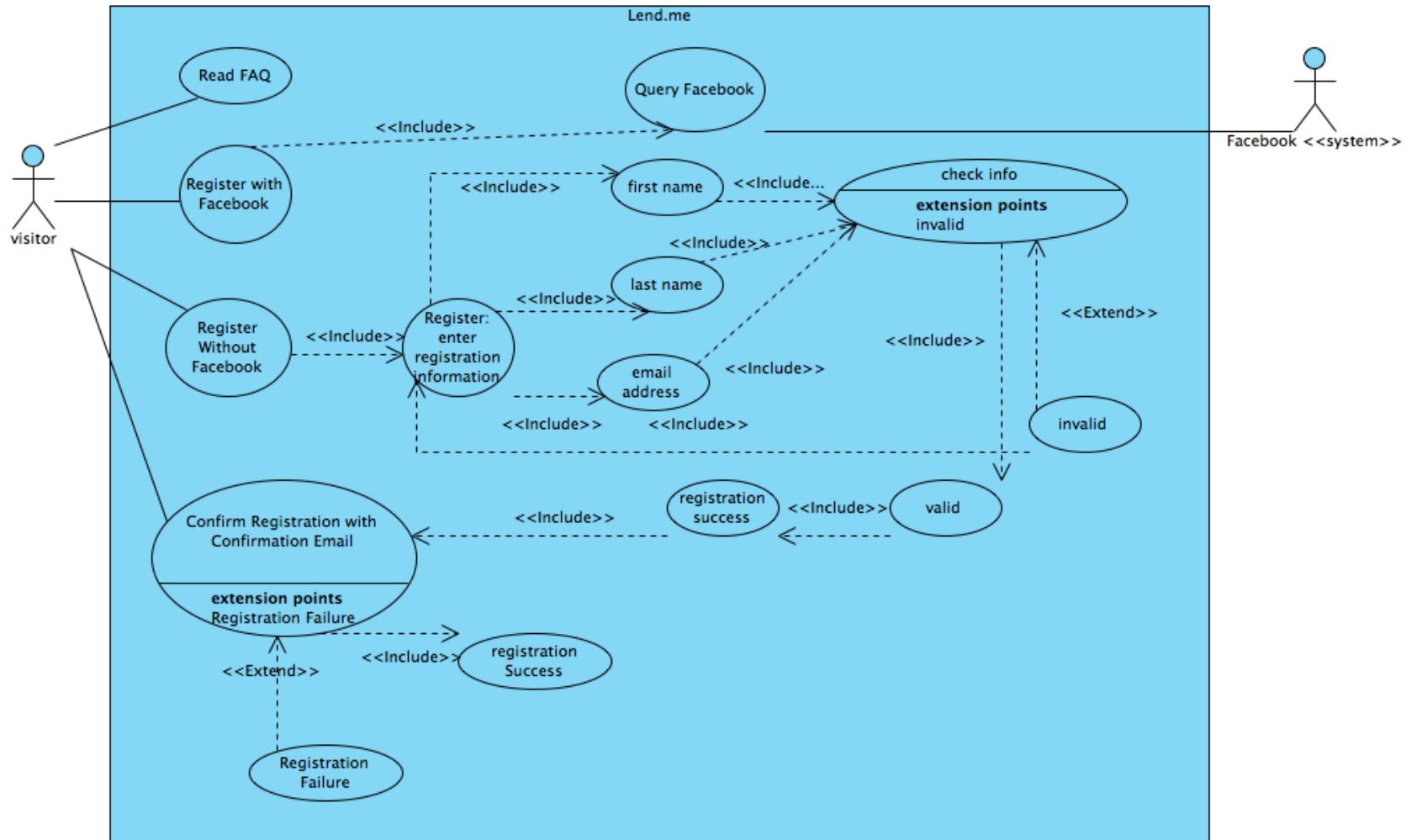
- Operation

1. The system should verify a user's email before allowing her to continue registration
2. The records returned to lenders and borrowers during a session should be updated in real time. The system should be capable of updating the displayed record count every few seconds

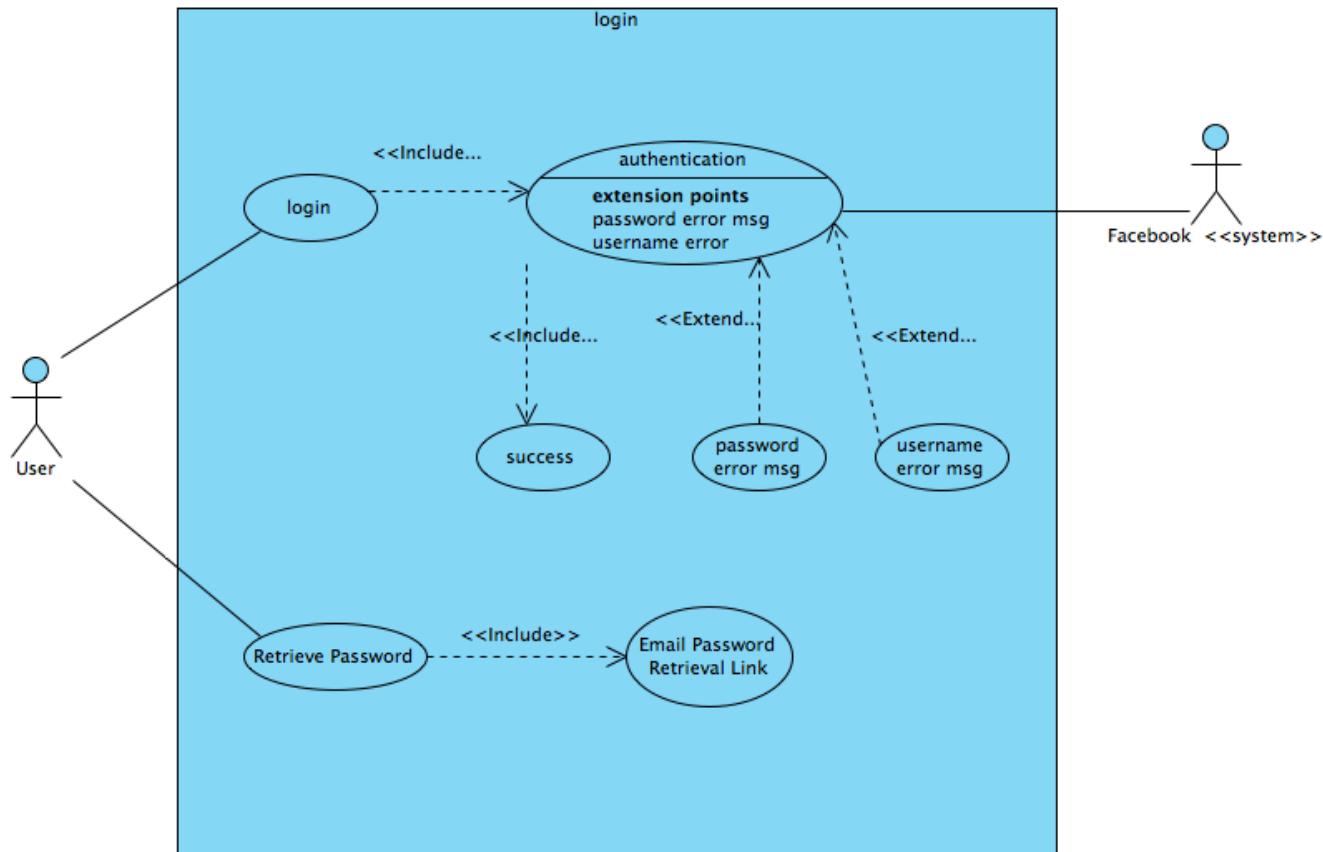
# Design

- Use Case Scenarios
  - Visitor use cases
  - Login use cases
  - User use cases
  - Lender & Borrower specific use cases
  - Class Diagram
  - Database modeled with MySQL Workbench
  - Data Dictionary
  - Data modeled with JSON documents
  - Prototype

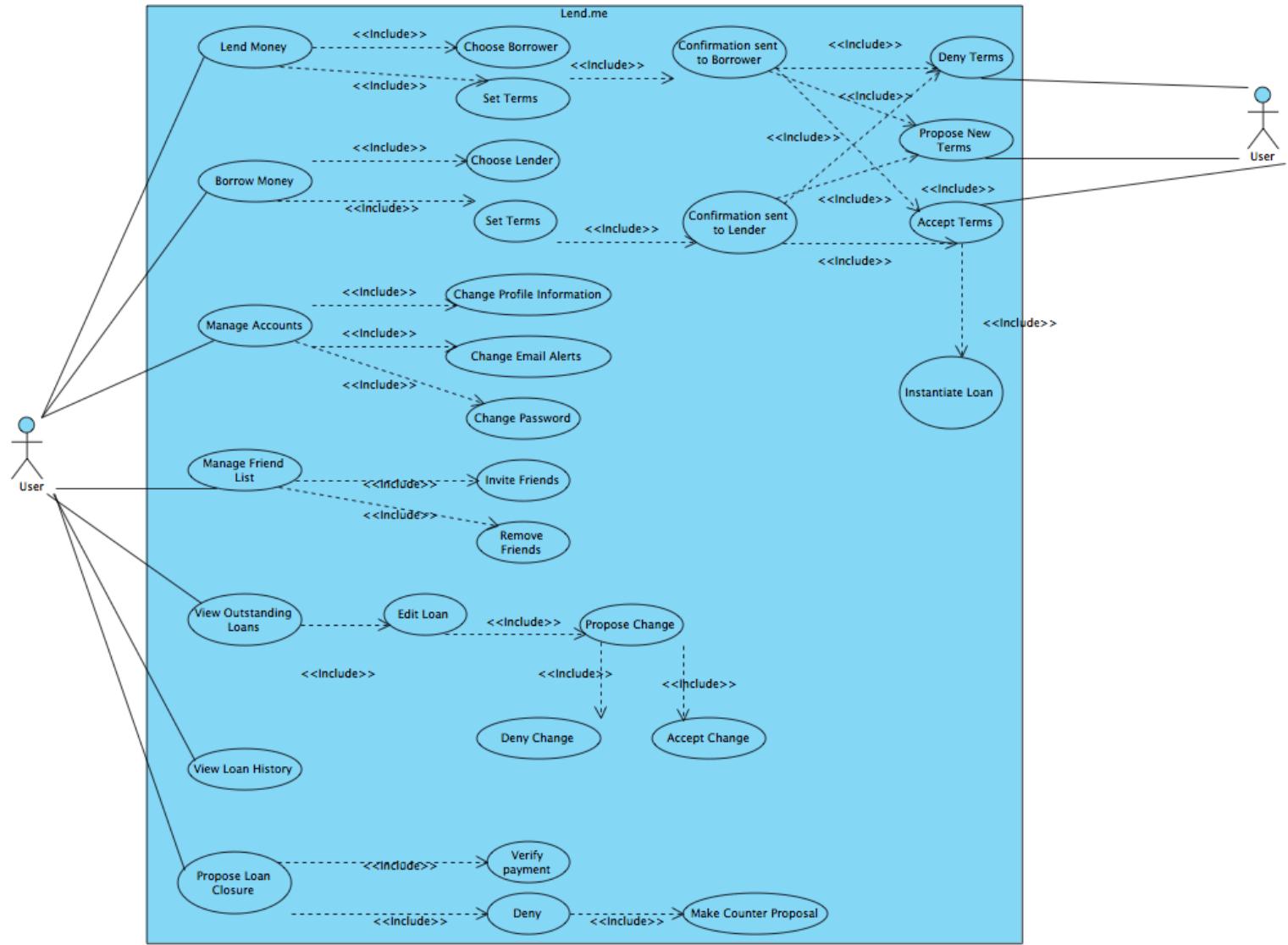
# Visitor use cases



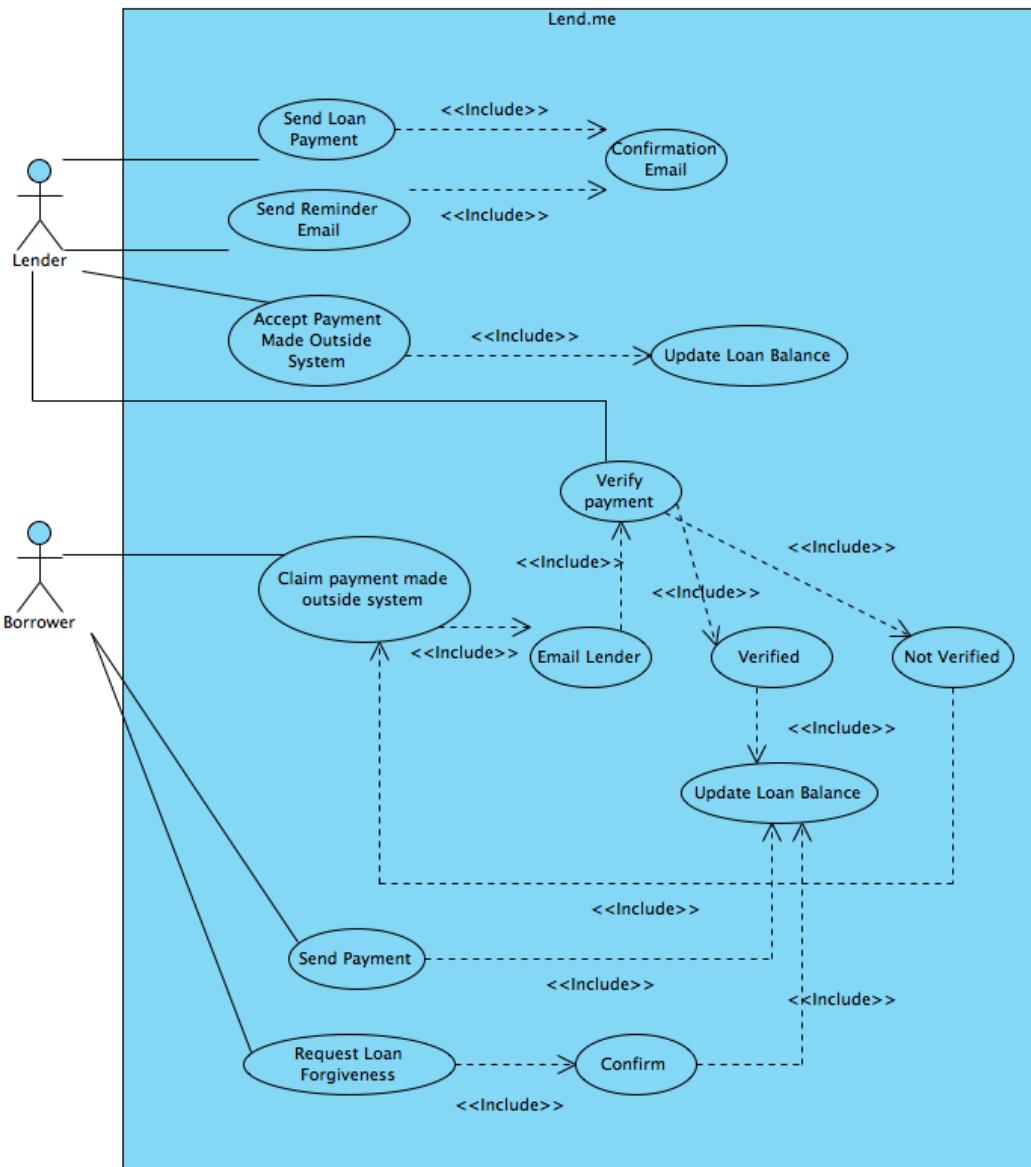
# User login use cases



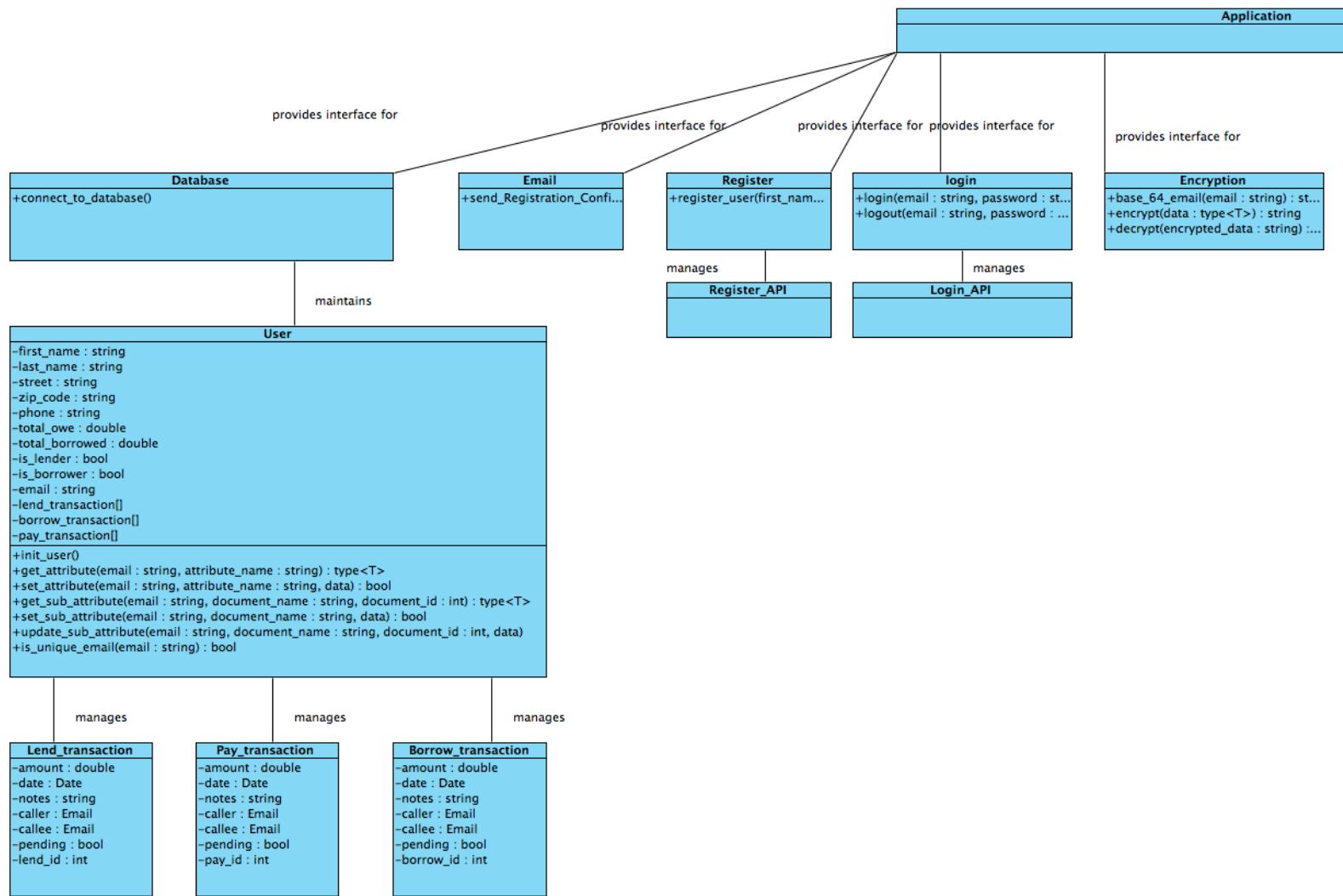
# User use cases



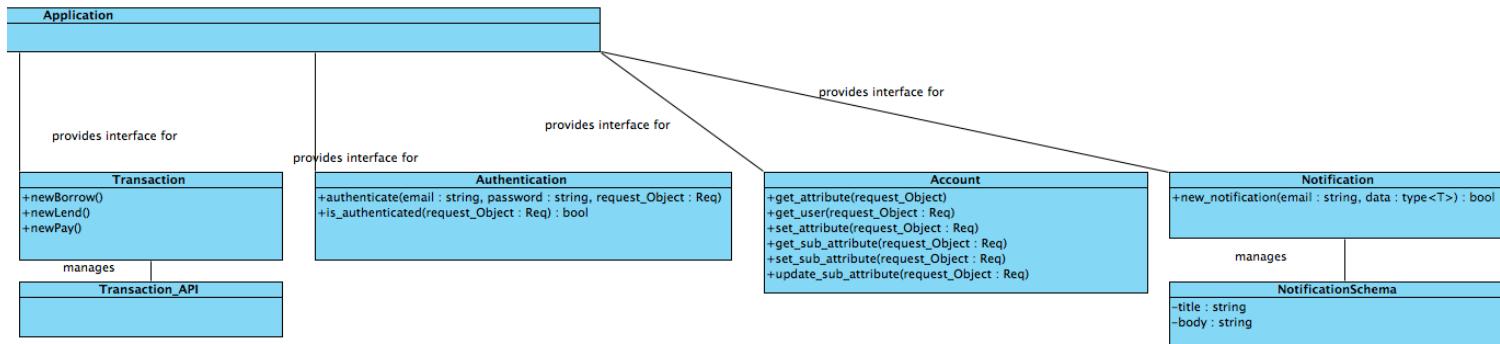
# Lender & Borrower Specific Use Cases



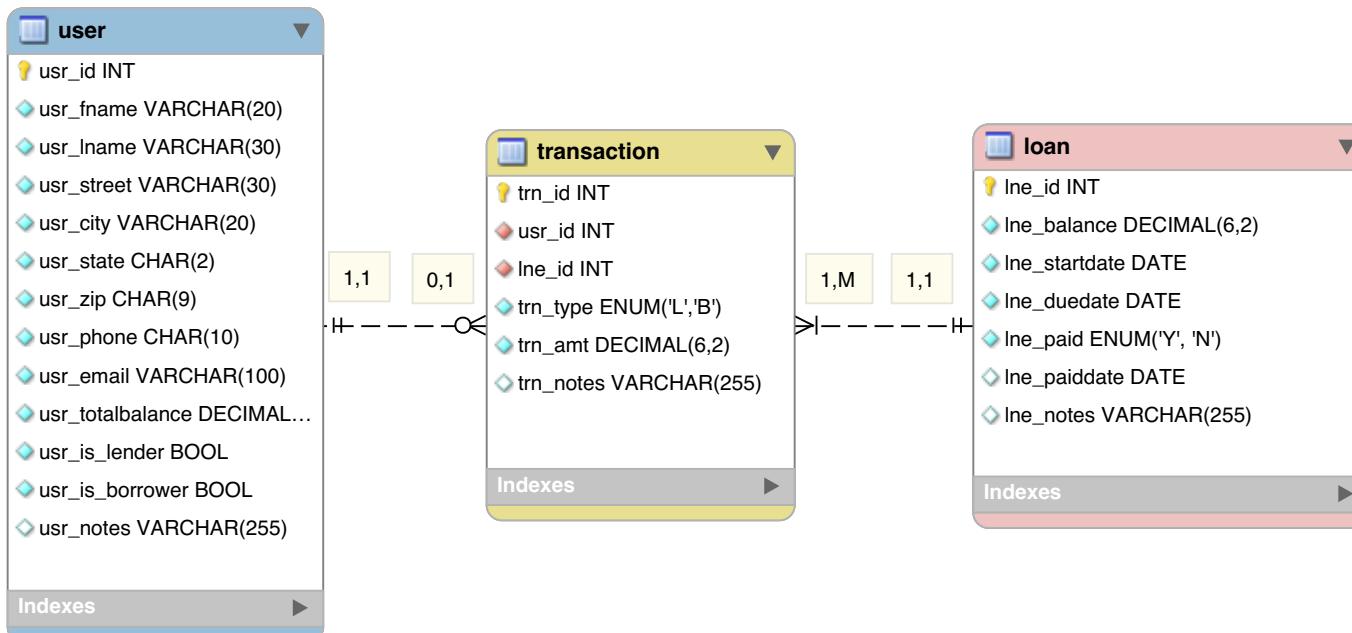
# Class Diagram



# Class Diagram cont.



# Preliminary Database Modeled in MySQL



with MySQL workbench

# Data dictionary

Table Name	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK Reference Table
user	usr_id	User Identification	INT	123456	0-9	Y	PK	
	usr_user_name	Username	STRING	usrName	a-z, 0-9	Y		
	usr_fname	User First Name	STRING	First	a-z, 0-9	Y		
	usr_lname	User Last Name	STRING	Last	a-z, 0-9	Y		
	usr_street	User Street	STRING	123 Street	a-z, 0-9	Y		
	usr_city	User City	STRING	City	a-z, 0-9	Y		
	usr_state	User State	STRING	AZ	a-z	Y		
	usr_zip	User Zip Code	STRING	12345	0-9	Y		
	usr_phone	User Phone Number	STRING	5553211234		N		
	usr_email	User Email	STRING	xxxx@xxx.com	a-z, 0-9	Y		
	usr_totalbalance	User's Total Balance	DOUBLE	1234567.89	0-9	Y		
	usr_is_lender	User is a lender	BOOL	TRUE	TRUE or FALSE	Y		
	usr_is_borrower	User is a borrower	BOOL	TRUE	TRUE or FALSE	Y		
	usr_notes	User Notes	STRING	Notes	a-z, 0-9	N		
transaction	trn_id	Transaction ID	INT	123456789	0-9	Y	PK	
	usr_id	User Id	INT	123456789	0-9	Y	FK	user
	lne_id	Loan Id	INT	123456789	0-9	Y	FK	loan
	trn_type	Transaction Type	ENUM('L', 'B')	L	L, B	Y		
	trn_date	Transaction Date	DATE	2013-10-06	0-9	Y		
	trn_amt	Transaction Amount	DOUBLE	123456.12	0-9	Y		
	trn_notes	Transaction Notes	STRING	Notes	a-z, 0-9	N		
loan	lne_id	Loan Id	INT	123456789	0-9	Y	PK	
	lne_balance	Loan Balance	DOUBLE	1234567.12	0-9	Y		
	lne_startdate	Loan Start Date	DATE	2013-10-06	0-9	Y		
	lne_duedate	Loan Due Date	DATE	2013-10-06	0-9	Y		
	lne_paiddate	Loan Paid Date	DATE	2013-10-06	0-9	N		
	lne_paid	Loan Paid Back	BOOL	TRUE	TRUE or FALSE	Y		
	lne_notes	Loan Notes	STRING	Notes	a-z, 0-9	N		

# Database modeled for NoSQL

## User Schema

```
{  
    "_id": {  
        "$oid": ""  
    },  
    "modified": {  
        "$date": ""  
    },  
    "created": {  
        "$date": ""  
    },  
    "__v": ""  
  
    "email": "",  
    "password": "",  
    "street": "",  
    "zipcode": "",  
    "phone": "",  
    "total_owed": "",  
    "total_borrowed": "",  
    "is_lender": "",  
    "is_borrower": "",  
    "phone": "",  
    "last_name": "",  
    "first_name": "",  
  
    "lend_transaction": [],  
    "borrow_transaction": [],  
    "pay_transaction": []  
}
```

## Lend Transaction Schema

```
{  
    "_id": {  
        "$oid": ""  
    },  
    "modified": {  
        "$date": ""  
    },  
    "created": {  
        "$date": ""  
    },  
    "__v": ""  
  
    "amount": "",  
    "date": "",  
    "notes": "",  
    "caller": "",  
    "callee": "",  
    "pending": "",  
    "lend_id": ""  
}
```

JSON documents

## Borrow Transaction Schema

```
{  
    "_id": {  
        "$oid": ""  
    },  
    "modified": {  
        "$date": ""  
    },  
    "created": {  
        "$date": ""  
    },  
    "__v": "",  
  
    "amount": "",  
    "date": "",  
    "notes": "",  
    "caller": "",  
    "callee": "",  
    "pending": "",  
    "lend_id": "",  
}
```

## Pay Transaction Schema

```
{  
    "_id": {  
        "$oid": ""  
    },  
    "modified": {  
        "$date": ""  
    },  
    "created": {  
        "$date": ""  
    },  
    "__v": "",  
  
    "amount": "",  
    "date": "",  
    "notes": "",  
    "caller": "",  
    "callee": "",  
    "pending": "",  
    "lend_id": "",  
}
```

# Prototype

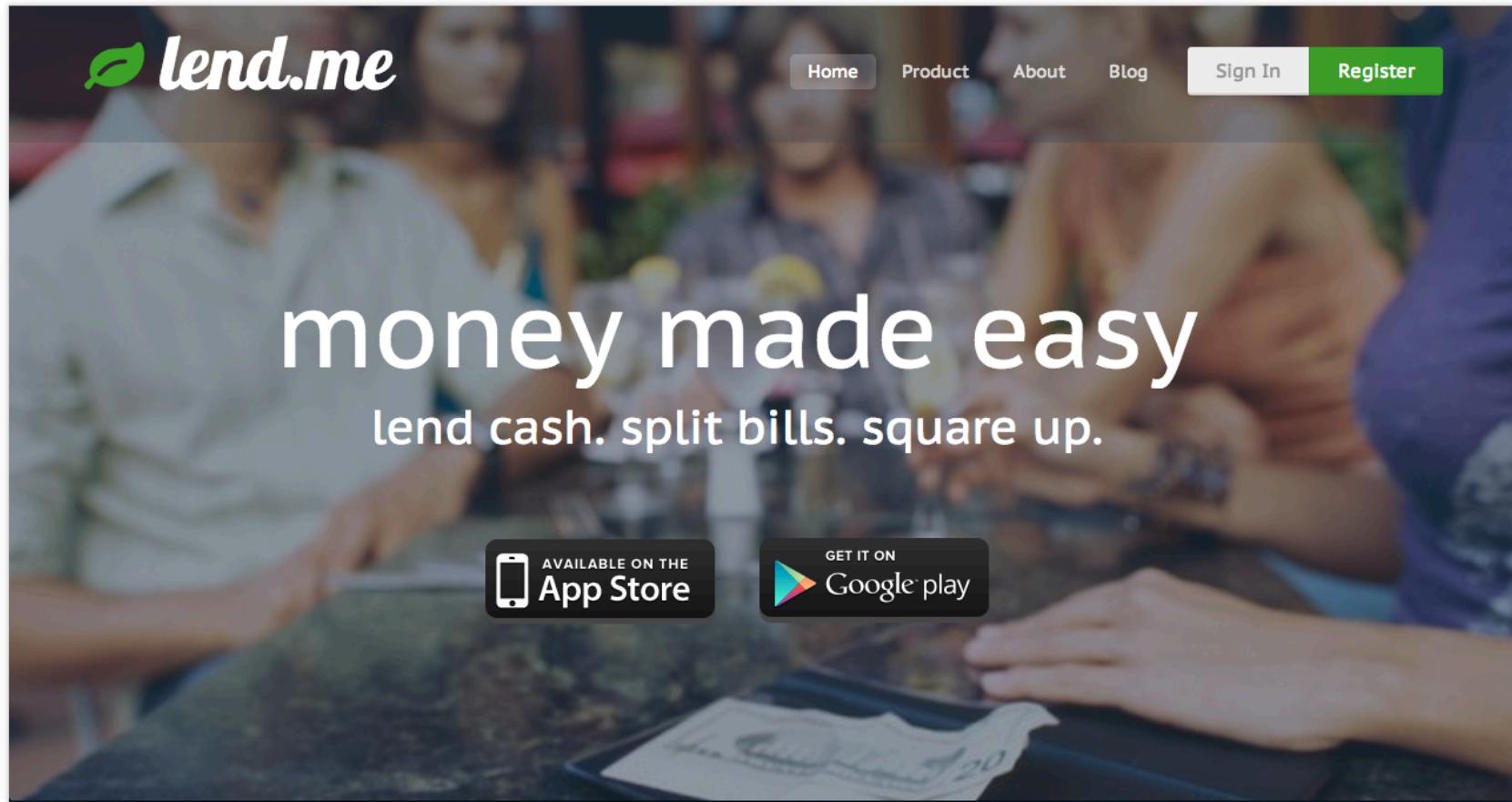
A wireframe prototype of a user interface for a lending or billing application. The top navigation bar includes a logo, user profile (Connor Black), and links for "New Transaction", "Logout", "Accounts", "Friends", and "Settings". The main content area displays a list of transactions under the heading "Owed \$250.00" and "Owe \$50.00".

Owed	To	Amount	Description	Action
\$25.00	by Edward	\$25.00	For dinner, due next week	Remind Edit
\$100.00	by Damon	\$100.00	For groceries, due in two days	Remind Edit
\$25.00	by Edward	\$25.00	For dinner, due next week	Remind Edit

# Conclusions

- There is a market for an application that provides the functionality of Lend.me
- The system is feasible to design within the time frame provided for the course.
- Challenges:
  - Mainly educational:
    - Javascript
    - Node.js
    - JSON

# Questions?



**lend.me**

Home   Product   About   Blog   Sign In   Register

# money made easy

lend cash. split bills. square up.

AVAILABLE ON THE  
App Store

GET IT ON  
Google play