FINAL PROJECT REPORT

1. Team information:

   Team #29

Name: *Zhiyue Li*	UNI: *zl2966*
Name: *Chang Lei*	UNI: *cl3910*
Name: *Rahul Sehrawat*	UNI: *rs3688*
Name: *Melody Zhao*	UNI: *xz2996*

2. Target user population:

   People who sometimes have group expenses with others. We don’t always share expenses with a group of people.

   We updated the target population from people who frequently have group expenses to people who sometimes have group expenses. This is because our application is a light-weighted one, people who always have group expenses are more willing to use existing solutions like splitwise.

3. Project concept:

   We make it easy for people to split the bill by removing redundant features that are unnecessary, for example, accounts. The burden of bills verification and payment management is mostly carried by the actual payer. Participants only need a few clicks to know the debt. The amount of money a user should pay is accurately calculated based on the specific items that the user claims.

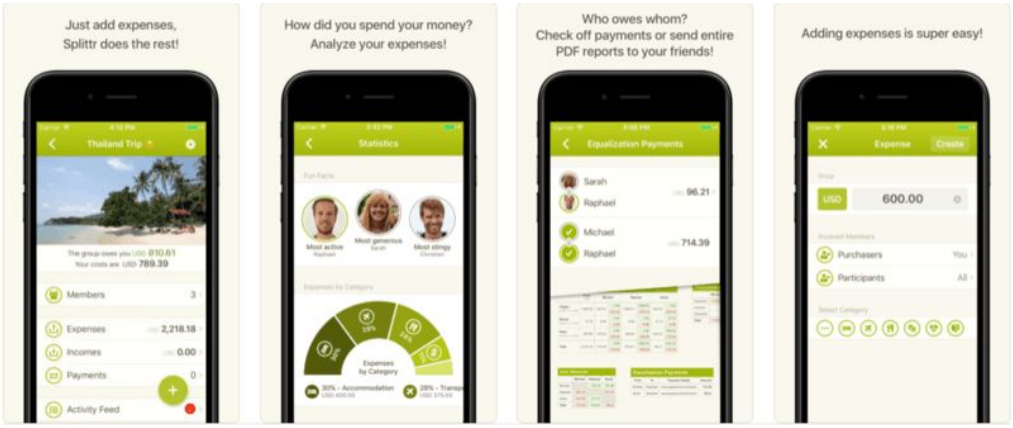
4. Our ambition is to make an intelligent and convenient bill splitting tool different from existing solutions. First, it will split the bill wisely, not on average or by percentage. Second, we want to make a super light-weighted tool that focuses on splitting the bill.

5. Comparative analysis (Updated):

   Splitter

   - Good:
     - Sole focus is on bill splitting
     - Straight forward process and customer journey and therefore high usability
     - Overall, very positive customer reviews for the above two reasons
   - Bad / Areas of improvement:
     - Focus is heavily on group trips and therefore the tracking process is catered to that special audience as opposed to being flexible

UI:


Just add expenses, Splittr does the rest!

How did you spend your money? Analyze your expenses!

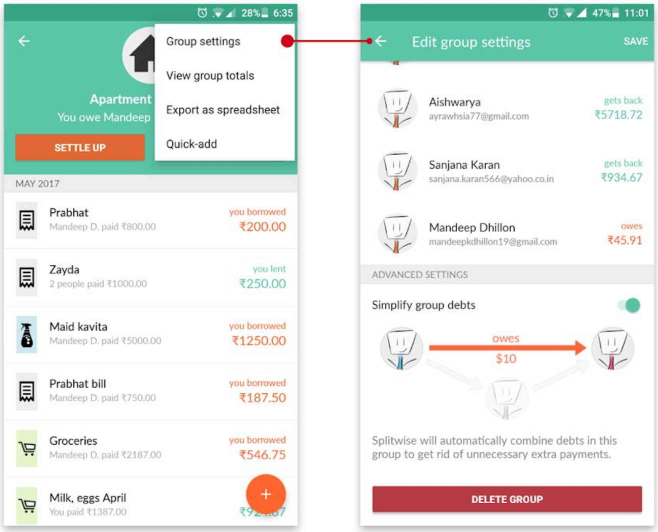
Who owes whom? Check off payments or send entire PDF reports to your friends!

Adding expenses is super easy!

Splitwise

- Good:
  - Free
  - Simple and focused on accounting for the user so that the math is simplified
- Bad / Areas of improvement:
  - No integration with payment apps
  - No photo documentation
  - Complicated analytics dashboard
  - The above three bullet points would ideally be addressed in our offering if we were to go to market

UI:


Apartment
You owe Mandeep

SETTLE UP

Group settings
View group totals
Export as spreadsheet
Quick-add

MAY 2017

Prabhat
Mandeep D. paid ₹800.00
you borrowed ₹200.00

Zayda
2 people paid ₹1000.00
you lent ₹250.00

Maid kavita
Mandeep D. paid ₹5000.00
you borrowed ₹1250.00

Prabhat bill
Mandeep D. paid ₹750.00
you borrowed ₹187.50

Groceries
Mandeep D. paid ₹2187.00
you borrowed ₹546.75

Milk, eggs April
You paid ₹1387.00
₹92[illegible].67

Edit group settings SAVE

Aishwarya
ayrawhsia77@gmail.com
gets back ₹5718.72

Sanjana Karan
sanjana.karan566@yahoo.co.in
gets back ₹934.67

Mandeep Dhillon
mandeepkdhillon19@gmail.com
owes ₹45.91

ADVANCED SETTINGS

Simplify group debts

owes $10

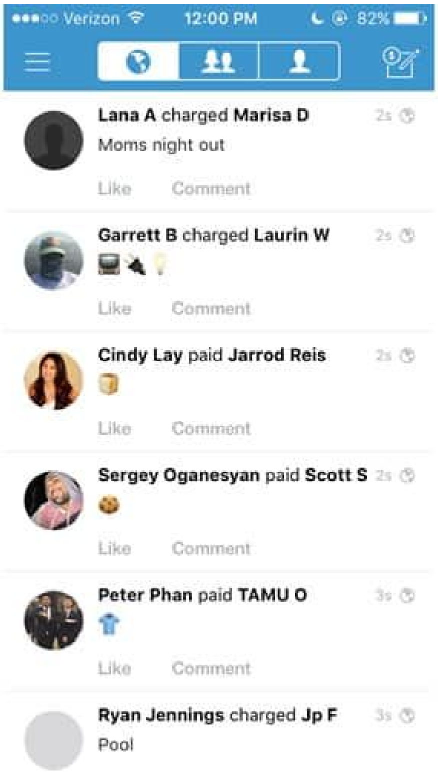
Splitwise will automatically combine debts in this group to get rid of unnecessary extra payments.

DELETE GROUP

Venmo

- Good:
  - Platform functions like social network which is very appealing for users
  - Instant payments
  - Good customer support user interface
  - Multiply payment support
- Bad / Areas of improvement:
  - Too many embedded fees for withdrawals or other payments
  - Too many ID and other checks that are not in flow of onboarding process
  - Cannot cancel transaction if both parties are not using Venmo (as to promote own product)
  - We would like to streamline the onboarding process and secure ID related issues on a regular cadence as opposed to randomly
  - We would definitely allow canceling of transactions regardless of what platform the party is using to increase usefulness
  - Information Transparent as the transaction will appear to contacts
  - Money transferred has to be calculated by users not convenient during bill splitting process

UI:



Lana A charged **Marisa D** 2s
Moms night out

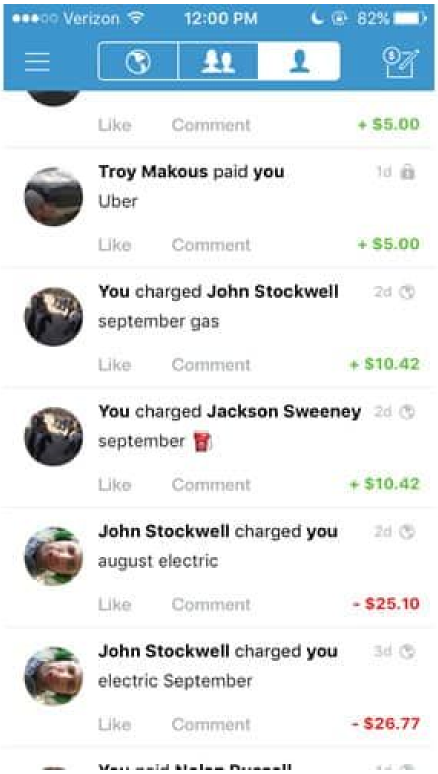
Garrett B charged **Laurin W** 2s

Cindy Lay paid **Jarrod Reis** 2s

Sergey Oganesyan paid **Scott S** 2s

Peter Phan paid **TAMU O** 3s

Ryan Jennings charged **Jp F** 3s
Pool



+ $5.00

Troy Makous paid **you** 1d
Uber
+ $5.00

You charged **John Stockwell** 2d
september gas
+ $10.42

You charged **Jackson Sweeney** 2d
september
+ $10.42

John Stockwell charged **you** 2d
august electric
- $25.10

John Stockwell charged **you** 3d
electric September
- $26.77

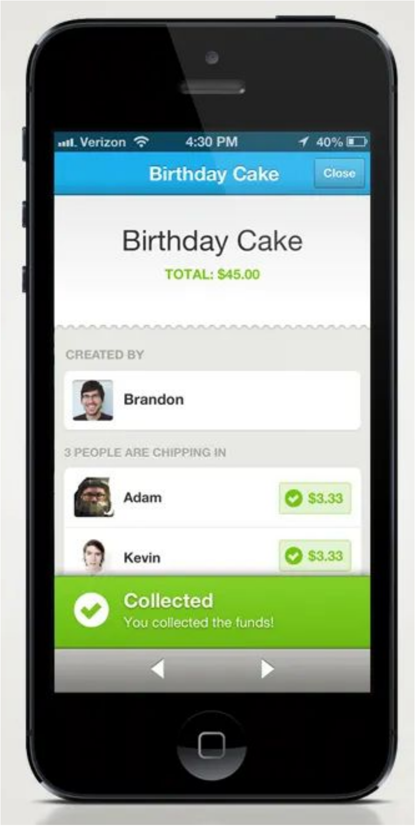
You paid **Nolan Russell**

GroupMe

- Good:
  - Easily accessible and pleasantly placed in the conversation

- Bad / Areas of improvement:
    - Hard to use as a service and very limited usability otherwise as well because this is a mere add on for a chatting platform, therefore the core focus is not on bill splitting. This reflects in the quality of the design which comes off rather undeveloped
    - Users would not use this regularly, only when they are planning something on GroupMe
    - Lacks analytics, editable lists of people, minimal integrations etc.

UI:



Birthday Cake

TOTAL: $45.00

CREATED BY

Brandon

3 PEOPLE ARE CHIPPING IN

Adam ✓ $3.33

Kevin ✓ $3.33

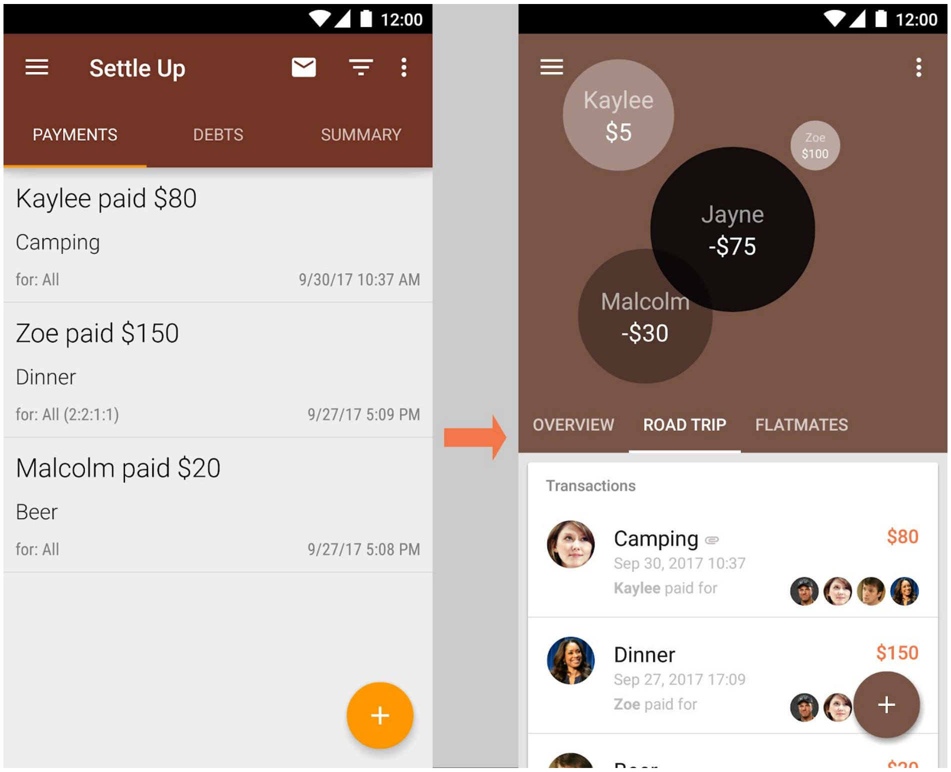
Collected

You collected the funds!

Settle Up

- Good:
    - Straight forward app for splitting expenses amongst groups, decent reviews (not very well known compared to some of the competitors)
- Bad / Areas of improvement:
    -

UI:



Kaylee paid $80
Camping
for: All 9/30/17 10:37 AM

Zoe paid $150
Dinner
for: All (2:2:1:1) 9/27/17 5:09 PM

Malcolm paid $20
Beer
for: All 9/27/17 5:08 PM

Camping $80
Sep 30, 2017 10:37
Kaylee paid for

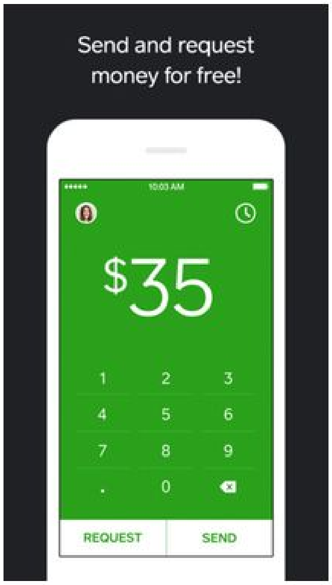
Dinner $150
Sep 27, 2017 17:09
Zoe paid for

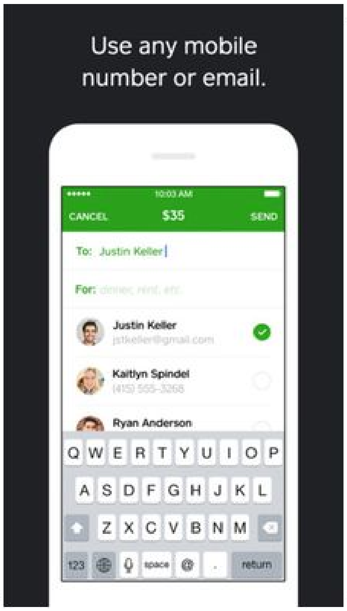
Beer $20

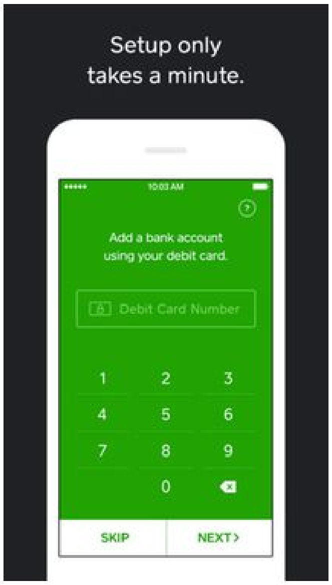
Square Cash

- Good:
  - Very easy to use, overall great add to the Square platform
- Bad / Areas of improvement:
  - Limited customer service visibility
  - Limited analytics as those efforts go more to the B2B side (as opposed to the D2C market)
  - Only one card linkable at a time - we would allow more than one card or bank to be linked ideally

UI:


Send and request money for free!

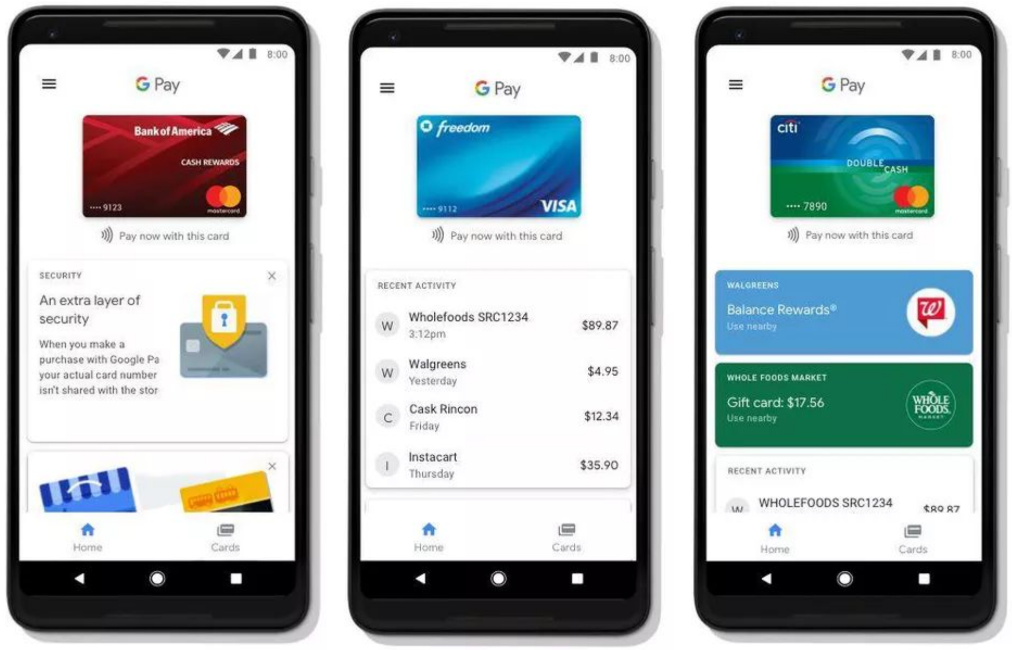

Use any mobile number or email.


Setup only takes a minute.

Google Pay

- Good:
  - Easy to set up and use
- Bad / Areas of improvement:
  - Heavily embedded deep within the Google family, which means like GroupMe, you will face pressure to start using additional services or integrations that also fall within the Google family (think Google docs, android phones etc.)
  - Not usable on all phones - there are user reviews that the quality of the offering may differ, or may be not usable at all if you sway away from traditional android
  - This instills some bias in the user interface which our offering will not have ideally

UI:

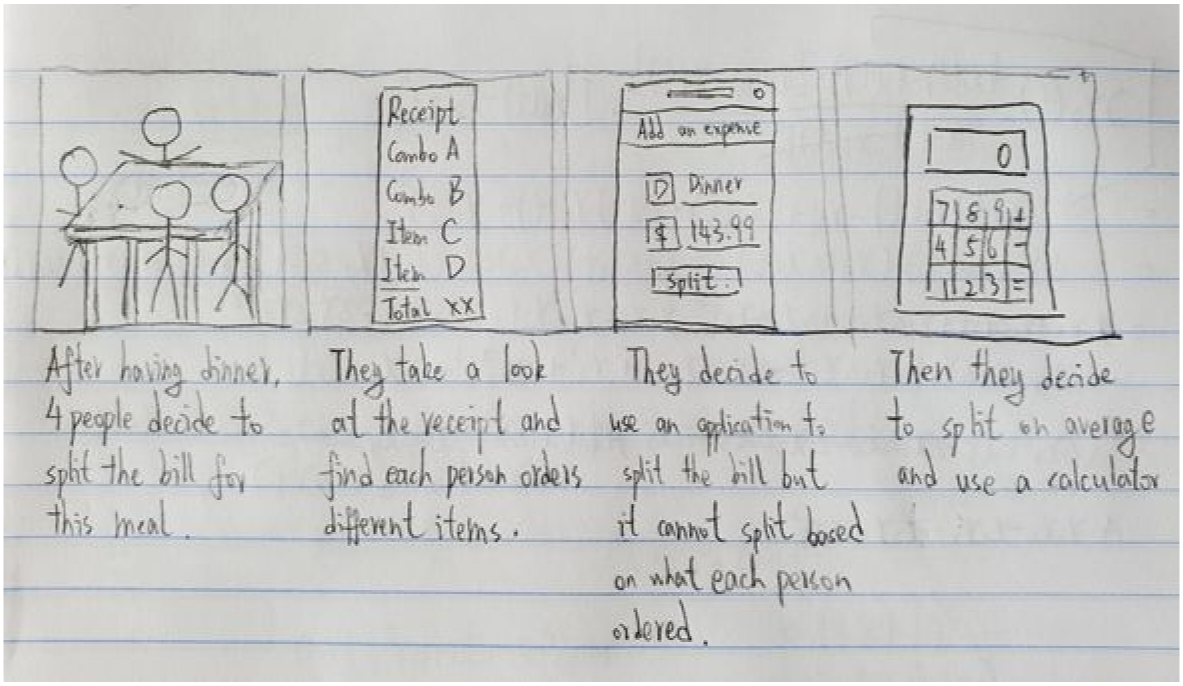

G Pay
Bank of America
CASH REWARDS
···· 9123
Pay now with this card
SECURITY
An extra layer of security
When you make a purchase with Google Pa your actual card number isn't shared with the stor
Home
Cards
freedom
···· 9112
VISA
RECENT ACTIVITY
Wholefoods SRC1234 3:12pm $89.87
Walgreens Yesterday $4.95
Cask Rincon Friday $12.34
Instacart Thursday $35.90
citi
DOUBLE CASH
···· 7890
WALGREENS
Balance Rewards®
Use nearby
WHOLE FOODS MARKET
Gift card: $17.56
Use nearby
WHOLE FOODS
WHOLEFOODS SRC1234 $89.87
8:00

6. Explanation of design solution: in the homepage, the user has two options, the first one is to start a new splitting, the second one is to participate in a splitting started by another user.

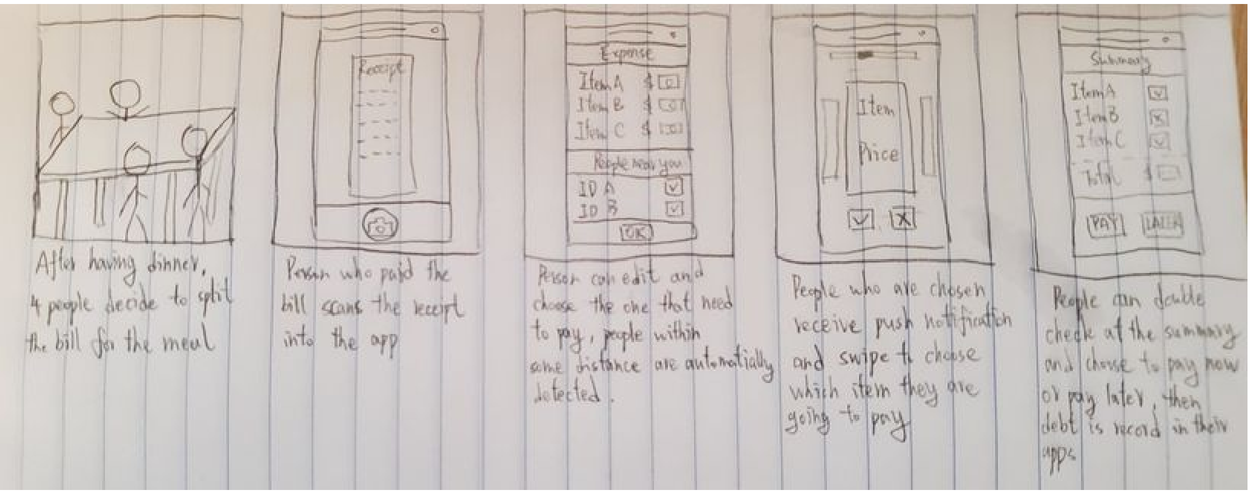
    a. The start button directs the user to the page of bill details. The user can choose to input manually or scan automatically. If the scanning result is not correct, the user is able to correct that directly. After recording the bill, the summary page appears and a unique code for the current splitting will be generated. The user has the option to send the code through social media apps.

    b. The participate button allows the user to type the invitation code. After entering the code, the items on the bill will be displayed on individual screens and the user can swipe between the screens and choose whether to pay for that item. After selection, a summary page appears for users to double check their selection. Users can also edit their selections on the summary page directly. Once the user makes sure everything is correct, they can choose to pay with their financial apps like Paypal and Venmo.

7. Storyboards:

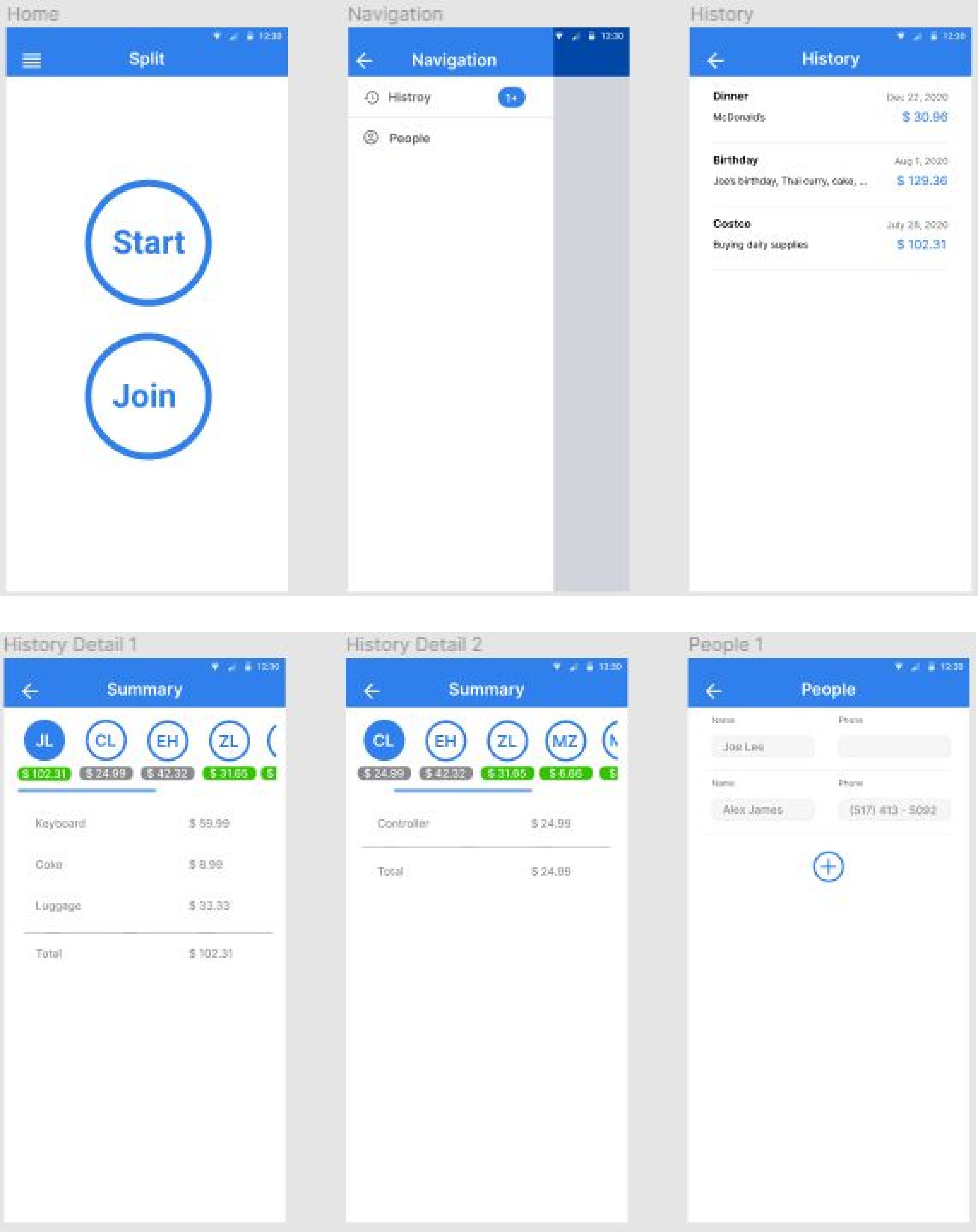

Receipt
Combo A
Combo B
Item C
Item D
Total XX
Add an expense
D Dinner
$ 143.99
Split
0
7 8 9 +
4 5 6 -
1 2 3 =
After having dinner, 4 people decide to split the bill for this meal.
They take a look at the receipt and find each person orders different items.
They decide to use an application to split the bill but it cannot split based on what each person ordered.
Then they decide to split on average and use a calculator

Storyboard of existing solution failure


Receipt
Expense
Item A $
Item B $
Item C $
People near you
ID A ☑
ID B ☑
OK
Item
Price
☑ ☒
Summary
Item A ☑
Item B ☒
Item C ☑
Total $
PAY
LATER
After having dinner, 4 people decide to split the bill for the meal
Person who paid the bill scans the receipt into the app
Person can edit and choose the one that need to pay, people within some distance are automatially detected.
People who are chosen receive push notification and swipe to chose which item they are going to pay
People can double check at the summary and chose to pay now or pay later, then debt is record in their apps

Storyboard of our design solution

8. Current prototype (Updated and final Figma layout (seperate from HTML file submission)):



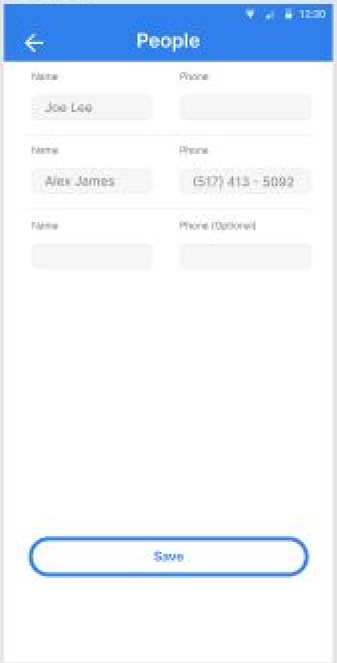
Dinner	Dec 22, 2020
McDonald's	$ 30.96
Birthday	Aug 1, 2020
Joe's birthday, Thai curry, cake, ...	$ 129.36
Costco	July 28, 2020
Buying daily supplies	$ 102.31

Keyboard	$ 59.99
Coke	$ 8.99
Luggage	$ 33.33
Total	$ 102.31

Controller	$ 24.99
Total	$ 24.99

Name	Phone
Joe Lee	
Alex James	(517) 413 - 5092

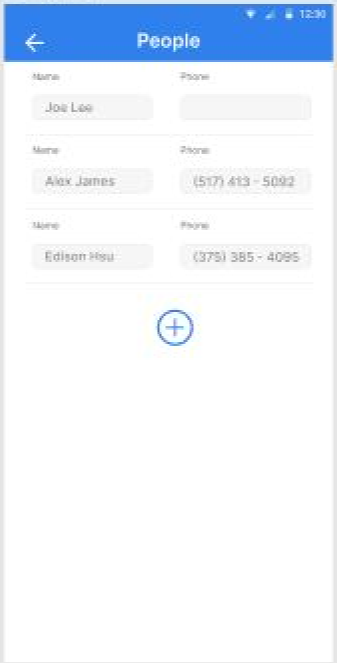
People 2



Name	Phone
Joe Lee	
Alex James	(517) 413 - 5092

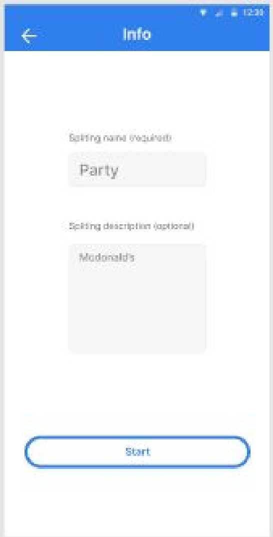
Name	Phone (Optional)

People 3



Name	Phone
Joe Lee	
Alex James	(517) 413 - 5092
Edison Hsu	(375) 385 - 4095

Info



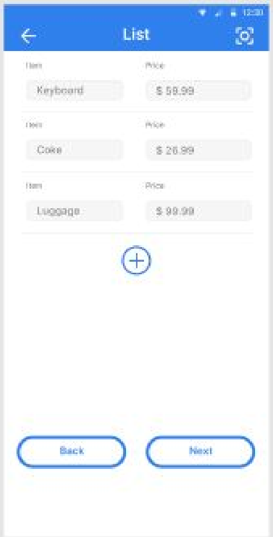
Spliting name (required)

Party

Spliting description (optional)

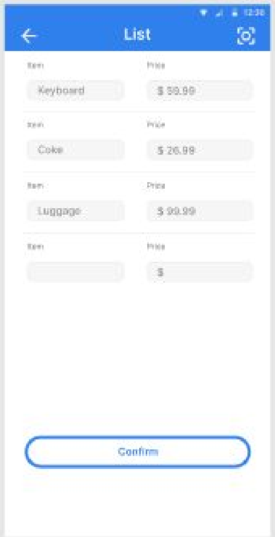
Mcdonald's

Item 1



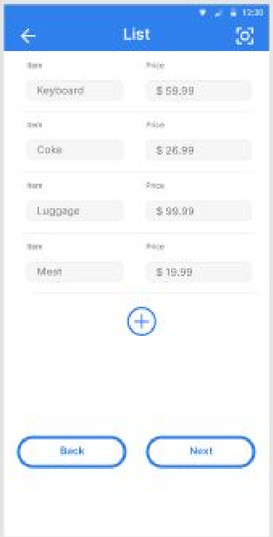
Item	Price
Keyboard	$ 59.99
Coke	$ 26.99
Luggage	$ 99.99

Item 2

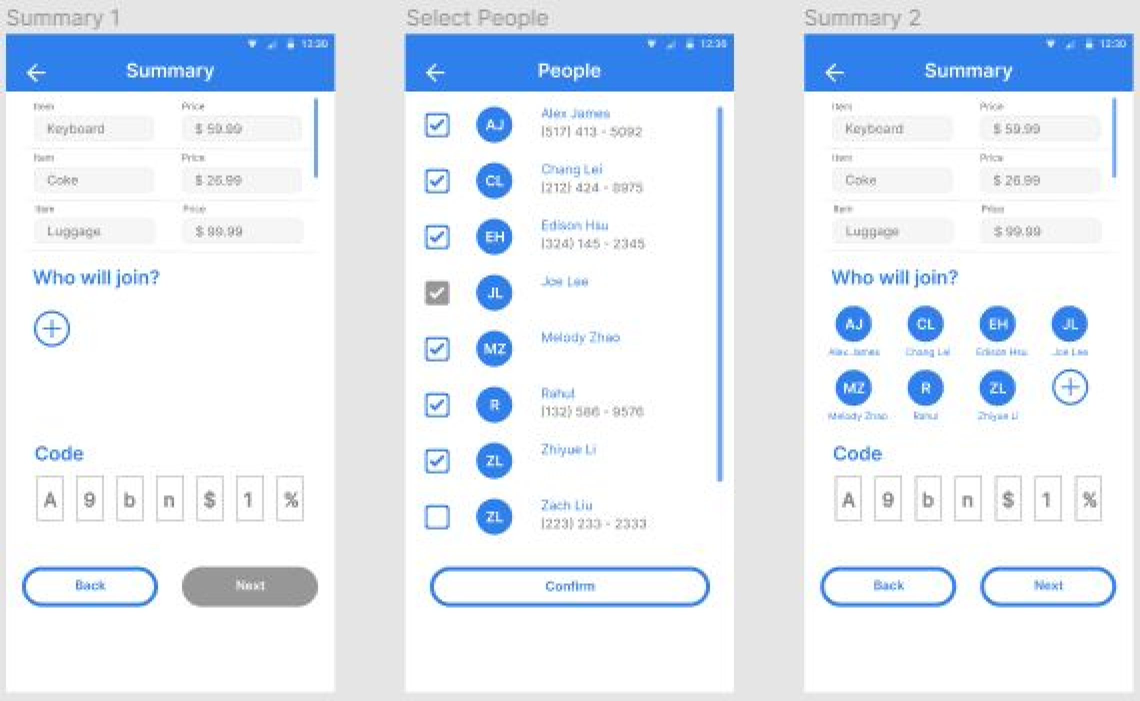


Item	Price
Keyboard	$ 59.99
Coke	$ 26.99
Luggage	$ 99.99
	$

Item 3



Item	Price
Keyboard	$ 58.99
Coke	$ 26.99
Luggage	$ 99.99
Meat	$ 19.99



Summary 1

Summary

Item	Price
Keyboard	$ 59.99
Coke	$ 26.99
Luggage	$ 99.99

Who will join?

Code

A 9 b n $ 1 %

Select People

People

☑ AJ Alex James (517) 413 - 5092

☑ CL Chang Lei (212) 424 - 8975

☑ EH Edison Hsu (324) 145 - 2345

☑ JL Joe Lee

☑ MZ Melody Zhao

☑ R Rahul (132) 586 - 9576

☑ ZL Zhiyue Li

□ ZL Zach Liu (223) 233 - 2333

Summary 2

Summary

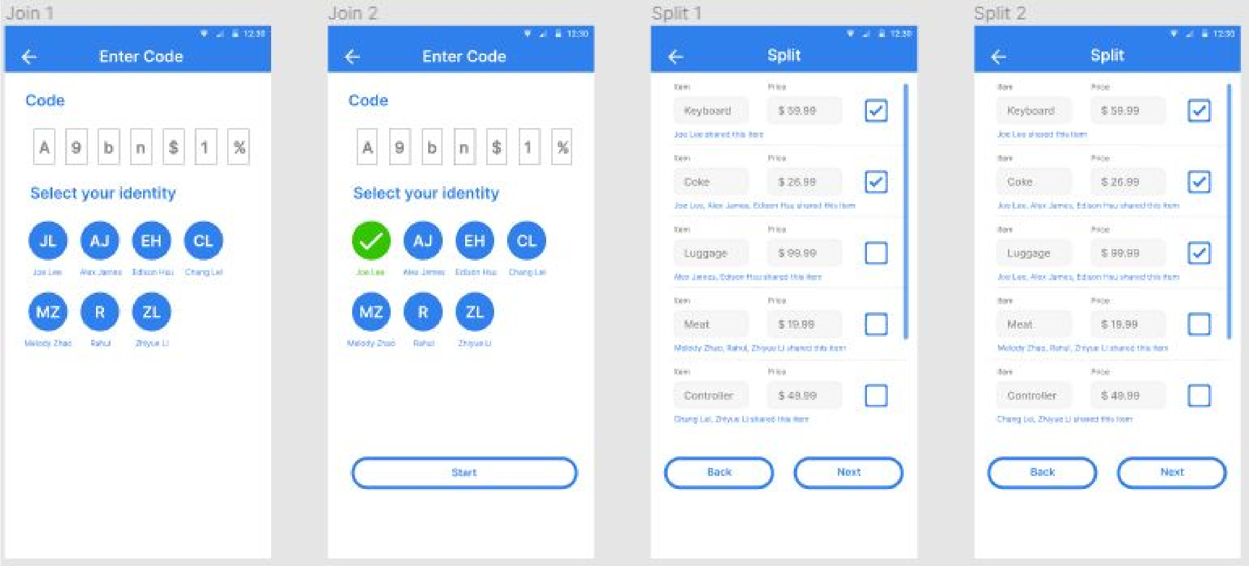
Item	Price
Keyboard	$ 59.99
Coke	$ 26.99
Luggage	$ 99.99

Who will join?

AJ Alex James, CL Chang Lei, EH Edison Hsu, JL Joe Lee, MZ Melody Zhao, R Rahul, ZL Zhiyue Li

Code

A 9 b n $ 1 %



Join 1

Enter Code

Code

A 9 b n $ 1 %

Select your identity

JL Joe Lee, AJ Alex James, EH Edison Hsu, CL Chang Lei, MZ Melody Zhao, R Rahul, ZL Zhiyue Li

Join 2

Enter Code

Code

A 9 b n $ 1 %

Select your identity

✓ Joe Lee, AJ Alex James, EH Edison Hsu, CL Chang Lei, MZ Melody Zhao, R Rahul, ZL Zhiyue Li

Split 1

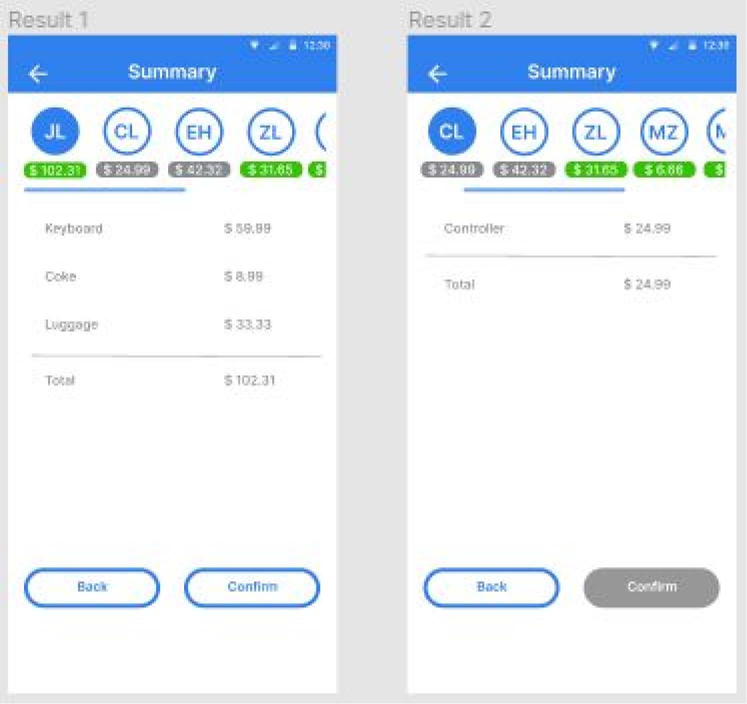
Split

Item	Price	
Keyboard	$ 59.99	☑
Joe Lee shared this item		
Coke	$ 26.99	☑
Joe Lee, Alex James, Edison Hsu shared this item		
Luggage	$ 99.99	□
Alex James, Edison Hsu shared this item		
Meat	$ 19.99	□
Melody Zhao, Rahul, Zhiyue Li shared this item		
Controller	$ 49.99	□
Chang Lei, Zhiyue Li shared this item		

Split 2

Split

Item	Price	
Keyboard	$ 59.99	☑
Joe Lee shared this item		
Coke	$ 26.99	☑
Joe Lee, Alex James, Edison Hsu shared this item		
Luggage	$ 99.99	☑
Joe Lee, Alex James, Edison Hsu shared this item		
Meat	$ 19.99	□
Melody Zhao, Rahul, Zhiyue Li shared this item		
Controller	$ 49.99	□
Chang Lei, Zhiyue Li shared this item		



Keyboard	$ 59.99
Coke	$ 8.99
Luggage	$ 33.33
Total	$ 102.31

Controller	$ 24.99
Total	$ 24.99

9. Risks to mitigate (Updated):

We think that the group has come a long way with additive and iterative feedback succession from IAs and potential users, and have covered bases such as streamlining the user journey, adding more optionality (effectively increasing user control over the process) and making it about par or better than other options out there by closely scrutinizing their pitfalls.
I think the risks that remain are largely around security and integrations. Our standalone app has done well to make the user journey very seamless, but we have not added onboarding and ID verification (amongst other verifications) which could potentially take a toll at the ease of use if not handled efficiently. We have seen this with other app reviews that users don't like repeated asks for documents, which causes delay in usability. This also plays into the security conversation as that is of utmost importance, and is underscored even more when taking potential integration capabilities into account.Internally, we imagine to frontload most of the checking so that we have an effective data lake that we can recycle until it becomes stale, which will require another refresh.

10. Project tasks:

Prototype UI design **Completed**

Estimate hours:	4
Due date:	11.16
Assigned member:	Entire team

High-fi UI design using Figma **Completed**

Estimate hours:	4
Due date:	11.20
Assigned member:	Zhiyue Li, Chang Lei

Digital UI design using HTML **Completed**

Estimate hours:	10
Due date:	12.1
Assigned member:	Zhiyue Li, Chang Lei

Javascript and interaction **Completed**

Estimate hours:	20
Due date:	12.16
Assigned member:	Zhiyue Li, Chang Lei

Design phase:

a. Get feedback from user **Completed**

Estimate hours:	30
Due date:	12.5
Assigned member:	Rahul, Melody Zhao

b. Improve based on feedbacks **Completed**

Estimate hours:	30
Due date:	12.12
Assigned member:	Rahul, Melody Zhao

Final UI design **Completed**

Estimate hours:	4
Due date:	12.16
Assigned member:	Entire Team

Write-up **Completed**

Estimate hours:	10
Due date:	12.20
Assigned member:	Rahul, Melody Zhao

Design Process

1. **Design process #1**:

   Design phase:

   Problem: Users want to see the detail of a splitting, for example, who shares which item.

   Key insight: Users need the ability to get the information they want quickly and clearly.

   Prototyping phase:

   First, users need to know who participates in a splitting. To achieve this, we added a new step before the swiping screens. Each one who participates in the splitting needs to pick up their identity from the list created by the person who starts the splitting. Of course they can undo selections if they pick up the wrong identities.

   Second, users need to know an item is shared by whom. To achieve this, we added a new area below each item detail that shows who is sharing the item.

   Evaluation phase:

   We evaluated our updated prototype by collecting feedback from users. From their feedback, we know that most of the users are satisfied with such feature. However, many of them also raised a new problem: It is time-consuming to create the list of people.

2. **Design process #2**:

   Design phase:

   Problem: Users think it’s time-consuming to create the list of people.

   Key insight: Duplicated operation should be avoided.

   Prototyping phase:

   To solve this problem, we first think of an idea that asks the user to create the list at the first time they use the application. After having such a predefined list stored in their application, they can create a new list by just selecting people from this predefined list.

   However, we soon realized that not everyone needs to create such a list at the first time using the application. In real life, it is common that only several people pay the bill, and others pay them instead. It means that this step seems useless to many users. As a result, we added a new screen before that asks the user whether they are the ones who always pay the bill in real life. Users are asked to create the list only if they answered yes.

Evaluation phase:

We evaluated our updated prototype by collecting feedback from users. From their feedback, we know that they liked this feature, but they also asked whether they can edit the predefined list easily. In order to achieve this, we decided to add the list to the side navigation bar.

3. **Design process #3**:

- Design phase:

  Problem: Users want to see the history of splitting.

  Key insight: Applications need to store the using history and users need the ability to see and look up for specific records in the history.

- Prototyping phase:

  To solve this problem, we need to implement two things: store the splitting history and create a new screen that shows the history.

  In order to make it easy and quick for users to see and look up records in the history screen, we added the history screen to the side navigation bar and a search bar at the top of the history screen.

- Evaluation phase:

  We evaluated our updated prototype by collecting feedback from users. From their feedback, we know that they think the history feature is essential. It is necessary to have the history because our application is related to money. The history feature makes it possible to double-check previous splittings and see whether there were mistakes.

Progress Report 1 feedback follow-up

1. Project concept is way too vague - how will you make it easy/fast for people to split the bill? What would make it so people choose your application over existing ones?

- Response: We make it easy for people to split the bill by removing redundant features that are unnecessary, for example, accounts. The burden of bills verification and payment management is mostly carried by the actual payer. Participants only need a few clicks to know the debt. The amount of money a user should pay is accurately calculated based on the specific items that the user claims.

2. Not too ambitious as this is a current problem that has many diverse solutions for (think venmo, tricount, paypal, manual, etc.) that could work almost as well as what you are doing. Not too original either because many existing solutions are aiming for the same user group and solution as well - is there a way to improve the ambition of your topic here?

- Response: Current solutions invariably require users to register an account that is linked to their financial applications. We save the trouble of creating a new account for most users. At the same time, the account function is kept for organizers who actually pay the bills.

3. Comparative Analysis - very weak, what you said is not working in your first and second example is something you carry on into your solution, there isn't much differentiation from that. Plus, you do not utilize the parts that are doing well to the best of their advantage either.

- Response: We've updated that section.

4. How will you input manual details or scan your receipt? How will this be implemented? How is this a reduction of work for the user if they still have to manually enter their details? How can you ensure that everyone is inputting the correct data if everyone needs to finish at the same time? How can your app catch and correct errors with splitting the bill? Seems time-consuming/ redundant possibly. I like the invitation code, but there needs to be security/trust measures implemented between users, especially involving money.

- Response: The scanning feature is designed to reduce user's work. It can be implemented easily because there are already many existing image identification solutions. However, it's apparent that the scanning feature cannot apply to all kinds of receipts, let alone the bills that don't have receipts, this is why manually input is necessary. If the scanning result is incorrect, the user can correct that easily. If the user makes a mistake himself, it is inevitable because everyone makes mistakes. We've already made it super easy for users to make corrections on the summary screen. To make the invitation code more secure, we decided to make the code more complicated, involving letters and characters. This will prevent people from joining stranger's splitting.

5. There is mention and acknowledgement of some risks that you weren't able to address but not much in the implementation and security risks of finding people nearby, or the tediousness of

going through every item on the receipt one by one. It is not just the functions might not be correct, but how to ensure that your current functions are handling your users the best they can. Risks include financial safety, connections with PayPal and Venmo like applications, how to resolve conflicts between users splitting the bill with different perspectives on what they owe, how to get users to use your application over other similar apps.

- Response: we abandoned the “finding people nearby” feature and replaced it with invitation code. The organizer/actual payer will verify the bill so that participants only need to claim the items they owe. For a specific item, the amount of money is divided evenly across all the users who claim it. These risks have been mitigated.

6. Nice design! I’d recommend allocating to user testing and upgrading of the design rather than the functionality if you are aiming for a Figma final deliverable.

   Response: We’ve updated that section.

7. I need more specifics in each of your sections + more information on how your user research/testing informs your design decisions. I can’t guess on these parts.

   Response: We’ve updated that section.

1. Code Implementation Details

   We used Figma to convert components of our prototype into SVG codes and put them in the HTML files, which makes our implementation highly consistent with the prototype. Furthermore, we added simple javascripts to enable basic interactions. No other third-party code, library, or resource was used.

2. Instructions to Run Your Program

   For an organizer or actual payer, click "Start" on the home screen to begin the process of creating a bill splitting event. For participants, click "Join" to begin the process of splitting. Users can add people to their contacts and see past bill splitting events respectively through "People" and "History" in the navigation bar (upper-left corner on home screen).

3. Progress Report 2 Feedback Follow Up

   The feedback we received from Progress Report 2 was from TA Karen. It was actually quite similar to the one received from Progress Report 1. We met as a group and identified that the reason for this was because our skeleton of the prototype remained the same, while the changes we received from user and IA feedback were reflected in iterations of the implementation piece, which came much later. I think that the reinforced feedback we received was on the following fronts -

   - Comparative analysis
   - Outlining, defining and securing the user journey
   - User testing feedback

   We internalized this feedback and performed the following actions - we regrouped to map out the space of payment splitting to better understand the competitive landscape and the other designs out there from both incumbents and new-gen companies. Not only did this exercise help us better understand the solutions that exist out there, but it also helped us identify what these designs did well and where they lacked. We were able to do this by both testing these solutions as well as synthesizing user feedback across platforms and channels. The second thing we did, which will reflect in our video submission, is understand how to define the user journey in a way that this can be used by kids in elementary school, so hopefully they too can be trusted with this. Speaking of trust, we took time to understand potential security leaks that can take place (to the best of our ability as fintech has many layers of security infrastructure that are quite complex from our conversations with industry participants). Finally, we were able to get additive testing and feedback from both the old set of users as well as try our solution on new users as well. I think we have come a long way in making the solution seamless, time efficient and easy to navigate. The new feedback that we received was that they want analytics and integrations. This is obviously heavy duty backend work, but their feedback came from a place of wanting to optimize their spending and budget. Further, while they like the solution, there is a sense of inertia that comes from older solutions (think Venmo, Zelle) and integration capabilities by the solutions banks provide. To tackle this, and something also included in the post-mortem, the next step would be to advance the solution in a way that there are sections in the application where the user can see their past transfers along with how that looks on a monthly and annual basis. Further, if this was mapped against expected salary (which could potentially just be inputted by the user), usefulness could be increased by providing visual depictions of personal finance management. Those were the steps our team took to better incorporate the feedback received.

4. Post Mortems

Rahul Sehrawat (rs3688)

After completing this project, I certainly had several takeaways on what went right, what went wrong and perhaps what I would have done differently. Starting with the good, I think the group did a very good job in internalizing and agreeing on an idea and distributing the work for efficient project management. I also really liked that we were able to repeatedly get additive and iterative feedback from potential users (both old and new as we did more design cycle reps) which allowed us to further improve our product. Another thing we did well was map out the competitive landscape (albeit slowly) to understand all the other major designs that exist out there and what people think about them. With this analysis, we were able to direct our focus on scrutinizing the various pitfalls different UI has to come up with something that is hopefully better than other solutions. When it comes to stuff that did not go well, I wish that we could have gotten even more reps because we are tackling a problem that already exists out there and those solution providers have optimized over several feedback loops, something I wish we could have done more. I think that would have further improved our design and made it even more usable by a larger audience for even more use cases. Another thing that we could have improved on is maybe picking a slightly lesser solved problem or identifying a outlined niche. This would have helped us tackle something where there is whitespace opportunity instead of going against the greats (PayPal, Google) to provide a better UI design - it was really a question of competition really, I think we were slightly too ambitious on that front. I think those are the primary things that went well, wrong and speak to what would have been done differently from our group if we were to undertake this project again!

Zhiyue Li (zl2966)

From my perspective, most of the project went well but we had a difficult time figuring how to make our project more ambitious. I think we did a good job on the Figma prototype. I tried my best to make its theme to be simplified and uniform. I also make the animation between different screens to be very similar to a real mobile application. For the HTML implementation, I used a little trick which is exporting components as SVGs. I had to do such things because the time is kind of tight during finals week. However, the advantage of doing such a thing is that our HTML implementation is highly uniform with our Figma prototype.

Just as I said before, we used to be stuck in making our project more ambitious. During TA meetings, Karin told us our project felt like an advanced calculator. As a result, we spent lots of time adding new features to our application. I think we did make it more ambitious at last.

If I were to undertake this project again, the most important thing is to start early. We wasted a lot of time on this one and thus, we had a number of ideas that were not able to be realized yet. Another thing is to do more research before making the decision of what to do for our project. I didn't do enough research this time and as a result, I was not familiar with editing applications. I couldn't do a strong comparative analysis as a result.

Chang Lei (cl3910)

We paid too much attention to the details of our high fidelity prototype. Although this made our design look good aesthetically, we were also prone to overlook the overall usability and usefulness. We should have made some low fidelity prototypes and conducted more user studies before we began to refine the high fidelity one.

Melody Zhao (xz2996)

In the beginning of the design process, our team is efficient to decide the project topic as all of us do not use bill splitting apps so the target population we discovered is distinct from us and we feel the idea is ambitious. The initial design comes from group member ZhiYue, and the first problem we faced is that currently there are many apps that exist like Venmo and Splitwise, and they are sophisticated enough to cover all the needs of the users we thought. What we did well here is we tried the current bill splitting apps like Venmo and Splitwise and listed their positives and negatives. After interviewing the target users about their ideas of those existing solutions, we discovered that those existing solutions are too complicated for sign up. Usually when users want to split, they hope the process is fast and clean. Therefore, the most distinction of our design is that we avoid abundant features like creating accounts.

In regard to what could be improved in our design, I feel the idea that the code generated by the organizer and then the users put the code to join the splitting event is still not ideal as I thought. As I interviewed some users, some of them felt hesitant to type code especially if the code is complicated, then it takes longer for the overall splitting process. However, we have not designed a better solution than sharing code. We have tried sharing links instead of sharing code, when users click the link, they will automatically join the event, but sharing links requires more time as the organizer needs to do multiply times.