



Redesign Project

Bill Splitting App

Vishnu Priyan K
16U130007



Designing a System/App for Smooth Payment and Splitting Experience in Restaurants

Going out for dinner is a fun and fondly remembered experience, with your loved ones or with your buddies. It's all fun and games up until the check shows up, bringing stress and math problems with it. This process has barely changed in a very long time. We have dinner, get the bill, and depending on the dynamics of the table, the bill either gets paid by one person, split into equal portions, or split according to what you had to eat, with the complexity of the math involved increasing in the same sequence.

This specific problem deserves a higher level of attention and a solution which solves it satisfactorily, due to the amount of time which the guest/waiter has to spend splitting the bill, which could add up into a large amount of time for the restaurant where the tables are occupied, when someone else could have been served.



Design Process:

User Research:

The aim of this round of research was to identify the problematic scenarios faced by people in restaurants when splitting bills, why is it an issue, and the tools to help them out.



Most common problematic scenarios:

- *People fall into two categories — even split, and pay your way. And people pay their way in lots of different ways.*
- *Talking about money with friends and figuring out how to split the check causes a lot of social anxiety.*
- *Regardless of how the bill is split, people worry about paying way more than their share.*
- *There are hidden costs to an uneven split. No one likes math at the end of a meal.*
- *Knowing how the bill will be handled in advance of a meal brings peace of mind.*
- *Even when splitting the check, not everyone invited is expected to pay: People want to have the ability to treat other members of their party, and allow other members to treat guests as well.*
- *Sometimes people prefer not to split the check — and some people always want to pay.*
- *Restaurants have an upper limit on the number of separate checks and card swipes or sometimes don't do them at all.*



Most common solutions:

- *Splitwise*
- *Google Pay*
- *Calculator*
- *If too much work needs to go into it, just split equally (Not fair always)*



Secondary Research:

There are a lot of apps and websites that help you split expenses more broadly like Venmo, Splitwise, Splid and specific ones like Tab, Billr.

The only aspect working properly for the apps is the possibility to keep track of the expenditure and how much people owe you. Other apps which facilitate on the spot splitting at the moment lack the ability to store this data of who owes whom how much.

The most common and preferred way to split bills even when so many apps are available, is still surprisingly manual calculations on the spot.

Another way that people use to get around this problem is by handling the whole transaction in cash, that way you have an upper cap and can choose to not dish out the extra bucks.



Anyone who has ever purposely ordered the cheapest item on the menu – or, barring that experience, anyone who has ever seen the Friends episode “The One With Five Steaks And One Eggplant” – knows what happens next. The people who ordered the steak or extra glasses of wine happily agree to an even split; those who stuck to water and appetizers eye each other warily before someone quotes Phoebe Buffay: “Uh-uh. No way. Cold cucumber mush for thirty-something bucks?”

“Splitwise would get annoying when a small amount someone owes you, or which you owe someone else shows and it says it’s outstanding”.

“Transactions of immaterial amounts back and forth”

“Requires a lot of manual input and planning, making it quite an inconvenient task during what is supposed to be a fun night out.”



Establishing Personas:

Establishing Personas early in the design process is useful, so that the steps forward can be validated from the perspective of the User.

User Persona: Vishal Venkat

Vishal likes to go out to new and different places with his friends. He is constantly bothered that he either has to pay for more than what he ate, or he has to sit and calculate the accurate amounts for everyone when splitting the bill.

Vishal does not want to waste time and he doesn't want a relaxing and fun night to end with stress. He likes the experience, but doesn't want it to leave a bad taste.

He sometimes skips out when he has to go to dinner with too many people or if certain people will show up who are not nice.



Ideation:

Booking tables and Inviting Guests:

One person can assume the role of planning the place and making a table booking, and based on the number of people responding to invites sent via the app, the final decision can be made.

Ability to look at recommendations in the app for that restaurant and decide before-hand, your preferences if need be:

Everyone who has accepted the invite from your dinner-master can look at the menu and decide what they want to eat based on recommendations, if they please.

Personalised orders and bill directly to the app on your phone, with history of dinners and participants:

You can pay directly from the app, using credit/debit cards stored on the app or the option to use UPI, or simply register as a cash payment and get done with it, and get a comprehensive history of dinners and payments through the app.



Taking it back to the user:

A quick round of validation of ideas with users, resulted in valuable feedback that helped define the solution even more. The feedback was also integrated into the system.

Scope Definition:

The scope confirms what has been ideated and keeps context as we move forward on the information Architecture and User Flows.



Introduction:

The app helps decrease the complexity of the whole process after we have received the dinner check, and serves some bonus purposes like rebooking tables, payments and payment history.

The ideal outcome would be a system where everything about a night out is centralised, and happens automatically. That's what this solution points towards.



System Description:

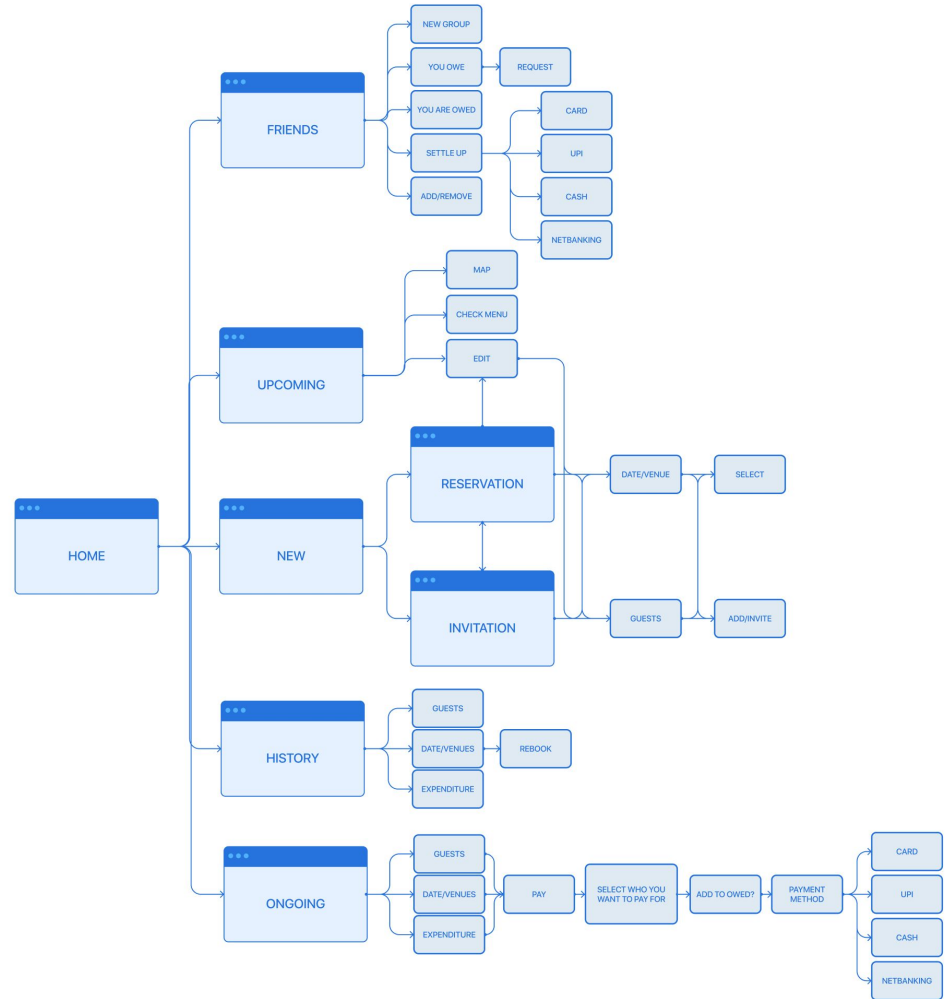
The Platform houses both users and restaurants to create a network of time-efficient and planned reservations without the burden of calculating amounts to be paid, resulting in more diners per night for restaurants and less time spent stressing for customers. Users can leave ratings also in the platform, which serves as an additional data point for reference.

Security:

As the app will have access to the spending habits of the users, and their credit/debit cards and UPI, privacy must be a top priority. Ideally TouchID or FaceID lock for the app as well.

Information Architecture:

With the scope in place, it was simpler to come up with the different cards denoting different data structures in the app. They were roughly categorised based on the card sorting technique by a few users.





Wireframing:

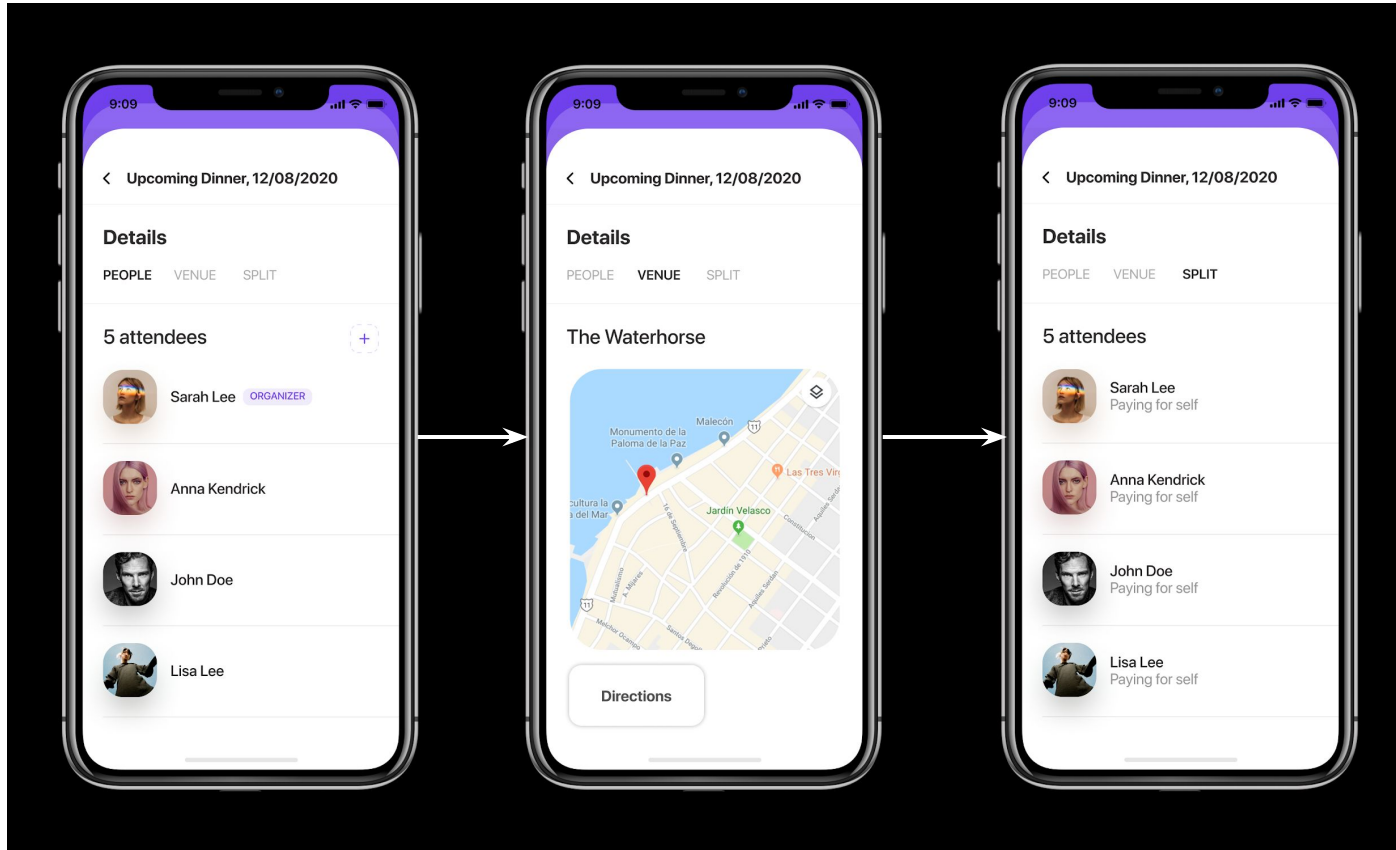
After a lot of iterations, the wireframes on paper were finalised. These were very low fidelity and dealt mostly with how data was presented.

Visual Design Patterns:

Having consistent design language in place is important. I have put together a quick style guide before starting out designing the surface of the app. I've used a Fibonacci series to define the font sizes, as well as, the layout spacing. This ensures a highly consistent design experience.

User Flows:

The Flow to check the details of an upcoming reservation:



User Flows:

The Flow to pay at the end of a Dinner:

