Idea Whiteboard

App features 100

- Create a trip
  - add trip name
  - add trip members
  - trip id (auto generated)
- Add members to trip
- Add expense in a trip
  - add expense
  - expense id (auto generated)
  - add expense name
- Update expense in a trip
  - ⇒ only lender can update
- Delete expense in a trip
  - ⇒ only lender can delete

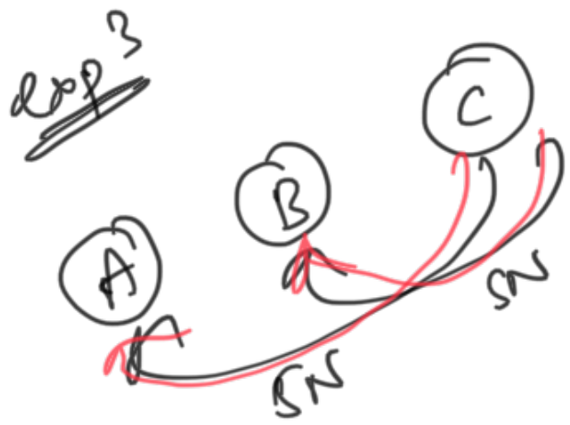
- Settle with a member in a trip {v2}
  - ⇒ can be added as another expense with expense name as "settle"
  - ⇒ actual transfer of tokens to lender
  - → settle amount ≤ owed amount
- Settle with a member across trips {v2}

✱ Types of expenses

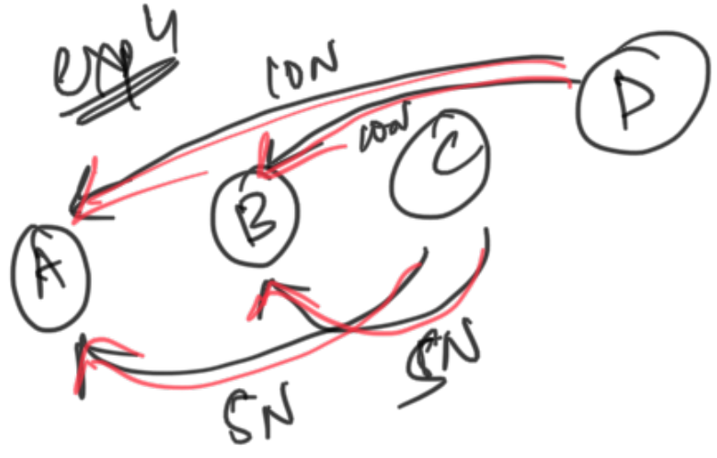
Currently
- Complex lending can be broken down into a batch of simple expense txns

eg:- exp 4 can be
- A owes C 5N
- B owes C 5N

- A owes D 10N
- B owes D 10N



- The simplification logic can be done at frontend
- $ to N conversion will also be done at frontend, if needed



- Currently, all simple expenses in a complex txn will have diff. expense ids; The DS can be changed in V2.0 to support complex expense as one

✲ Frontend Flow

① Acc login

② My Trips

Login with Near

Trip 1
Trip 2
Trip 3
⋮
+ Add

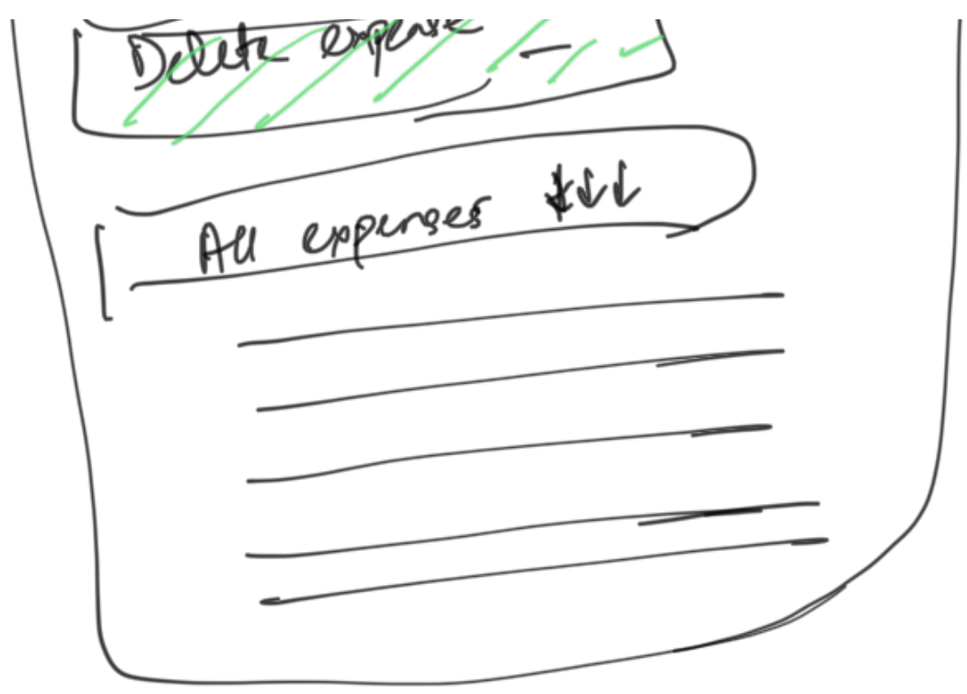
③ View Trip

Trip # 1
Members
A B C D
Owed Owe
$ 1000
Owed/Owe Breakdown
A +$100
B -$00
Settle

Add expense +

④ Add Trip

Trip # +
Members +



★ Contract Stuct

struct ExpenseTracker {

- try_id_by_owner

- trip_metadata_by_trip_id
  - > trip members
  - > trip name
- trip_expenses_by_tripid
  - > txns added seq. with txn_id
  - > new txns can be added to trip_id map with inc txn_id
  - > delete/update txns in a tripid map with txn_id

✲ Methods on Contract 

- new

- add_trip
- view_trip_metadata_by_trip_id

- add_trip_members
- view_trip_id_by_account_id

- add_trip_expense
- update_trip_expense
- delete_trip_expense
- view_trip_expense_ids_by_trip_id
- view_trip_expense_by_expense_id

- get_expense_summary_by_trip_id_account_id
- get_total_expense_summary_account_id {v2}

- settle_expense_by_trip_id_account_id {v2}
- settle_total_expense_account_id {v2}

✨ v2 roadmap 

- build frontend
- additional impl methods
- code optimization → reduce contract costs wherever possible
  → integration testing
  → add timestamps
  → valid account ids
  → ...more...
- frontend ability to register expenses in ₺ or N (saved in N or blockchain)
- push to mainnet