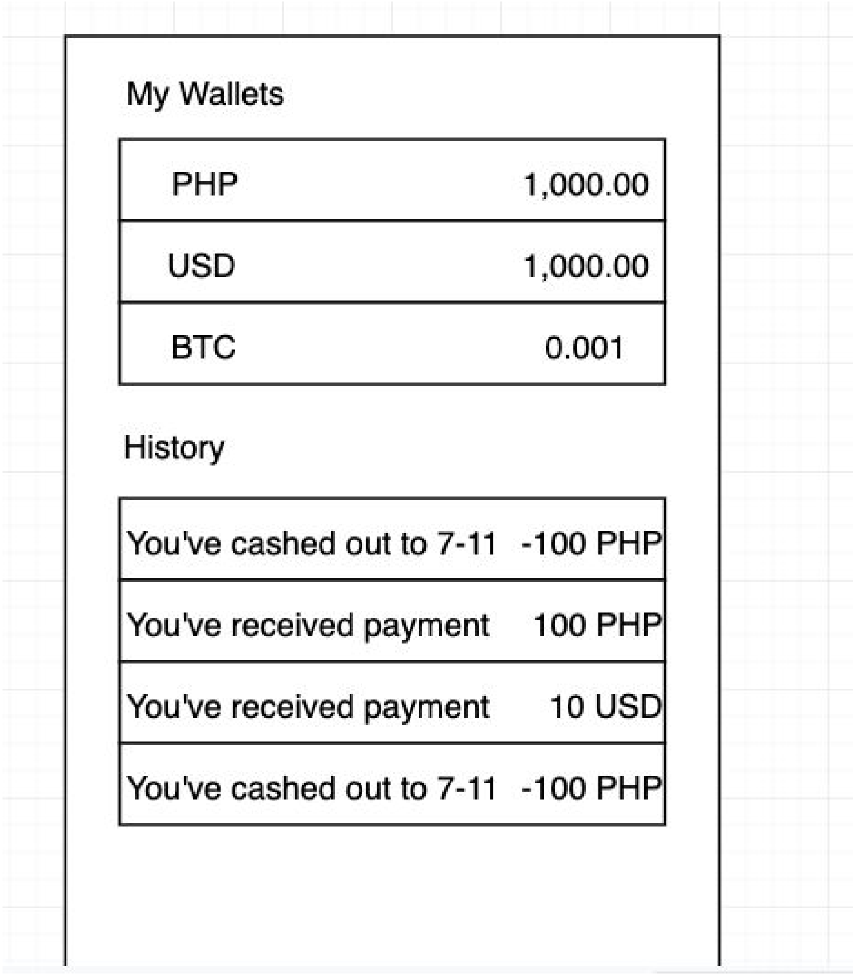
Coding assignment

Please develop a functional mobile app demo which:
1. Displays two lists on the same screen
2. The first list is a list of wallets the user has
3. The second list is a list of transactions the user has performed

Please display the information following details under the **API** section

Wireframe



My Wallets

PHP	1,000.00
USD	1,000.00
BTC	0.001

History

You've cashed out to 7-11	-100 PHP
You've received payment	100 PHP
You've received payment	10 USD
You've cashed out to 7-11	-100 PHP

API

You may implement the API using a mock (e.g. mockable.io, amock.io), server-side implementation is not a requirement. You may provide alternate assumptions/implementations to us in this regard.

The API requests and responses provided below are a rough guideline, you are welcomed to introduce changes to them, as long as proper reasonings are given :

1. List of wallets:

Request:
GET `/wallets`

Response examples:

Status: 200	{"wallets": [
	{
	"id": "1001",
	"wallet_name": "PHP",
	"balance": "1000.23"
	},
	{
	"id": "1002",
	"wallet_name": "USD",
	"balance": "100.10"
	},
	{
	"id": "1003",
	"wallet_name": "ETH",
	"balance": "0.000000000000010026"
	}
	]
	}

Status: 429	{
	"errors": {
	"non_field_errors": {
	"code": "throttled",
	"message": "Request has been throttled"

	} } }
Status: 500	Server error

2. List of transactions

Request:
GET `/histories`

Response examples:

Status: 200	{"histories": [
	{
	"id": "2001",
	"entry": "incoming",
	"amount": "100.23",
	"currency": "PHP",
	"sender": "John",
	"recipient": "You"
	},
	{
	"id": "2000",
	"entry": "outgoing",
	"amount": "10.10",
	"currency": "USD",
	"sender": "You",
	"recipient": "7-11"
	},
	{
	"id": "1999",
	"entry": "incoming",
	"amount": "0.000000000000000105",
	"currency": "ETH",
	"sender": "Keith",
	"recipient": "You"
	}
	]
	}

Status: 429	{

	"errors": {
	"non_field_errors": {
	"code": "throttled",
	"message": "Request has been throttled"
	}
	}
	}

Status: 500	Server error

Requirements

- We should be able to easily get your application up and running in our environments. Be sure to provide any notes to assist with this
- Please allow showcase of error state, and/or mixture of states for the two lists
- Please allow showcase of loading state, if needed, delaying handling of response by few seconds is appreciated
- Please provide us with any assumptions/alternatives made
- Challenge is for few evenings, there is no strict time limit, but we recommend a maximum of 3 evenings

Things we're interested in:

- Proven expertise in your tools of choice
- Ability to take charge of a project with minimal guidance
- Care is taken to test and budget for various error cases / scenarios
- Well written and useful documentation, if any required