

Cash Machine

- Cash Machine handles incomes of money.
- It can work with different sources of money: Cash, Credit Card, Bank Transfer
- It has a limit of 20.000 amount for total processing, everything more is declined.

Cash Source:

- It can accept only banknotes of 1, 5, 10, 50, 100 for Cash source
- It has 5 inputs for quantity of each type of banknotes
- It has a limit of 10.000 of amount in Cash, everything more is declined

Credit Card Source:

- It takes as inputs: Card Number, Expiration Date (MM/YYYY), Cardholder, CVV (3 digits), Amount
- It can accept Card Number with 16 digits and only ones which starts with digit '4' (like 4123 4567 8912 3456)
- Expiration Date must be at least 2 months bigger than current month

Bank Transfer

- It takes as inputs: Transfer Date, Customer name, Account number (6 alphanumeric), Amount
- Transfer Date can't be older than current date

Implementation

- Must be implemented 3 separate forms with inputs for each Source of Money
- All inputs are required
- On form submit the transactions must be stored in Database (total amount and inputs as JSON)
- All validations must be written using Laravel Validator, maybe some custom rules need to be written
- Cash Machine must check on transaction submit if amount limit isn't exceeded by calculating total amount stored in Database
- On successful submission User must be redirected on a New Page with transaction details stored in Database (ID, Total, Inputs)

Requirements

- Each source must be implemented as a separate class which will implement a common interface:

```
interface Transaction
{
    /**
     * Validate Inputs
     */
    public function validate();

    /**
     * Return total amount
     */
}
```

```

    */
    public function amount();

    /**
     * Return Inputs
     */
    public function inputs();
}

```

class CashTransaction implements Transaction {} class CardTransaction implements Transaction {} class BankTransaction implements Transaction {}

- Cash Machine must be a class which will handle Transaction and will validate and store it in Database

```

```php
class CashMachine
{
 /**
 * Store transaction in Database
 */
 public function store(Transaction $transaction)
 {
 //
 }
}

```

- Must be implemented a factory for Transaction initialization

```
$transaction = TransactionFactory::make(CashTransaction::class, $request);
```

- Code Style PSR-12, Best Practices