

要件定義書

- 要件定義書
 - 目的
 - サービス要件
 - 機能概要
 - 機能要件
 - 支出CRUD機能
 - 支出登録機能
 - 支出個別参照機能
 - 支出参照機能
 - 支出更新機能
 - 支出削除機能
 - 支出合計参照機能
 - 支払額集計比率変更機能
 - 支払額集計通知機能
 - データ定期削除機能
 - 非機能要件
 - システム方式・構成
 - アプリケーションPod
 - データベースPod
 - データ定期削除Pod
 - 支払額集計Pod
 - 支払額通知Pod
 - 利用者一覧
 - 外部インターフェース
 - 情報・データ

目的

2人暮らしとなると、生活費をどう折半するかや、相手に支払いを催促するのも、なんだか気が引ける。。こんな面倒なことはシステムで自動化しちゃおうという発想。

今回は、各々が購入して発生する費用に対して、支払いが少ない方がどれくらい支払えば良いかを自動計算して、支払いの催促を通知するシステム。(以降、生活費自動計算通知システム、もしくは本システムと呼称)を構築してみ、自身のエンジニアリングスキル向上と同棲生活のちょっとした不便の解決を目指す。

今回は生活費を、光熱費と共有品購入費(食材等の生活をする上で共有で使用・消費するもの)で発生するものとする。

サービス要件

- 光熱費・共有品購入費について、各々が支払った金額を月末に集計をして、追加の支払いが必要なユーザにいくら支払いが必要かを通知する。
- 集計する対象の費用は光熱費と共有品購入費とする。
- 使いにくいIFだとせっかく作っても使ってもらえなくなる。その為、ユーザフレンドリーなIFとする(LINEとか)。

- ランニングコストは極力抑える。(費用がかかると結局、使わなくなるよね。)

機能概要

機能要件

支出CRUD機能

ユーザが光熱費・共有品購入費で支払った支出の金額をCRUDできる機能。光熱費・共有品購入費は区別してCRUDできるようにする。以下の機能を有する。

支出登録機能

ユーザが支出をシステムに登録する機能。システムに登録後はIDを払い出す。

支出個別参照機能

「支出登録機能」にて登録した金額を参照する機能。「支出登録機能」時に払い出したIDをリクエスト時に指定して対象のリソースの参照を行う。

支出参照機能

「支出登録機能」にて登録した金額の一覧と払い出したIDを参照する機能。

レスポンスする金額の一覧はリクエストを受けた月の一覧とする。

例：10/8にリクエストを受け付けた場合、10/1 00:00~ 10/8 リクエストを受け付けた時点でシステムに登録されている金額の一覧を返す。

また、レスポンスする金額の一覧はユーザ毎にまとめてレスポンスする。

例： ユーザA： 2023/10/8 10:00 : 1000円 2023/10/8 10:10 : 500円 ユーザB： 2023/10/9 11:00 : 1500円
2023/10/10 10:10 : 300円

LINEメッセージAPIで一度にシステムからユーザに対して、送信できるメッセージの文字列に上限等の制約がある場合は、最新数件を返却することとする。以降の工程で確定することとする。

支出更新機能

「支出登録機能」にて登録した金額を更新する機能。「支出登録機能」時に払い出したIDをリクエスト時に指定して対象のリソースの更新を行う。

支出削除機能

「支出登録機能」にて登録した金額を削除する機能。「支出登録機能」時に払い出したIDをリクエスト時に指定して対象のリソースの削除を行う。

支出合計参照機能

「支出登録機能」にて登録した金額の合計を参照する機能。

システムに登録された支出の合計を月毎に参照できるようにする。

レスポンスは最新2月分の支出合計を含めるものとする。

例：

ユーザA： 9月登録合計：1500円

- 光熱費：700円

- 共有品購入費：800円

10月登録合計：3000円 - 光熱費：1400円 - 共有品購入費：1600円 ユーザB：

9月登録合計：2000円

- 光熱費：1000円

- 共有品購入費：1000円

10月登録合計：10000円

- 光熱費：5000円

- 共有品購入費：5000円

支払額集計比率変更機能

月末に支払額集計通知機能で集計を行う際に各ユーザが支払う金額の比率を変更する機能。

支払額集計通知機能

- 月末に、集計月にユーザが登録した支出の合計を算出し、その金額をベースに、支払いが少ない方のユーザに対して、いくら支払う必要があるか通知&支払いを促す機能。
- 支払いを促したユーザから支払った旨の回答が得られない場合は、そのユーザに対して2日に一度、リマインドを行う。
- 支払いが必要なユーザは支払いが完了後、システムに完了したことを登録する。システムへの登録を確認した後、リマインドを停止する。
- 本機能の集計ロジックは以下の通り。
 - ユーザA,Bが居て、支払額集計比率はそれぞれ0.6と0.4と設定していた場合。ユーザAがその月に50000円の支払い。ユーザBがその月に40000円の支払いをしていたとする。
この場合、合計は90000円で比率は0.6と0.4なので、ユーザAは本来54000円の支払いが必要。既に支払っている50000円-54000円を引いて、4000円をユーザBに追加で支払う必要がある。これを数式に当てはめると以下の通り。
ユーザA支払額：x
ユーザA支払比率：x1
ユーザB支払額：y
ユーザB支払比率：y1
算出口ジック：
 $(x+y) \times x1 = \text{ユーザAが本来支払う金額}(R1 \text{と置く})$
 $x - \text{result} = \text{ユーザAの支払い過不足金額}(R2 \text{と置く})$ 。もし、ユーザR2が正の値の場合、ユーザBに支払いの不足が生じているので不足金額をBに請求する通知する。ユーザR2が負の値の場合、ユーザAに支払いの不足が生じているので不足金額をBに請求する通知する。

データ定期削除機能

システムに登録した支出金額・支出金額合計等のデータについて、過去62日を残し、削除する。本機能は日次で実行することとする。

非機能要件

個人で使用するシステムになるので細かい非機能要件は設定しないこととするが、使用しやすさ(ユーザフレンドリーなIF)とランニングコストを抑えることは意識する。

システム方式・構成

アプリケーションPod

本システムの外部に機能を提供するPod。

LINEと連携するため、基本的にはLINEプラットフォームとの連携を想定する。現段階での実装イメージは、LINEからの通知によるイベントドリブンを想定しており、イベントを検知したら、それを契機に裏でAPIを実行する流れを想定する。

データベースPod

ユーザが登録する支出金額や、月毎の集計結果等を保存するデータベース

データ定期削除Pod

ユーザが登録した支出金額や、月毎の集計結果等を62日経過後に削除するPod。削除処理は日次で実施する。現段階ではk8sのCron機能で実現することを想定するが、後の工程で確定する

支払額集計Pod

月末に各ユーザの支出金額をベースに支払いが少ないユーザが多いユーザに対して、いくら支払う必要があるのかを集計する機能。

支払額通知Pod

支払いの少ないユーザに対して支払いの催促をするPod。支払った旨の回答がユーザから得られない場合は2日に一度、支払いの少ないユーザに対して、支払いのリマインドを行う。

利用者一覧

2人で使用することとする。ただし、将来的に3名以上の利用に拡張できる作りとする。

外部インタフェース

ユーザフレンドリーなシステムを目指して、UIはLINEとして、本システムはLINEプラットフォームと連携する。基本的にユーザはLINEを使用して本システムの操作を行うが、一部、ユーザが支払う金額の比率を変更する機能については、LINEではなく、APIでのみ操作できることを考える。後の工程で確定する。

情報・データ

以降の工程で検討する。

以上。