あとでやるｃｈ機能概要

１．目的

ふとしたときにあとでやろうと思いついたことを忘れないようにしたかった
スマホで適当にメモをしてもメモを見ること自体を忘れてしまう（ものすごく物忘れ体質）
かといってカレンダーなどに登録してリマインダー機能を使おうと思うと意外と手間

文章を送信するフォームだけのＷｅｂサイトをつくって
定期的にメールで同じ内容を送信してもらおう！

そう思って調べているうちにリマインくんというＬＩＮＥ公式Ｂｏｔを見つけた
・「宿題」　(✓)「いつですか」「明日の９時」(✓)「登録しました」
ーこのように２回の送信でタスクの登録が出来る
こうして指定した時間にＬＩＮＥが送られてくるというようなものだった

LINEは使い慣れているし　こんな便利なものがあるなら
もうこれで良いのでは？

その時間指定の一手間すら面倒くさい

帰りにお水買ってこうとかふと思いついたときに、わざわざ帰りの時間は18時だからとか考えたくない
天気が良いから後で布団干そうとか、時間なんて明確に決まっていない場合もある

リマインくんの時間指定なしバージョン
あとでやるｃｈを作ろう！　と考えた
（個人的な使用目的なのでいろいろかなり似せている）

時間指定せずにどのタイミングで送れば良いのか？　←１０分毎とかだと迷惑
ー「あとで」が～１時間後くらいを指す
ー「あとで」が１～３時間後くらいを指す
ー「あとで」が３～６時間後くらいを指す
ー「あとで」が６～１２時間後くらいを指す
ー「あとで」が１，２日後～くらいを指す
おおまかにこの５パターンくらいの「あとで」があると仮定した

後でやろうということを忘れずにやりきれればいい
→「さあ、このタスクをやりましょう」という通知としての役割ではなく

あとでやることを覚えておくために目にする機会を増やすという役割にしよう

覚えている時間をなるべく連続させたい
↓
記憶の定着を考えて徐々に通知の間隔を長くしよう(忘却曲線もどき)

最初の通知は「１０分後」、その後は「３０分後」,「1時間後」,「3時間後」
,「6時間後」,「１２時間後」,「２４時間後」、その後もずっと「２４時間後」
でちょうどよさそう

１２時間後は大抵起きてない？わざわざ２４時間後ぴったりに来るのも意味がわからない？
ー １２時間後はなしにしよう
ー２４時間たってもやらないタスクなら重要度は低そうなので毎朝くらいがよさそう
こう考えて最終的には
　１０分後, ３０分後, 1時間後, 3時間後, 6時間後, 次の日の朝6時
のように決定した

つまり目的は、LINEで送信した文章を定期的に送り返してくるBOTの作成ということになる！！

２．環境

公開サーバ : Heroku　　(Webサーバ等詳しいことには言及しない)
フロントエンド : HTML + CSS(Sass) + JavaScript(TypeScript)
JSフレームワーク : Svelte
バックエンド : PHP　　※フレームワークなし
SQL : PostgreSQL + A5:SQL Mk2
テキストエディタ : VsCode
その他 : git　ー　Herokuでデプロイするのに使用
　　remotedebug_ios_webkit_adapter　ー　スマホのsafariでデバッグ用
　　スケジューラ　ー　定期的にLineを送信するためのプログラムを実行させる
　　line公式アカウント
PC : 富士通 AH77/WB
OS : Windows10

ローカルで試さず全てHeroku上で確認していたのでPHPのデバッグがログしか頼れず
少し面倒だった

なぜこの環境なのか

Q　　&	A
Heroku	AWSやレンタルサーバと違い無料だから　→　使用し続けることを意識 お手軽でLineBot作成記事も多く見つかったから
HTML	Liffを使って管理画面をWebページで実装できるから
TypeScript	使ってみたかったから（いまいち利点が分からなかった）
Svelte	前の会社ではVueを使っていたのでVueにしようかと思ったが、オーバースペックな気がしたので軽いと噂のSvelteを使用 ー Vueと考えは似ていたのでかなり書きやすかった
PHP Frameworks	PythonとPHPがLineBotの検索で多くヒットした Web業界を希望しているのでPHPを選んだ　（初めて触った） この規模だとLaravelは必要なく、ピュアPHPの方が高速だと思った
PostgreSQL	Herokuで簡単に使えたから
VsCode	軽くて高機能
ローカル環境？	バックエンドは簡単な処理しかないからデバッグモードなんて必要ないと思っていた

3．システム設計

1．DB設計

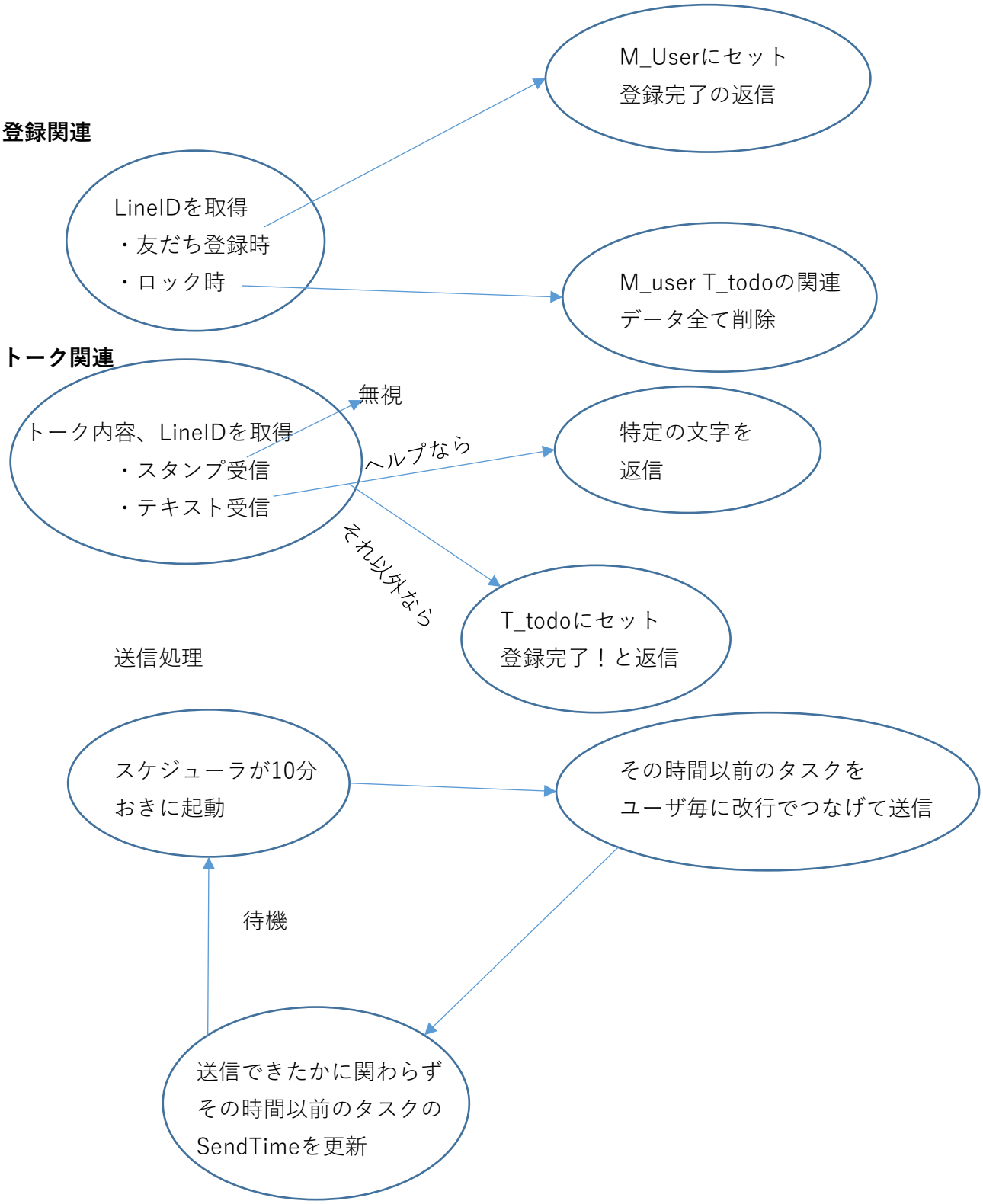
M_user

UserID	varchar(50) Primary	LineのID
ID	serial	内部での識別番号
IsMaster	boolean	デモ登録かどうかの見分け

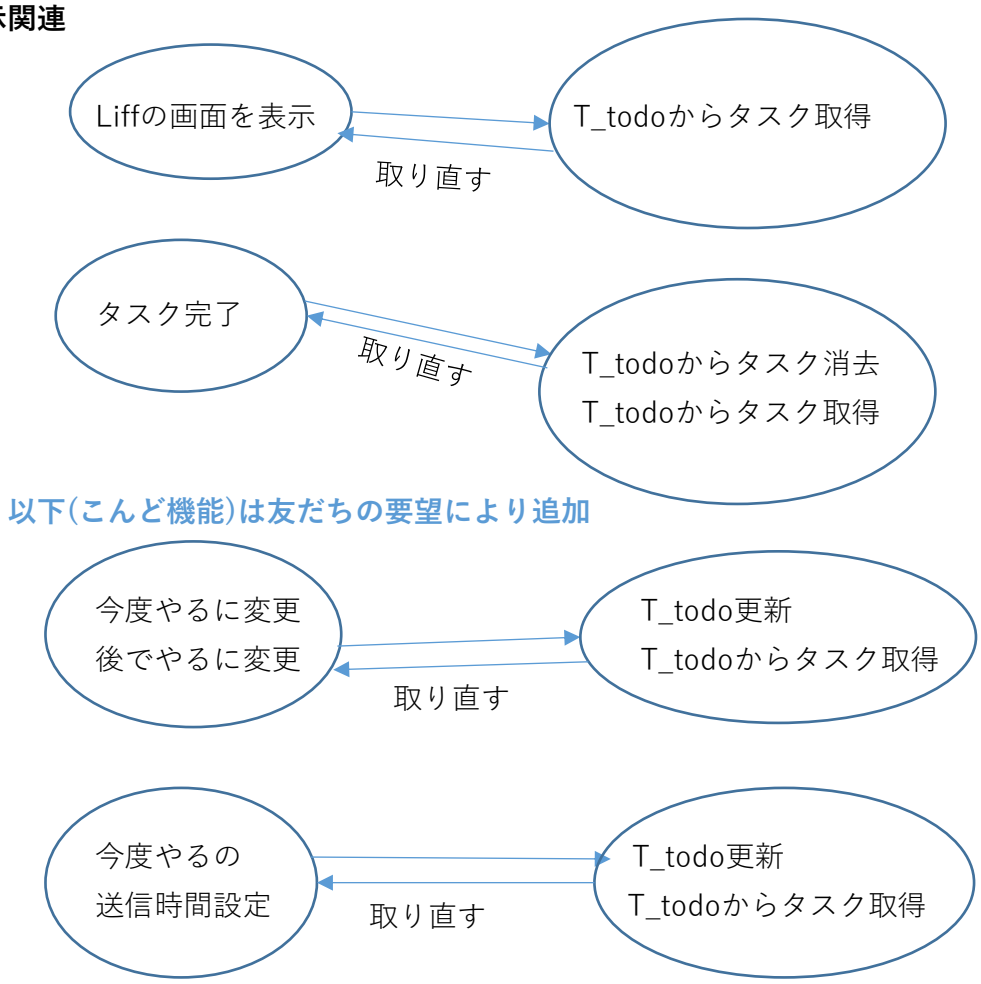
T_todo

ID	integer	┐ Primary	M_UserのIDと同じ
TaskNo	integer	┘	タスクに振る内部番号
Task	text		送信内容
SendTime	timestamp		次に送る時間
Count	integer		何回目の送信か
isNextTime	boolean		今度やるフラグ

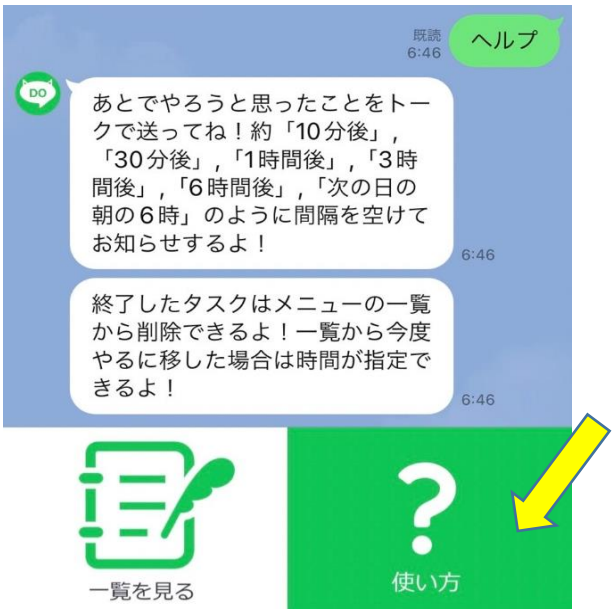
2.データの流れと動き


登録関連
M_Userにセット
登録完了の返信
LineIDを取得
・友だち登録時
・ロック時
M_user T_todoの関連
データ全て削除
トーク関連
トーク内容、LineIDを取得
・スタンプ受信
・テキスト受信
無視
ヘルプなら
特定の文字を
返信
それ以外なら
T_todoにセット
登録完了！と返信
送信処理
スケジューラが10分
おきに起動
その時間以前のタスクを
ユーザ毎に改行でつなげて送信
待機
送信できたかに関わらず
その時間以前のタスクの
SendTimeを更新

表示関連



3.詳しい機能



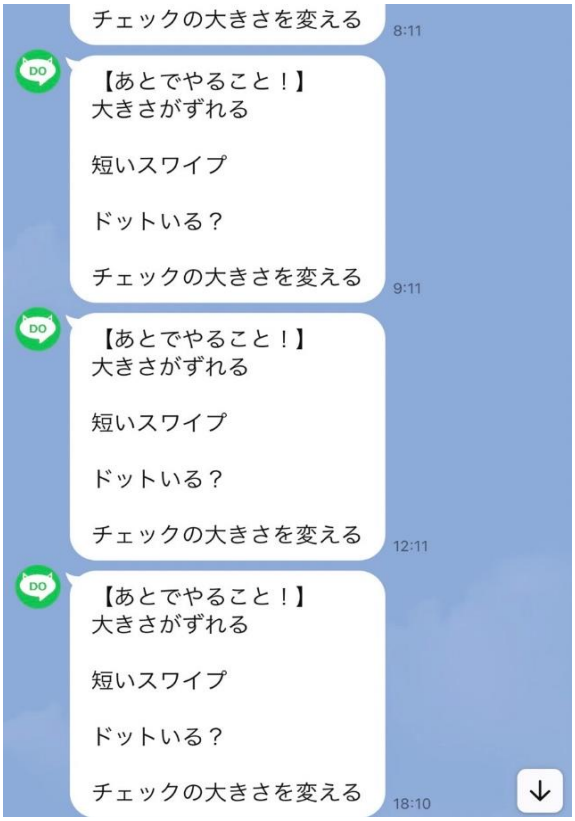
ヘルプ

・「使い方」を押す、または「ヘルプ」と入力すると、画像のように特定の文章が返信される

タスク登録

・「ヘルプ」以外の文字を入力すると、DBに登録され「登録完了！」と返信が来る

・改行とある程度の絵文字も容認している



リマインダー機能

・時間が来るとLINEでタスクの内容が送られる

・複数のタスクがある場合は空白行を入れて一つにまとめて送られる

・基本的に10分単位のはずだが、Herokuスケジューラの都合で数分遅れたりもする

・Line公式アカウントの送信機能は無料で月1000通しか送れないのでデモ登録者に送信はしない（本登録は現状4人だけ）

一覧表示 - あとで　（一覧を見るの初期ページ）

・トークを送り登録されたタスクがまず表示される画面

・←スワイプで削除、→スワイプでこんどに移動することができる

※時間は変更できない

一覧表示 - こんど （下のメニューバーで遷移）

・「あとで」で→スワイプされたものを表示

・時間の初期状態は「今度」で、送信されない

・←スワイプで削除、→スワイプであとでに移動
　することができる
（移動した初期時間は+20分の一桁目切り捨て）

・時間を変更することが出来る
　input type=datetime-localを使っている

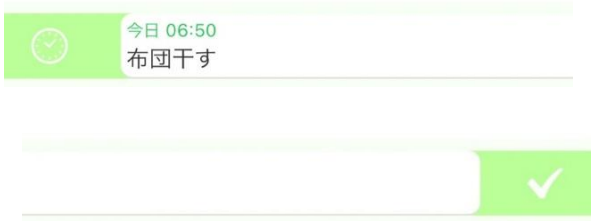
・こんどで送信されたタスクはあとでにいく

「あとで」と「こんど」を分けた理由
・あとではこのBotのモットーであるふと思いついたタスクを登録するためにある

・こんどは時間の決まっている予定や、いつかやりたいことを登録するためにある
　※時間設定までにはいくつかのステップを経なければならないが、予定を登録するときは
　　その手間を掛ける余裕があるはずなので問題ない

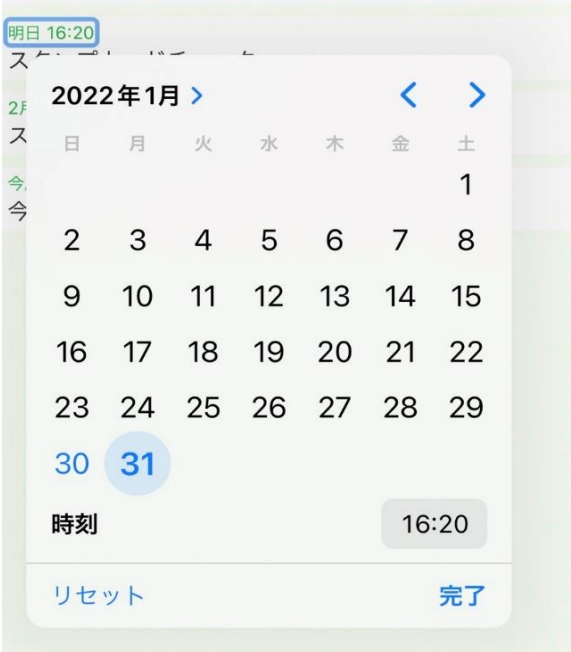
そしてそれら2つ、ちゃんとした予定とふとした思いつきは分けるべき！

4．こだわりポイント



スライド操作
・本家リマインくんでは×ボタンで削除
　スマホっぽくないと感じたのでスライドにした
・Gmailアプリを参考にして作成した
　基準点を超えるとアイコンが拡大される

実装の上での技術面
・検索しても似たようなことを解説しているサイトは無かったので頑張って実装
・Swiper.jsを用いて3枚のスライドを用意する
　1枚目と3枚目はただの透明のスライドで、2枚目にタスクの内容を表示している
　下の要素を緑色にし、アイコンをセットすることでスワイプ操作っぽくしている



時間設定

・本家リマインくんにはない機能
　友だちに欲しいと言われたので追加した
・HTML5標準のインプット要素なので安心かつ、
　意外とおしゃれ

実装の上での技術面

・入力欄はおしゃれだが、表示されるフォームは
　かなりださい
　さらに、datetime-localは ”劣悪：いくつかの要
　素は CSS でスタイル設定できません。”
　とされている要素の内の一つである
　なので値をバインドしてテキストとして表示し、
　フォームを透明にしてそれに覆い被せている

・内部的には'2022-01-01T12:15'のような文字列で持っているが、表記だけねじ曲げている
　値が無い場合は'今度'にして、わかりやすいように今日と明日は文字のまま表示している
　その他の日付は年内なら月から、年をまたぐなら年から表示している
・10分単位で設定するためのstepプロパティが効かないのでchangeイベントで切り捨てている
　（サーバ側でも一応切り捨てている）

件数表示

・本家リマインくんはそもそもタスクが２つに分
　かれていないので存在しない機能
・ネットのを丸ぱくりしただけなので技術的には
　なにもすごくないが、色のバランスとかがすご
　い気に入っている

その他のこだわり

・生のSQLを実行しているので変更しやすくするため全てストアドを呼ぶようにした
・AltoRouterを使ってルーティングをすることでなるべくPHPをまとめた
・マルチタッチ不可など、不具合が起きないように動作の制限をした

5. 微妙なところ

・Swiperで短いフリックの制御が出来なかったのでスッと消そうとしても消えない
　ー 短いフリックを許可させると誤タッチで消えてしまう恐れがある
　ー フリックの強さ等で判断させるのは少し難しそうだったので辞めた
・10分単位でしか設定できない
　ー Herokuスケジューラでぴったり送るのは無理そう
　ー 10分単位の方が見栄えが良いし、実装もスッキリさせられる

・身内で使う分にはいいが、サービスとしては実用に耐えない
　ー DBのバックアップを定期的にとった方が良い
　ー サーバで起こったエラーでログは出すようにしているが通知されたりはしない
　ー クライアント側のエラーはエラー表示してサイトを閉じるだけ
　ー 表示崩れがないかを4人分しかチェックしていない
　ー ユーザを増やすとお金がかかるが、回収手段がない
　ー そもそも大抵の人はリマインくんでいい

6．今後の進展　（いったん満足しているが余裕があればやりたいこと）

・LINEの代わりにメールでも送れるように
・「使い方」を押したときにデモページを表示したい