

2020

ユーザ管理システムのレシピ°



長谷川 喬恒

自由の象徴 Work desk

2020 年 7 月 30 日

内容

1	はじめに	2
1-1	概要	2
1-2	今日の料理は？	2
1-3	料理工程を明確にする	3
1-4	スケジュールの予定を作成する	4
1-5	スケジュールを管理する	5
2	登録のレシピ	7
2-1	材料	7
2-2	お好みで	7
2-3	調理方法	8
2-4	ちょい足しレシピ	10
3	一覧表示のレシピ	11
3-1	材料	11
3-2	調理方法	11
3-3	ちょい足しレシピ	12
4	食卓をかこもう	13
5	ごちそうさまでした	14

1 はじめに

1-1 概要

本資料は、研修課題であるプリントと Web ページの内容を終えた人に対して、最後の課題として課せられた“**秘伝のレシピ**”である。

Java の Form 画面作成を理解している前提で進行していきます。

そのため、各工程は Form 上に配置または実装されることとします

※自力で画面を作成してほしいため画面イメージは極力表示しない

※今までの技術を活用して作り切れることが目的で、どんな環境を構築すれば

目的を果たせるかについても考えてみてください

1-2 今日の料理は？

ユーザ管理システムを作っていきたいと思います。ユーザデータを

登録する登録画面と登録したデータを確認できる一覧表示画面の

2 画面を想定しています。システム—DB の関係で稼働することにな

るかと思います。

1-3 調理工程を明確にする

料理を始める前に今回作るについて、どのタイミングで何がされるのか？そして、次に何が行われるかについての調理工程を書いてみましょう。例えば、野菜をゆでるためには、「鍋に水を入れる→鍋を火にかける→沸騰するまで待つ→鍋に野菜を入れる」のような工程が必要になります。もし、野菜が茹で上がらなかったら鍋に水が入っている？火にかけている？など原因を1つ1つ確かめることもできます。そういった、目に見えにくい一連の流れを確認するためにも調理工程は必要になってきます。（ちなみに「鍋に水を入れる→鍋に野菜を入れる」の流れを見てどう感じるでしょうか？「鍋を火にかける→沸騰するまで待つ」を削り、鍋に水を入れただけで野菜を茹でようとしています。この時点で流れが変ですね。実際に料理を作る前にこの調理工程を書くことでこのような間違いも事前に直すことができます。）

※このことをフローチャートと呼んでいます。各々で調べてフローチャートを

作ってみてください。

1-4 スケジュールの予定を作成する

スケジュール管理をしましょう。たとえば、8月1日～9月1日までの1ヶ月間にユーザ管理システム完成を目標とした場合

8月1日～3日 までは画面の大まかなイメージ図を作る

8月4日～10日 までは新規登録画面を作る

8月11日～20日 までは検索画面を作る

8月21日～9月1日 デバッグ・完成

などと記述して、この通りに調理を進行していきます。さらに、

「完成形→各画面→各機能→クラス・メソッド」と細分化して、

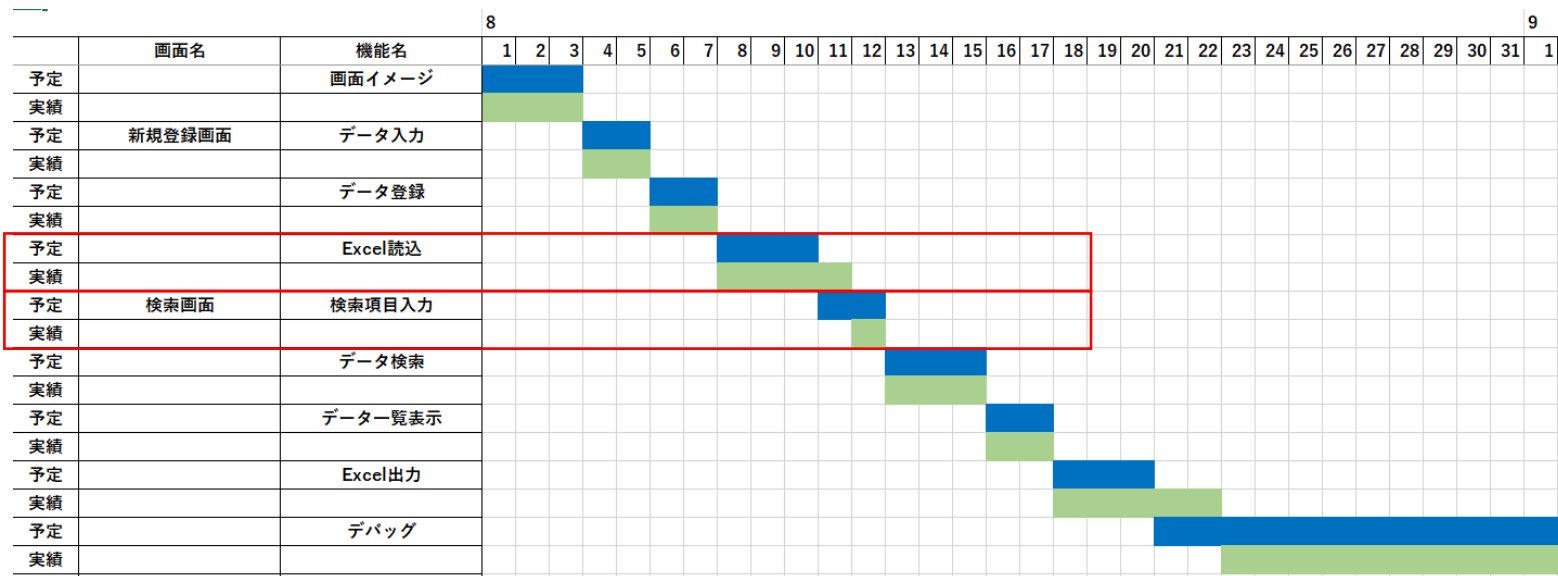
機能ごとにおおよそかかる時間を決めてスケジュールを組むことで
難しい・簡単な部分が判断しやすく、より正確な時間の割り当てができます。

※新規登録画面でも、名前入力といった入力フィールドや入力したデータを登

録する処理など分けることができます。このようにして実装の難易度を知る

ことでスケジュールの遅れ対策を事前に防ぐことができます。

1-5 スケジュールを管理する



スケジュールを組んだら調理を開始して、実績をスケジュールに書き込んでみましょう。図では、Excel 読込で 1 日超過しています。

その後、検索項目入力では 1 日で作業が落ちてしまいました。このように、予定と比較したずれに対して視覚的に遅れているや早く進んでいることを確認できます。そして、遅れてしまっている作業に関しては発見が早ければ早いほど対策もできます。先ほど説明したとおり、図では Excel 読込の遅れを次の作業である検索項目入力で帳消しにしています。こういった形で計画→実行→結果→再調整を繰り返して完成させていきましょう。

※今回、スケジュール管理の説明では[ガントチャート](#)を使っています。スケジュール管理はそれ以外にも[さまざま](#)な管理方法があります。状況に合わせて使い分けてみましょう。

※スケジュール管理にはもう一つ大切な役割があります。それは、終わりが見えることです。終わりが無い作業を長々と続けたくはないですよね？モチベーションを長く保ためにもスケジュール管理は必要なのです。

※今回、ユーザ管理システムでは[Trello](#)を使ってみてください。タスク管理ツールとして使われていて、とてもシンプルに扱うことができます。さらに、[elegantt](#)を追加してください。これは Trello 上でガントチャートを扱える拡張機能です。使い方を詳しく知りたい場合は[こちら](#)などを参考にして下さい。

2 登録のレシピ

2-1 材料

名前ラベル、名前入力、生年月日ラベル、

生年月日入力、性別ラベル、性別入力、

郵便番号ラベル、郵便番号入力、

都道府県ラベル、都道府県入力、住所ラベル、住所入力、

備考ラベル、備考入力、クリアボタン、登録ボタン



2-2 お好みで

お好みで、入力してもらいたい項目を添えてみても良いでしょう。

例えば、ペットはイヌ派？ネコ派？金魚派？など、自分なりの工夫

が加えられるようになれば1人前へ一歩前進できるかも！？

2-3 調理方法

名前ラベル	名前入力（文字列入力）
生年月日ラベル	生年月日入力（年月日入力）
性別ラベル	性別入力（文字列入力）
郵便番号ラベル	郵便番号入力（数値入力）
都道府県ラベル	都道府県入力（文字列入力）
住所ラベル	住所入力（文字列入力）
備考ラベル	備考入力（文字列入力）
クリアボタン 登録ボタン	

※項目名が入力される項目になる 例:名前入力の名前が入力される

※DB の設定については入力項目である程度イメージできるかと思います。

詳細は各自で考えてみてください。

※上記の順番・位置関係で Form が構成されている想定です。

クリアボタンが押されると Form 内の入力項目が空白または初期化されます。

登録ボタンが押されると

名前入力 1 文字以上入力されているか確認する

生年月日入力 日付型に変換できるかを確認する

性別入力 1 文字以上入力されているか確認する

郵便番号入力 1 桁以上数値が入力されているかを確認する

都道府県入力 1 文字以上入力されているか確認する

住所入力 1 文字以上入力されているか確認する

備考入力 確認無し（自由入力）

上記の通りに確認が行われ、入力が正しくなければ登録失敗のダイアログを表示し、確認が取れた場合のみデータ登録処理が行われる

~~※確認処理について、揚げ足取りのような質問は返答しません~~

~~—文章をよく読み解釈してください!!—~~

2-4 ちょい足しレシピ

- ① 生年月日入力を文字列入力からカレンダー入力に変更
- ② 性別入力を文字列入力からラジオボタンに変更
- ③ 郵便番号から都道府県・住所を自動入力
- ④ Excel ファイルからデータ登録（ファイル読込→確認→登録）
- ⑤ データの登録に失敗した際の対策（ロールバック処理など）



3 一覧表示のレシピ

3-1 材料

一覧表示ウィンドウ、検索ボタン



3-2 調理方法

検索ボタン
一覧表示ウィンドウ

検索ボタンが押されると DB 内に登録されているユーザデータを
取得して一覧表示ウィンドウに表示する。

※上記の順番・位置関係で Form が構成されている想定です。

※一覧表示ウィンドウについて、開発環境によって呼び方は様々かと思いま
す。今回はウィンドウと呼んでいますがユーザデータを表示できれば、それ
以外の指定はありません。

3-3 ちょい足しレシピ

検索ボタンがあるのに検索項目がありませんね。検索項目を付け足してみましよう !!

- ① 名前検索（部分一致・完全一致をラジオボタンで切り替える）
- ② 生年月日検索（2つの指定した生年月日の範囲内で取得する）
- ③ 都道府県検索（北海道地方、東北地方、関東地方、中部地方、近畿地方、中国地方、四国地方、九州・沖縄地方からチェックボックスで選択した範囲内で検索する）
- ④ 検索処理を別タスクにて実行（検索処理中でも別の操作ができるようにする）
- ⑤ 一覧表示したデータを Excel ファイルに出力する

※③は素直に都道府県で選択した場合、項目数が 47 必要になる ため今回は
地方区切りにしています。（47 でも止めはしません）

※④は DB のデータ量が多くなってくるにつれ、検索処理をしている時間が長
くなっていきます。その間、画面の操作ができなくなるのをタスクや非同期
処理などで解決していきます。

4 食卓をかこもう

皆さん、お手製の料理は完成しましたでしょうか？もし、完成したのであれば食卓にそれぞれの料理を並べてみてみましょう。使ったDB・エディタ・言語とソースコードをまとめてお互いに見せ合ってください。これまで学んできた内容が同じでも調べて行き着いた答えまで同じではないはずです。お互いの料理を咀嚼することで知識の幅やコーディングの修正点が客観的に見えてくるかと思います。



5 ごちそうさまでした



最後まで読んでいただきありがとうございます。

プログラムに絶対の正解はないと筆者は考えています。それは、時代の移り変わりで改善されるものや変わってしまうものがあり、その時その時で出来ることが変わってしまうからです。（ゲームでいうところの互換性みたいなものですね、PS3 から PS4 になった時に昔のゲームができなくなりました。ただ HD リマスタやリメイクがでて出来るようになったゲームも出てきましたね。これも時代の移り変わりによるものだと思いますね。）

長くなりましたがまとめると踊るバカと踊らないバカ、同じバカなら踊らにゃ損!!

答えが出せている時点でそれは間違いではなく 1 つの考え方です。

結局まとまってない・・・。