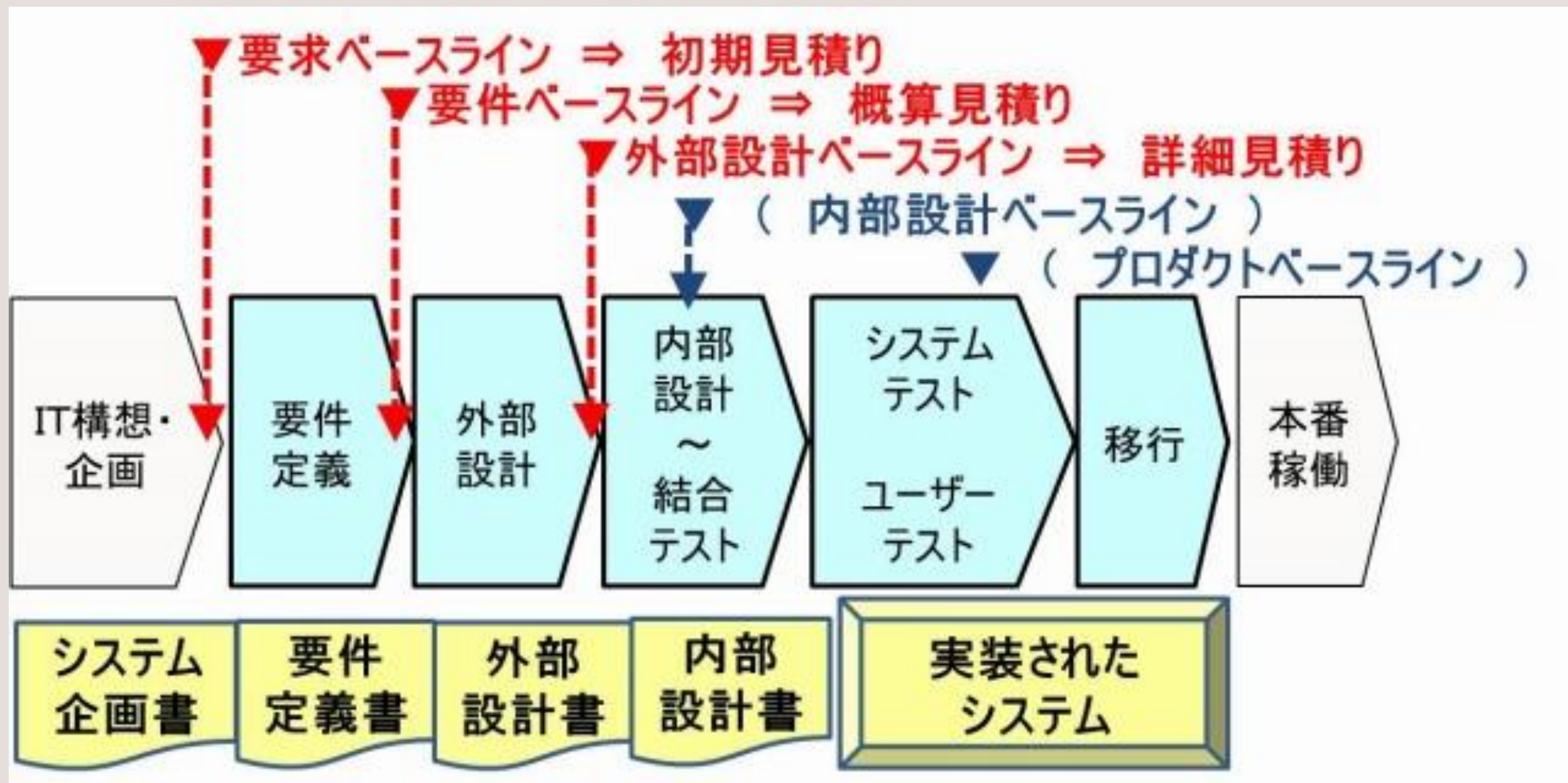





# システム設計実践演習 第7回

# 仕事の流れについて



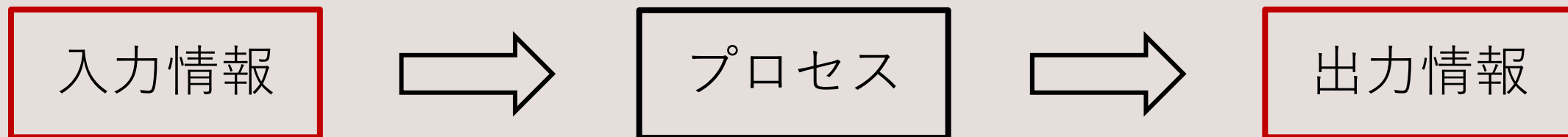


# 基本設計 (外部設計)

# 基本設計(外部設計)とは

基本設計では、システムを「入力→処理→出力」で考えたとき、「入力」と「出力」に関連する部分を設計します。

要するに、**システムの外側から見える動作の設計**を行います。



# 基本設計で考える項目

- システム構成(システム構成図)
- 画面設計
- 帳票設計
- インターフェース設計（他のシステムとの連携）

# システム構成(システム構成図)

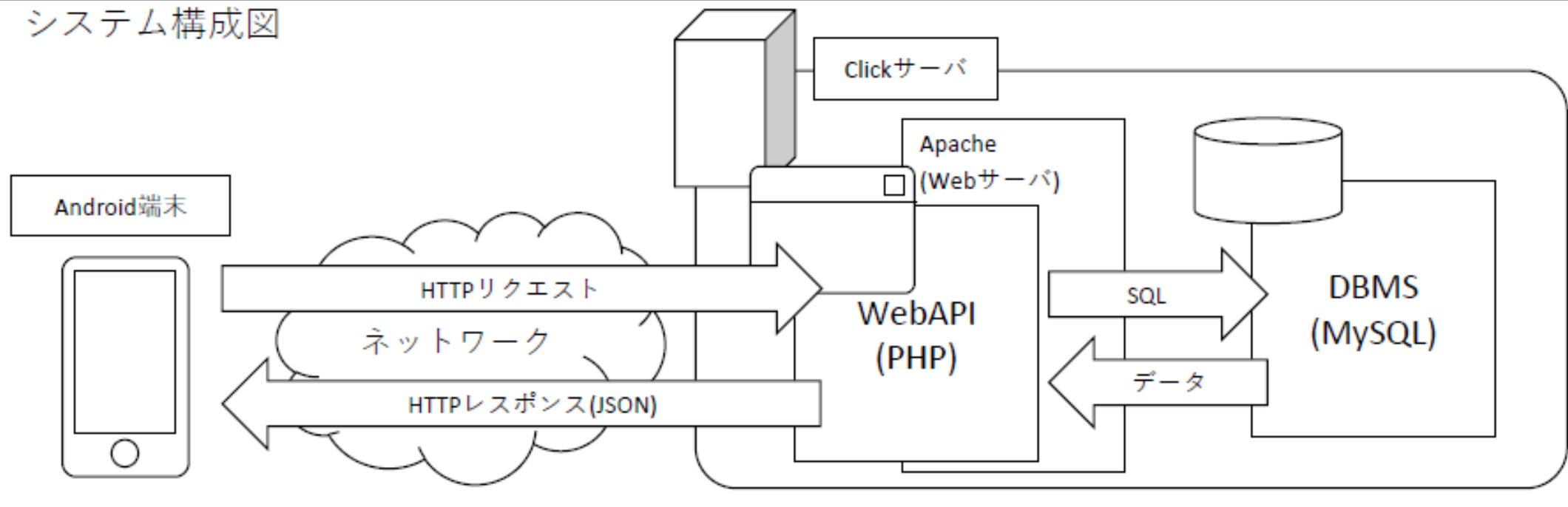
システム構成とは、システムを運用するにあたり必要なサーバーやOS、周辺機器およびハードウェアの組み合わせとなります。

ITシステムはそれぞれ異なる技術を用いた要素を組み合わせているので、**システムの機能やネットワークの構成、サーバーの配置やサービスの連携状況などをグラフィカルに一覧でき、システム全体像を把握する必要があります。**

そのため、基本設計では**システム構成図を作成**します。

# システム構成図例

本番環境 システム構成図



# 画面設計

要件定義では、画面遷移図を作成しましたが、個々の画面の詳細なイメージは描いていません。このため、画面における必要項目の洗い出しや優先順位、使い勝手を考慮した配置などを検討し、実際のシステムの見た目を示す資料を作成します。



# 画面設計で必要な設計書

- 画面一覧（要件定義で精査済み）
- 画面遷移図（要件定義で精査済み）
- 画面共通設計
- 画面レイアウト
- 画面入力設計

※画面一覧と画面遷移図は、画面設計で過不足が出た場合は  
資料を更新すること

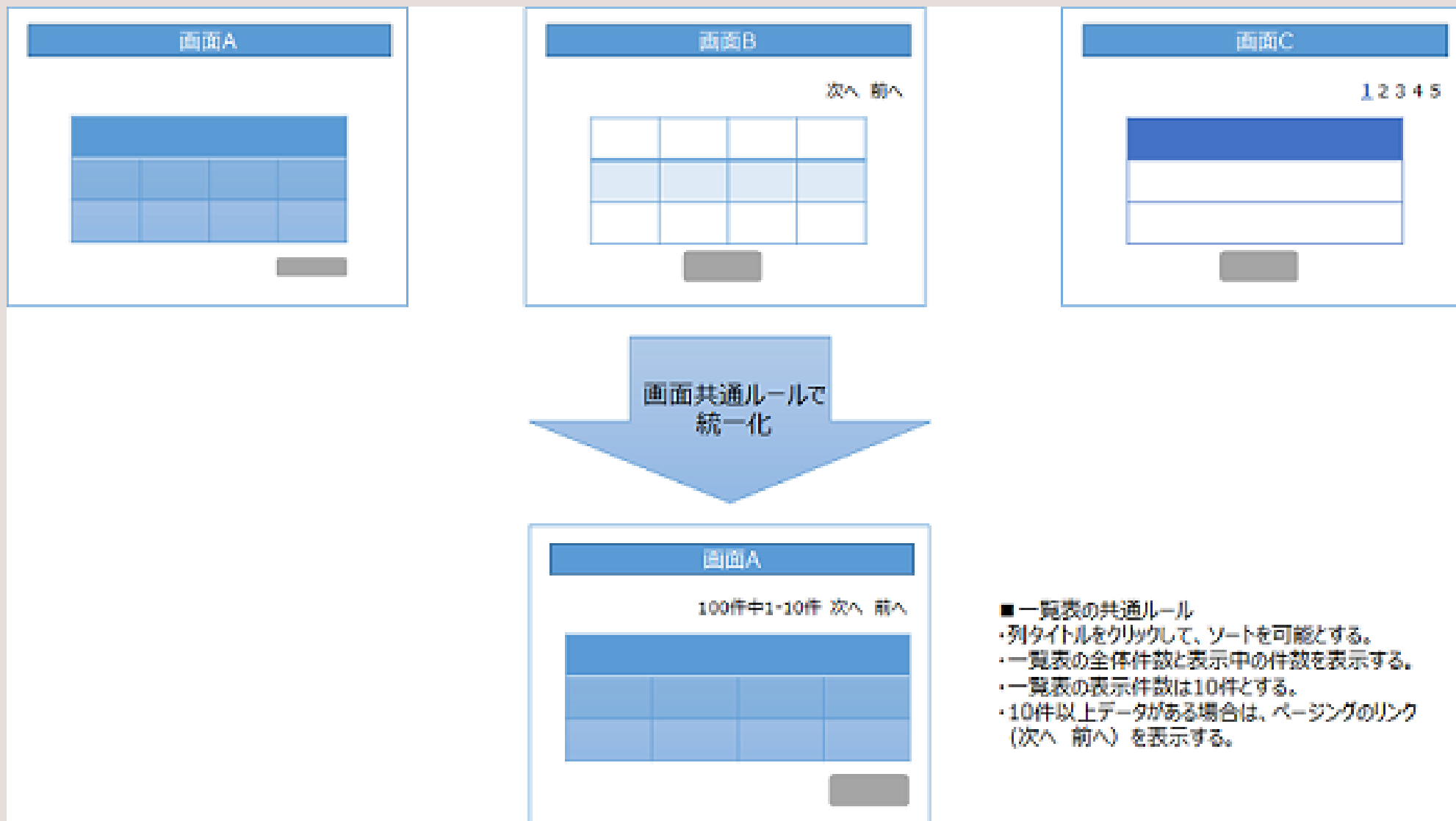
# 画面共通設計

ヘッダーやフッター、メニューなど複数画面で共通の表示を行う場合があります。その共通部分の設計を行うことで、各画面設計での見た目や操作感を合わせることが出来ます。

共通設計では、まずは「共通」となる部分を決めます。システムを通して共通の部分もあれば、サブメニュー領域のように特定のパターンで共通となる部位もあります。

数が多くなる場合は「画面共通部位一覧」のような形で一覧管理しましょう。

# 画面共通設計の概念



# 画面共通設計

全体のヘッダーやフッターといったものは対象としてイメージしやすいですが、個々の設計をしていく中で「これは共通化した方がよいのでは？」と気づくこともあります。

**共通設計→個々の設計という流れにとらわれすぎずに、状況が許せばより適した形で設計作業を進めましょう。**

注意点としては、共通化をしすぎると小回りが利かない（個々の要件に対応しづらい）デメリットがあります。

# 画面レイアウト

画面のレイアウトを設計することで、画面のプログラムや必要となる機能が洗い出せます。また、画面レイアウトとして作成することで顧客が見てわかりやすい形で伝えることができます。さらに「モックアップ」を作成すれば、実際にブラウザで触れる形で確認を取ることができるので、認識ズレを避けやすくなります。

- ・モックアップとは？

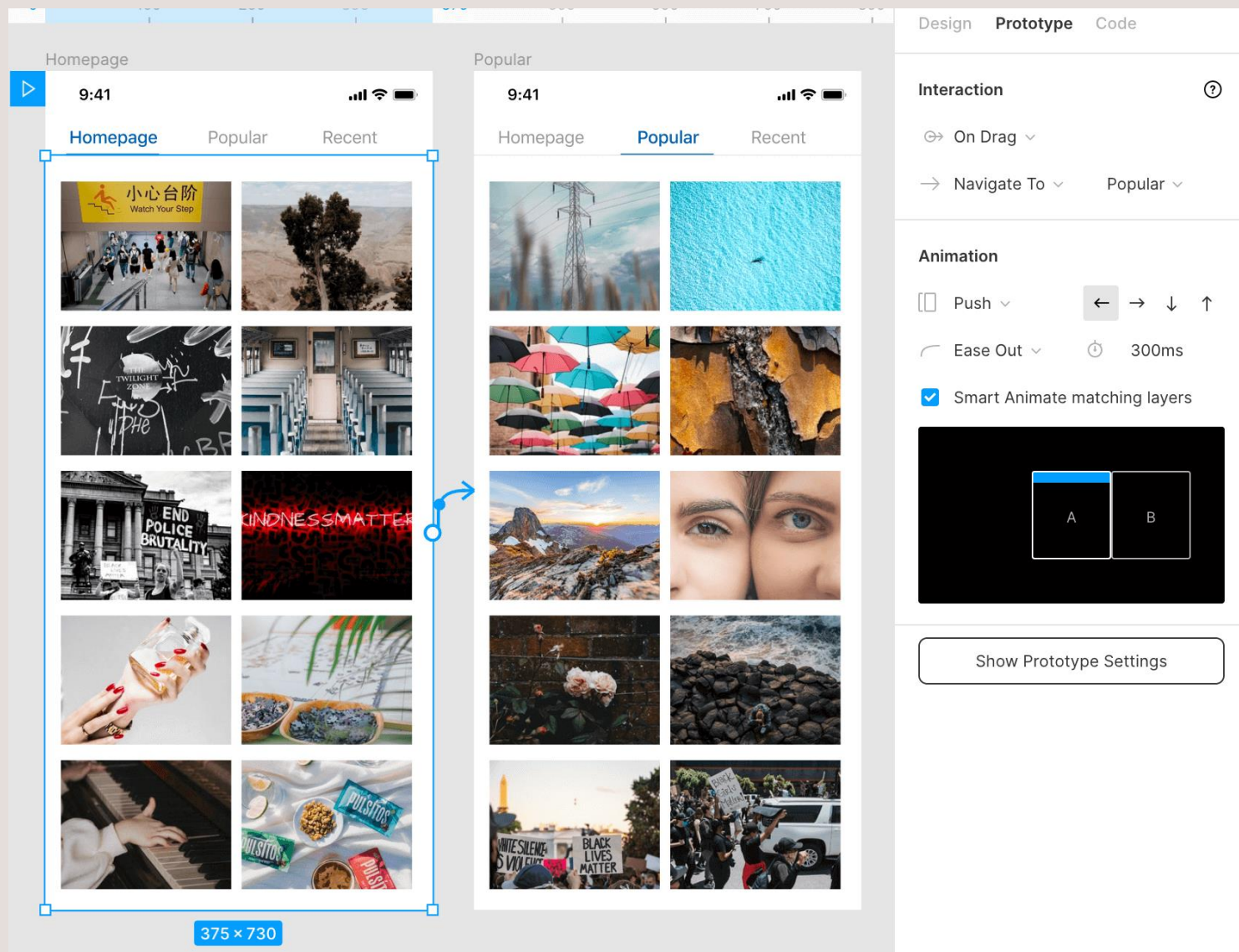
<https://mieru-ca.com/heatmap/blog/what-is-mockup/>

# 画面レイアウト

レイアウトは画面ごとに作成してきます。内容としては**どこにどのような情報、入力エリア、ボタンを配置するのかを決定**してきます。標準化方針があればそれに従いましょう。

また、**それぞれの機能(ボタンを押したときの挙動など)についても説明をしていきます。基本設計では概要を明示するにとどめておき**、具体的な処理は内部設計で内容を設計していきます。

# Figmaを使った画面レイアウト設計



# 画面入力設計

画面の入力項目は、どんな入力値もOKとするわけにはいきません。入力の定義を厳密に行うことで、どのようなチェックをすれば良いのかを明確にするとともに、ダウンしづらい設計をすることが出来ます。



# 画面基本設計例

基本設計書	画面ID	login.jsp	作成者	前田	修正者	
	画面名	ログイン	作成日	2012/4/1	修正日	
1. 画面レイアウト						
<div>ログイン</div> <div>社員コード: <input type="text"/></div> <div>パスワード: <input type="password"/></div> <div>ログイン</div>						
2. 機能説明						
利用者: システム管理者及び社員						
概要: URL入力後の初期画面とする。						
予め登録されている社員番号でしかログインすることはできない。						
3. 項目説明						
NO	項目名	部品	桁数	属性	入力可	備考
1	社員コード	テキストボックス	10	文字	○	
2	パスワード	テキストボックス	10	文字	○	マスク表示
3	ログイン	ボタン	-	-	-	

Figmaなどの別ツール  
で作図した場合は、  
画像を張り付ける

入力項目に条件がある  
なら定義を記述する

# 帳票設計

帳票とは、領収書や納品書など「決まった形式で出力するデータ」のことです。昔は紙で出力していましたが、昨今はPDF化して提供することも多いです。

レイアウトを整えるのは意外と厄介だったりします。

# 帳票設計で必要な設計書

- 帳票一覧(要件定義で精査済み)
- 帳票レイアウト

# 帳票レイアウト

帳票レイアウトを設計することで、プログラムで情報を出力する位置を明確にすることが出来ます。また、1枚の枠に入りきるかどうかを確認することが出来ます。

画面は比較的表示領域が自由に対して、帳票は「1枚にはいりきること」、「改ページの設計が必要」といった点に気を付けなければいけません。また、表示領域が限られているため「桁落ち(情報落ち)」しない設計が求められます。

# 帳票基本設計

10px

16px

10px

納品書

10px

1

2

3

ZZ9/ZZ9 (ZZ9)

4

YYYY/MM/DD 24:59

5

6

出荷元コード

XXXXXXXXXX

出荷部門

NNNNNNNNNNNNNNNN

7

8

伝票番号

XXXXXXXXXX

お届け先コード

XXXXXXXXXX

9

10

11

お届け先名

NNNNNNNNNNNNNNNN

御中

NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

12

13

14

15

16

10px

10px

10px

出荷元コード

XXXXXXXXXX

出荷部門

NNNNNNNNNNNNNNNN

売上区分

X

発注区分

X

支払期限

YYYY年MM月DD日

ヘッダ

伝票ヘッダ部  
(2ページ目以降は、  
売上区分、発注区分、支払期限は印

納品明細

21

10px

No	17	品番	18	商品名	19	20	単価	数量	22	合価	消費税	23	備考	24
1	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
2	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
3	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
4	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
5	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	77	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
6	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	77	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
7	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	77	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
8	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
9	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
10	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
11	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
12	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
13	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
14	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
15	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
16	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
17	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
18	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
19	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							
20	XXXXXXXXXXXXXXXX	NNNNNNNNNNNNNNNN	¥2,ZZZ,ZZ	ZZ	¥2,ZZZ,ZZZ,Z	¥2,ZZZ,ZZ	NNNNNNNNNNNNNNNN							

合計金額

¥2,ZZZ,ZZZ,ZZZ,Z

25

うち消費税

¥Z,ZZZ,ZZZ,Z

26

以上、確かに納品致しました。

10px

●●県△△市□□町4-21

株式会社

◆◆◆◆◆◆◆◆

明細行部  
(25行を超えたら改ページする)

合計行部  
(最終ページにのみ印刷すること)

# インターフェース設計

他サービスや他システム（外部接続先）との接続方式を設計します。**自システムからすると統一した接続方式でシンプルにしたいですが、相手あってのことですのでそう簡単にうまくいきません。**

また、自システムから外部にデータ提供を行うケースがある場合、その提供方法もあわせて設計する必要があります。

# インターフェース設計に必要な設計書

- 外部IF・API一覧（要件定義で精査済み）
- 外部IF・APIレイアウト

# 外部IF・APIレイアウト

具体的にどのようなデータをどのような形式で格納するのか。その細かな仕様を定義するのがIFレイアウトです。接続接続さきにも公開する重要な設計書となります。

自システムの内情を知らない開発者が見る可能性がある設計書です。自システム内でしか理解できない専門用語や誤解を招く表現が無いかよく確認しましょう。認識齟齬がきっかけでシステム障害が発生すると、企業間の責任問題に発展する可能性があります。



# 外部IF・APIレイアウト例

## ■基本情報

IFID	IF001
IF名	見積情報連携
FROM	新CRM
経由	Zapier
TO	在庫検索・見積計算処理(新CRM)
タイミング	随時
プロトコル	HTTPS(POST)
データ形式	JSON
要求URI	https://xxxx.xxx/estimate

## ■リクエストヘッダ

No	項目名	属性	最大バイト数	必須
1	Content-Type	-	-	○
2	apikey	-	-	○

## ■リクエストボディ

No	項目名	属性	最大バイト数	必須
1	pId	英数	10	○
2	quantity	数値	2	○
3	customerName	全半角	100	○
4	zipcode	数値	7	○
5	address	全半角	200	○
6	tel	数値	11	○
7	contactPerson	全半角	50	○

## ■レスポンスヘッダ

なし

## ■レスポンスボディ

No	項目名	属性	最大バイト数	必須
1	statusCode	-	-	○
2	pId	英数	10	
3	status	数値	1	
4	deliveryDate	日付	-	
5	estimation	数値	10	
6	quantity	数値	2	
7	customerName	全半角	100	

# Swagger UIを使ったAPI設計書出力

Schemes

HTTP

Authorize

pet

Everything about your Pets

POST

/pet

Add a new pet to the store

PUT

/pet

Update an existing pet

GET

/pet/findByStatus

Finds Pets by status

GET

/pet/findByTags

Finds Pets by tags

GET

/pet/{petId}

Find pet by ID

POST

/pet/{petId}

Updates a pet in the store with form data

DELETE

/pet/{petId}

Deletes a pet

時間に余裕があったら・・・

◦デザインについての話をする・・・