

20分 設計とは、設計の心得
30分 要件定義、構想設計、基本設計、詳細設計
10分 テーブル設計、テーブル仕様
15分 演習・質問・相談

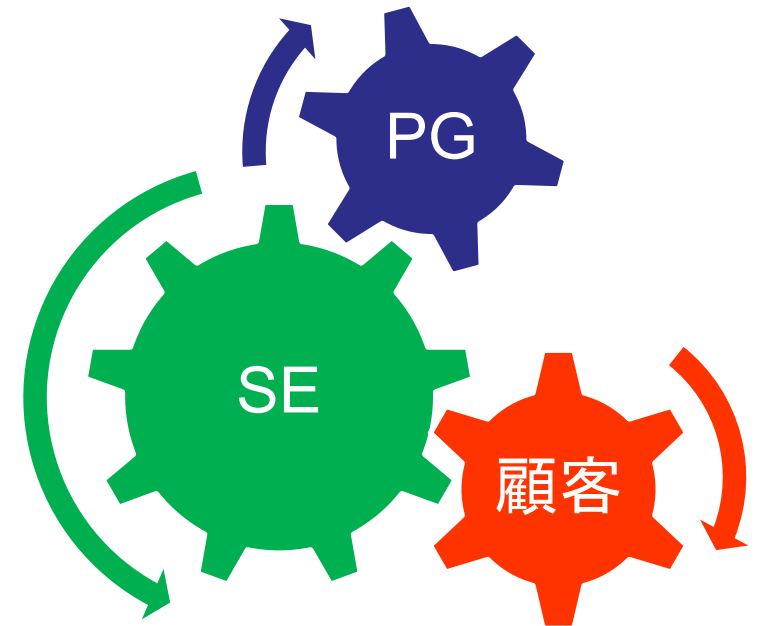
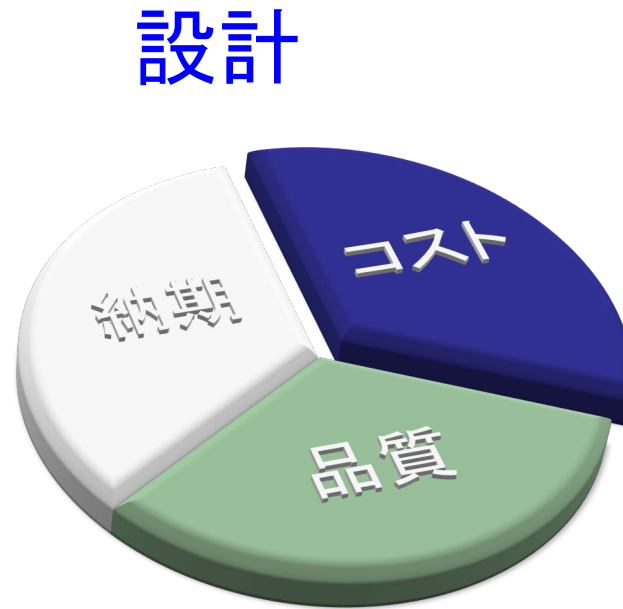
技術教育 (Web)

第1回：設計編

実装から見た設計の進め方

目次

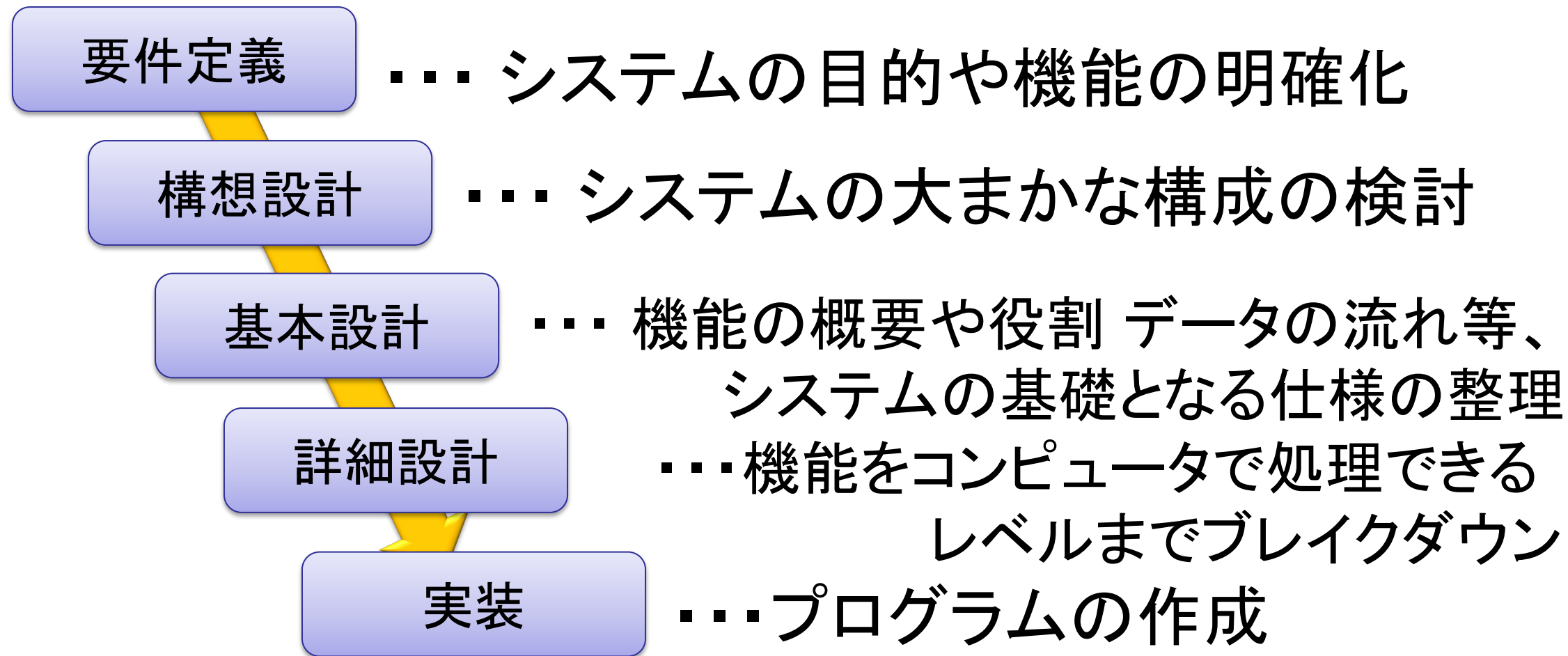
1. 設計とは
2. 要件定義
3. 構想設計
4. 基本設計
5. 詳細設計
6. テーブル設計



設計とは

- 顧客から現状の課題や目的などをヒアリングし、それらを要件としてまとめる
- その要件から基本設計と詳細設計等を行い、プログラマーと協力しながら最終的な設計を完成させる
- 主にシステムエンジニア(SE)が担当
- 業務内容に関する知識、コミュニケーション能力、プログラマーと会話をする技術的知識が不可欠

実装までの開発プロセス



(SCSK九州様「講義資料」より抜粋)

悪い設計（設計書）

- コストや納期を意識せず、顧客の要求を鵜呑みする設計
 - 開発コストが高くて、赤字になる
 - 納期が守れず信用を失う
- 顧客の要求を十分に理解せず、実装できない曖昧な設計
 - 手戻りが多く、赤字のリスクが高まり、納期が守れない
- 記載不備・分かりづらい設計

設計レビュー時によくある指摘事項

- 記入漏れ
 - 改版履歴、指摘事項への対応、画面遷移のラベル名
- 表記ゆれ
 - 画面名、ボタン名等に表示揺れ。例:「取り消し」、「取消」
- 矛盾
 - 設計文書間に機能、画面等が矛盾。例: 構想設計と基本設計の内容が矛盾
 - [画面遷移図]画面名や遷移関係が矛盾、双方向矢印

指摘内容/確認内容(例)

「初版」となっていますが改版履歴によると初版ではありません。

「実施済み」のバイト情報が一覧に表示されるが、実施済みの場合も編集操作が可能なのか？編集不可とすべきでは？

トップ画面(バイト提供者)からバイト削除画面への遷移は画面遷移図にはない。

バイト詳細画面は全ユーザ共通の画面だが、画面遷移図では別々の箱である。

バイト登録確認画面はD13では「バイト希望登録確認画面」で、画面遷移図では「バイト希望確認画面」とされており、画面名に揺れがある。

「登録」を押したときの遷移先は「トップ画面」とされているが、バイト一覧画面の誤り？

指摘内容/確認内容(例)

「作業時間帯」は「作業可能時間帯」に統一。用語の統一について再点検を

バイト詳細画面はユーザ権限によって異なる別々の箱になっているが、画面一覧では同一の項目である。画面一覧のC12に関する記述を参照

バイト確定取り消し画面の「取り消し」は「取消」に修正

矢印のそばに遷移のイベントが書かれてはいるが、矢印が双方向なので、どちらの方向への遷移なのか分かりづらい。

画面一覧と遷移図の画面名が一致していない。

「詳細画面」は「バイト詳細画面」と同一画面であるため、遷移図内に2つ出てくるのはおかしい。画面の箱を1つに統合するか、権限毎に遷移図を分ける必要がある。

要求仕様：議事録管理システム開発

下記、基本機能を満たすWebシステムを開発する。

1. 議事録の登録・編集 (優先度：高)

- － 議事録を登録する
- － 登録した議事録の一覧表示、編集を行う

2. 内容検索 (優先度：中)

- － 参加者や会議の種別等での絞り込み検索
- － 議事録中の任意の文字列を検索

3. 回覧機能 (優先度：低)

- － 参加メンバー全員へ回覧のメールを通知する

要件定義

〔入力〕要求仕様書

顧客の依頼
(要求)

顧客の曖昧な要求を分析し、システムの目的に沿った機能や性能をまとめ、顧客と合意すること

〔出力〕要件定義書

- ①コンセプト
- ②製品名
- ③機能一覧
- ④イメージ図(任意)

(SCSK九州様「講義資料」より抜粋)

要件定義の目的

- 顧客の依頼(要求)を受けて、機能要件・非機能要件を適切に決める
- 予算額、コスト(工数)、開発期間、顧客の満足度などを意識しながら要件を決める
- 要件定義書は開発工数や必要な経費を計算する根拠
 - 要件が多いほど、経費を多く請求できる
 - 要件ごとに必要な工数が少ないほど納期を守りやすい

要件定義のポイント

- 要求と機能が一対一ではない
 - 一つの要求に複数の機能が含まれる場合がある
- 明言されていない機能がある
 - 顧客が明言していないが、実装すべき機能がある
- 開発コストの高い機能がある
 - 実装可能だが、膨大なコストがかかる機能がある
- 非機能要件も要件である
 - 非機能要件も要件であり、経費を請求できる

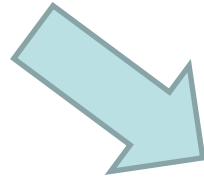
機能要件例

要件	要求	対応
ユーザ認証	なし	追加
議事録作成	〔要求1〕 議事録の登録・編集	分割
議事録編集	〔要求1〕 議事録の登録・編集	分割
議事録検索	〔要求2〕 内容検索	
議事録一覧	〔要求1〕 議事録の登録・編集	分割
議事録回覧	〔要求3〕 回覧機能	
閲覧状況確認	〔要求3〕 回覧機能	追加

構想設計

〔入力〕要件定義書

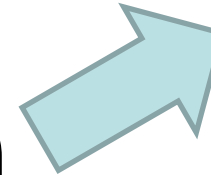
- ①コンセプト
- ②製品名
- ③機能一覧
- ④イメージ図(任意)



要件定義書に基づき、
システムの画面構成を
決め、システムの規模
感を掴む

〔出力〕構想設計書

- ① 画面一覧
- ② 画面遷移図



(SCSK九州様「講義資料」より抜粋)

構想設計〔画面一覧〕

- 複雑な機能は複数の画面で対応
 - － 複雑な機能を分割せず、大きな塊になると、実装が難しくなる
- 画面概要をわかりやすく具体的に書く。例、議事録一覧画面
 - － 悪い例：
「過去に作成した議事録を日付順に一覧表示する。」
 - ・ 表示する対象は？表示する具体的項目？
 - － 良い例：
「過去に作成した参照権限のある議事録を日付順に一覧表示する。
表示内容は会議日時、会議名、回覧状況ボタン等を含む。会議名をクリックすると閲覧画面に遷移する」

機能 vs 画面(例)

• ユーザ認証機能

- ログイン画面
- エラー表示画面

• 議事録登録機能

- 議事録基本事項登録画面
- 議事詳細登録画面
- 参加者登録画面

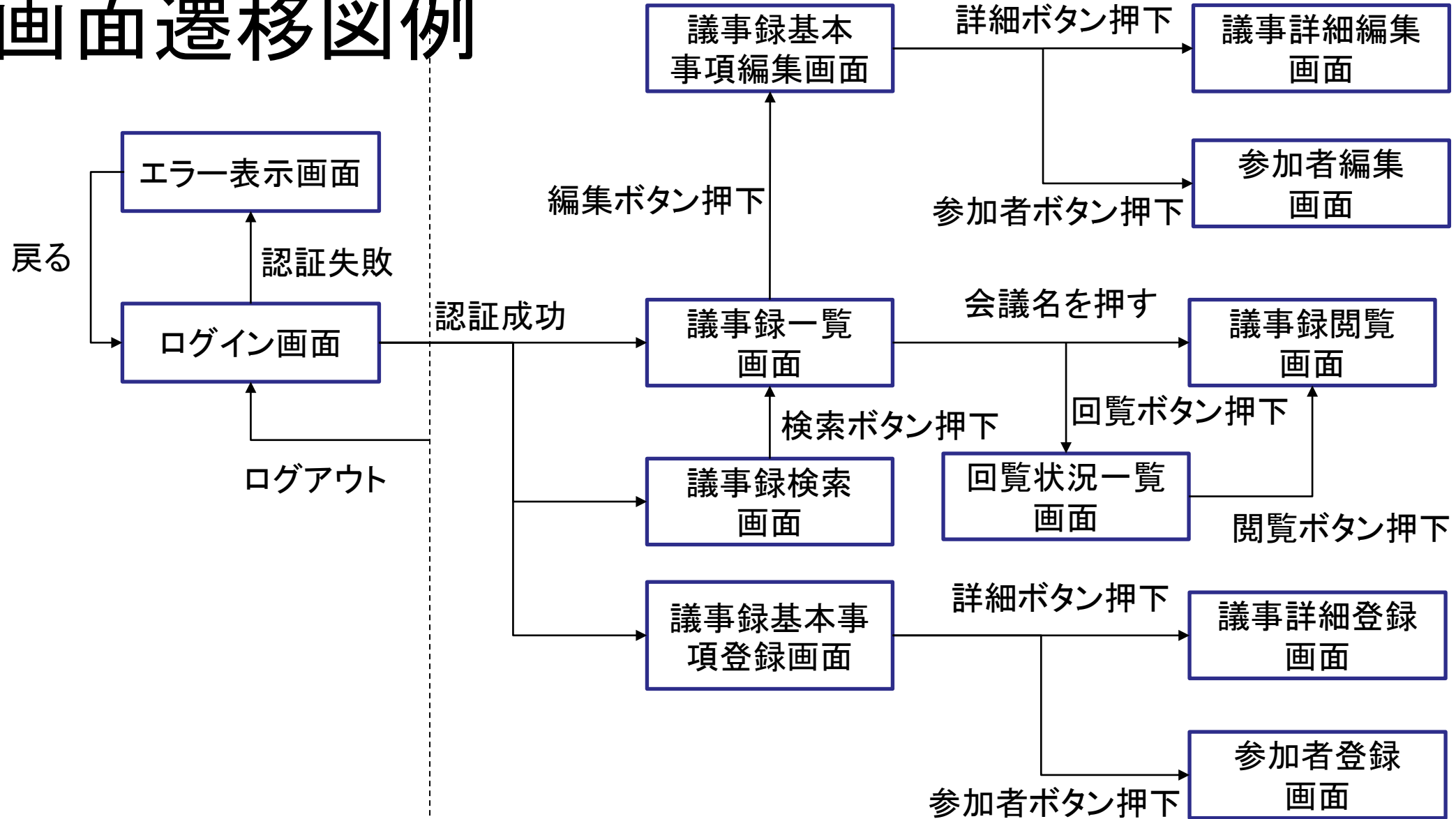
• 議事録回覧機能

- 議事録閲覧画面
- 閲覧状況一覧画面

構想設計〔画面遷移図〕

- 画面遷移図と画面一覧の画面名に矛盾がないこと
 - 誤字脱字に注意
- 矢印は片方向でイベント名も併せて記入していること
 - 双方向矢印や、イベント名未記入はNG
 - イベント名例:「編集ボタン押下」、「認証成功」
- 利用権限別に画面遷移図を分けて描く
 - 画面数が多いとき、遷移図を権限別に分けて書くと良い
- その場で編集可能な描画ツールを使うと便利
 - 例えば、Excelの図形描画ツール

画面遷移図例

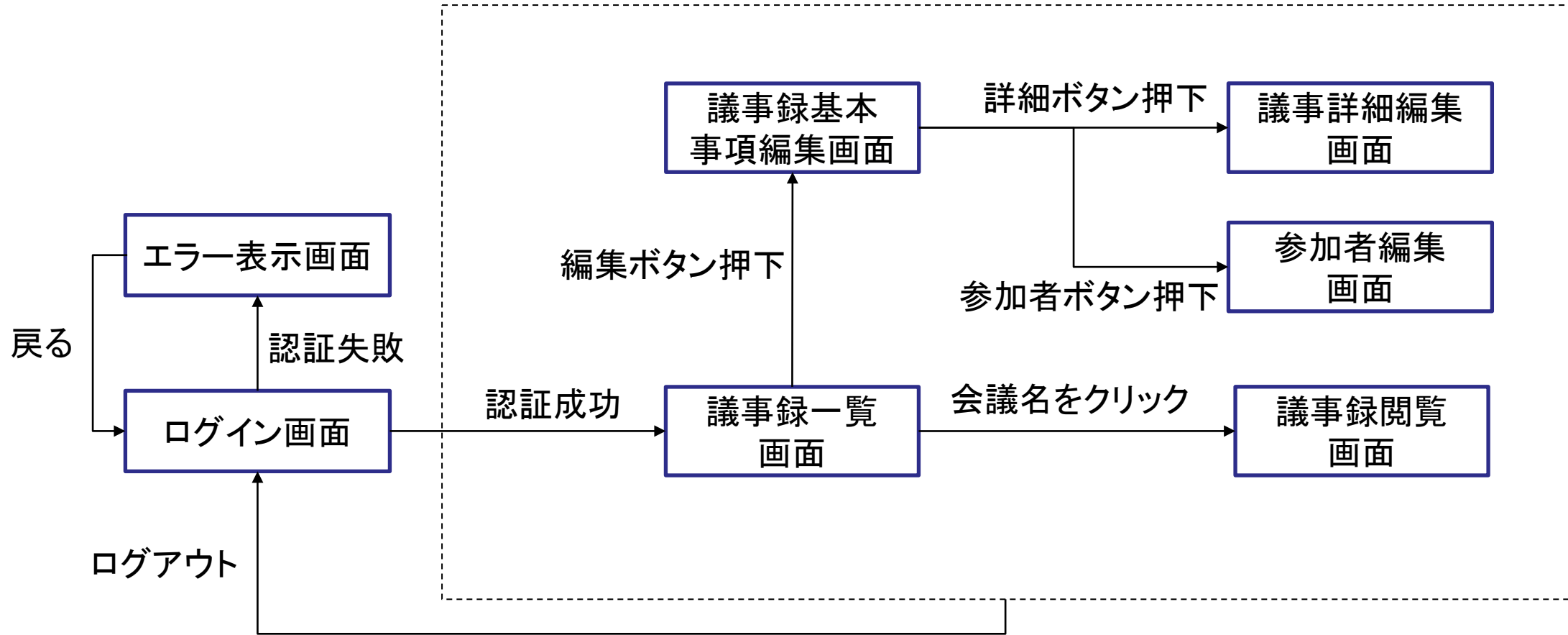


構想設計のまとめ

- 複雑な機能を複数の画面で対応
- 画面概要をわかりやすく具体的に書く
- 画面名、矢印、イベント名などに記入不備のないように
- 要件定義書との整合性を保つように

演習

- Excelで以下の画面遷移図を書け



基本設計

〔入力〕構想設計書

- ① 画面一覧
- ② 画面遷移図

〔出力〕基本設計書

- ① 画面設計書
- ② テーブル設計書

システムの外部仕様
(構成要素と振る舞い)
を定義し、各構成要素
の入出力や、使用する
データを明確化

画面設計書	画面のレイアウト、画面の項目についての説明
テーブル設計書	システムで使用するデータの定義 ※テーブル、項目、データ型

(SCSK九州様「講義資料」より抜粋)

基本設計〔画面設計書〕

- 顧客提供の資料(図表等)を積極的に利用
 - 画面の項目を決めるために、顧客提供の資料が大切
 - 項目名、フォーマットなどの画面の項目を決める
- 画面に適切なデータ例を使う
 - 無意味な文字列ではなく、データのフォーマットや種類を示す適切な例を与えることが重要
 - プルダウンリストの選択肢を項目の説明に明記
- 画面設計書をExcelで直接編集できるよう
 - 別のツールで作成したものは即座に編集できない

画面

会議録登録

良くない例

会議名	
日時	年 月 日() : ~ :
場所	
部署	
議題	

会議録登録

改良版

会議名	要件定義書顧客レビュー ①
日時	2021-10-18 ② 9:30 ③ ~ 11:00 ④
場所	本社 小会議室A ⑤
部署	株式会社KSU Web1班 ⑥ ▼
議題	(1)議事録システム要件定義書説明 (2)納期・予算額の確認 (3)その他 ⑦

登録 ⑧

取消 ⑨

No	項目名	説明	I/O	備考
①	会議名	会議の名称	I	文字列。最大60文字
②	開催日	会議開催の日付	I	日付入力
③	開始時刻	会議開始の時刻	I	時刻入力
④	終了時刻	会議終了の時刻	I	時刻入力
⑤	場所	開催場所（会議室等）	I	文字列。最大60文字。
⑥	部署	開催の部署	I	部署名のプルダウンリスト
⑦	議題	議題の内容	I	テキスト
⑧	登録	会議録の内容をDBに登録	I	ボタン
⑨	取消	編集内容をリセットする	I	ボタン

入力項目の種類

種 類	入力部品	説 明
短い文字列	テキストボックス	画面上で1行以内入力できる文字列
長い文字列	テキストエリア	長さ不明、複数行になりそうな長文
日付、時刻	日付、時刻	HTML5日付・時刻の入力部品を利用
単一選択	(選択肢が少ないとき) ラジオボタン	<ul style="list-style-type: none">• 選択肢以外の入力を避けたいときに使う• 項目の説明に<u>選択肢の定義</u>を明記 (例、1-上司,2-PL)
	(選択肢が多いとき) プルダウンリスト	
複数選択	チェックボックス	
数値	テキストボックス	HTML5では便利な入力部品 (range, number) がある

詳細設計

〔入力〕基本設計書

- ① 画面設計書
- ② テーブル設計書

〔出力〕詳細設計書

- ① 画面設計書(詳細)
- ② テーブル設計書(詳細)

システムの内部仕様
(オブジェクト間の処理の流れやタイミング)を明確化、データの処理方法や、データベース構成を定義

画面設計書(詳細)	画面の入出力項目、操作内容等を説明したもの
テーブル設計書(詳細)	システムで使用するデータの定義 ※長さや必須有無、主キーの設定

(SCSK九州様「講義資料」より抜粋)

詳細設計

- ・ 画面上の操作内容、画面裏での処理の流れやタイミングを、できる限り詳細に記載
- 実装できるテーブルの詳細を記載
- 画面や操作は基本設計までと矛盾のないように
- 記載漏れ、標記揺れ、誤字脱字などに注意

テーブル仕様の考え方

- 画面や機能の実現に必要なデータを網羅する
- 業務内容から「**実体**」と「**関連**」を洗い出す
- 業務内容からデータ項目を決める
 1. 入力画面で**入力されるデータ項目**
 2. 出力画面に**表示されるデータ項目**
 3. 機能の**実現に必要な他のデータ項目**
例：ユーザ権限をチェックするに**ユーザ種別**が必要

テーブル仕様の設計

- 列名（物理名）は大文字・小文字を区別しない
 - `USER_ID` と `user_id` は同じ列になる
 - PHP変数名は原則小文字
`$USER_ID`(×) → `$user_id`(○)
- 整数の主キー項目は自動採番(SERIAL型)を使う
 - 重複を許さない主キー項目は整数の場合、
`AUTO_INCREMENT`による自動採番を使うと便利
 - 外部キーで参照する際には、`INT`になる
 - 整数ではない項目(例:ユーザID)は、自動採番が使えない

自動採番(例)

```
create table t_review (  
  review_id SERIAL PRIMARY KEY comment '口コミID'  
  , eval_point INT NOT NULL comment '評価点, 1..5'  
  , review_comment TEXT NOT NULL comment 'コメント'  
  , rst_id BIGINT NOT NULL comment '店舗ID'  
  , user_id VARCHAR(32) NOT NULL comment '登録ユーザー'  
) comment '口コミ';
```

口コミID(review_id)に
値を与えずに口コミ登録

```
INSERT INTO t_review (eval_point, review_comment, rst_id, user_id) VALUES  
  (4, 'メニューも豊富でした', 1, 'u007')  
  , (5, 'お肉のボリュームが有って安い、肉好きの聖地。', 2, 'u003')  
  , (4, 'お酒好きの多い、ちょっとグルメな人たち', 1, 't001')  
  , (4, '透き通った活ヤリイカは禁断の美味しさ', 3, 'u002')  
  , (3, '駅から遠いのが難点だが、それでもまた利用したいお店', 1, 'u010');
```

自動採番(例)

自動採番による口コミID (review_id) 自動生成

review_id 口コミID	eval_point 評価点	review_comment コメント	rst_id 店舗ID	user_id 登録ユーザID
1	4	メニューも豊富でしたがいかんせんお腹がいっぱいでそんなにお料理を頼む事ができませんでしたが創作料理的...	1	u007
2	5	この店は、お肉のボリュームが有って安い、肉好きの聖地。	2	u003
3	4	お酒好きの多い、ちょっとグルメな人たち」向けのお店だと思いました。	1	t001
4	4	透き通った活ヤリイ力は禁断の美味しさ！活造りだけでは終わらないよ～	3	u002
5	3	駅から遠いのが難点だが、それでもまた利用したいお店	1	u010

まとめ

- 設計は顧客の課題や要求をもとに、形から中身までシステムを決めていく作業
- 設計はコスト、納期、品質を常に念頭に置く必要
 - 実装の技術的可能性、コスト(工数)等を十分に考慮
- 設計書を提出する前に、自己チェックが重要
 - 記入漏れ、表記ゆれ、矛盾がないか
 - そうしないと、NGの回数が増え、進捗に悪影響を与える

演習・質問・相談

次回の予告

- 実装の進め方
- PHP・SQLの基礎と実戦的応用
- サンプルの使い方