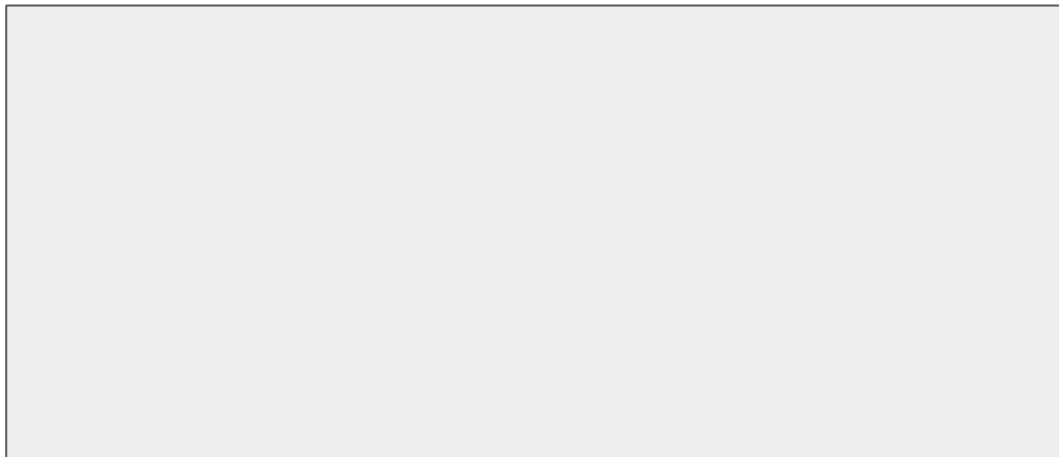


チームD

無人コンビニシステム

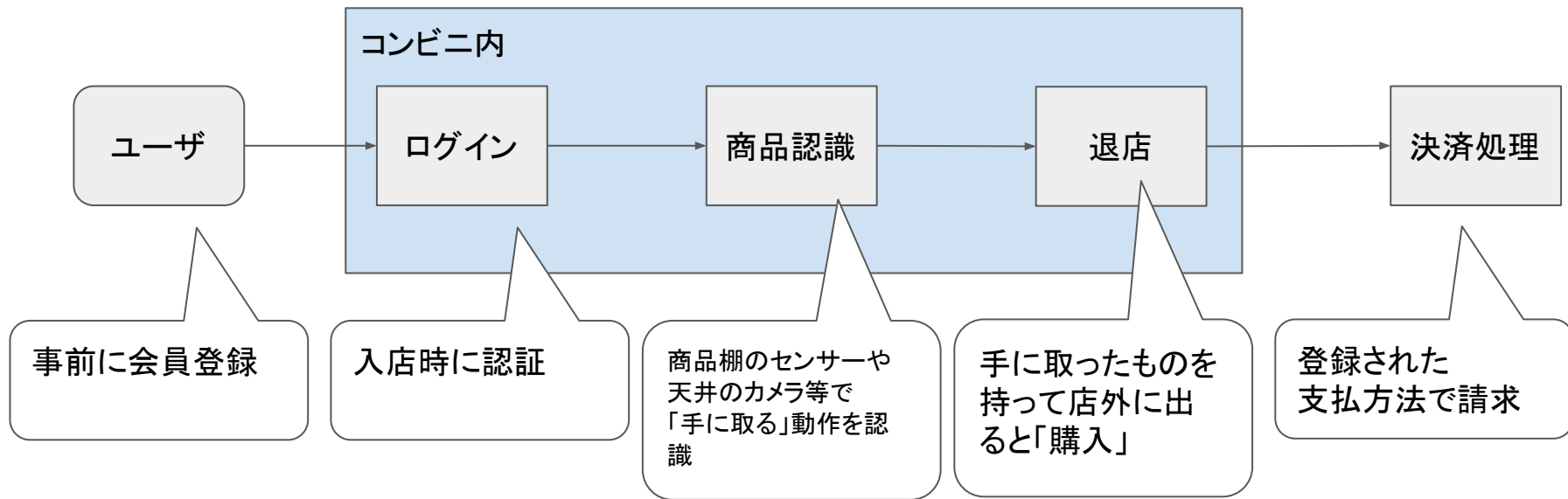


システム化構想

新規事業：無人コンビニシステム

事業目的	新たな購入体験の提供、利便性向上、人件費の削減、万引き防止
事業領域	買い物をよりシンプルにするサービス
事業特徴	ユーザ、店舗の両者が「買い物」を効率化
将来の展望	コンビニ加盟店の増加、コンビニ以外の店舗自動化
付加価値	コンビニ会社に関わらず加盟店であればユーザがどこでも利用可能
収益モデル	新規性・話題性による販売増、業務効率化によるコスト削減
類似システムの把握	Amazon Go

業務フロー



システムフロー

会員登録

会員情報DBに
レコード作成

ログイン(来店)

カメラの顔認証やスマ
ホアプリ認証

商品認識

・天井のカメラ・商品棚
のセンサでユーザと商
品を識別
・会員情報DB・在庫管
理DBを参照して、一時
的な明細(カート)を作
成

退店

明細を確定し本部に
送信

決済

・明細を集計してDB
更新
・決済会社に連携

在庫管理

・明細をもとに
在庫管理部(システム内or外)に情報
連携



お見積り

費用項目		人	日	単価	合計
会員登録	画面	1	15	15,000	225,000
	DB設計	1	5	15,000	75,000
マイページ		1	10		0
ログイン認証		1	5	15,000	75,000
パスワード再設定		1	3	15,000	45,000
在庫管理	DB設計	1	10	15,000	150,000
	画面	1	10	15,000	150,000
一時明細作成		1	20	15,000	300,000
明細確定・本部送信		1	3	15,000	45,000
明細受信	モジュール	1	20	15,000	300,000
	DB設計	1	10	15,000	150,000
在庫管理	画面	1	5	15,000	75,000
	モジュール	1	15	15,000	225,000
	DB設計	1	10	15,000	150,000
クレカ会社連携		1	10	15,000	150,000
			151		2,115,000

フロントエンド設計

ユーザアプリ
webページ

○会員画面

- ・基本情報(氏名、住所)
- ・支払い情報(クレカ、銀行口座等)
- ・認証情報(入店時のユーザ識別のための情報)

○ログイン認証

- ・ログイン画面
- ・ログイン結果通知

○購入明細

- ・購入商品一覧→商品詳細
- ・決済履歴

○返品

- ・返品履歴
- ・返品申し込みフォーム
- ・払い戻し明細

○店舗検索

- ・商品検索
- ・在庫状況確認

管理者アプリ
webページ

○顧客統計

- ・店舗検索
- ・売上→商品別
- ・来客者数
- ・在庫→商品別
- ・認証成否状況

○返品

- ・顧客ごとの返品状況・払い戻し状況

○顧客ごとの未払い情報

バックエンド設計

ユーザ関連

○会員情報VIEW

①会員情報ヘッダ

- ・基本情報(氏名、住所) DB

②会員情報明細

- ・支払い情報(クレカ、銀行口座等) DB
- ・認証情報(入店時のユーザ識別のための情報) DB

○購入明細DB

- ・会員情報、商品代金、日時

○返品DB

- ・ユーザID、商品ID、日時、返品状況

○店舗マスタDB

- ・店舗ID、店舗名、住所

管理者関連

○在庫管理DB

- ・商品ID、価格、数、店舗ID

○統計

- ・フロントエンドに対応したDB作成

要件変更

経営の判断でプロモーションを高めるために、会員のランクを作り、製品価格を一括、値引きすることにした。この際に、システムはどのように変更するか。

変更後一フロンテンド設計

ユーザアプリ webページ

○会員画面

- ・基本情報(氏名、住所)
- ・支払い情報(クレカ、銀行口座等)
- ・認証情報(入店時のユーザ識別のための情報)
- ・**ランク情報確認(現ランク、割引、累計利用額)**

○ログイン認証

- ・ログイン画面
- ・ログイン結果通知

○購入明細

- ・購入商品一覧→商品詳細
- ・決済履歴

○返品

- ・返品履歴
- ・返品申し込みフォーム
- ・払い戻し明細

○店舗検索

- ・商品検索
- ・在庫状況確認

管理者アプリ webページ

○顧客統計→**ランク別統計を追加**

- ・店舗検索
- ・売上→商品別
- ・来客者数
- ・在庫→商品別
- ・認証成否状況
- ・**値引き額合計**

○返品

- ・顧客ごとの返品状況・払い戻し状況

○顧客ごとの未払い情報

○**ユーザーランク定義機能**

- ・**ランク定義**
- ・**ランク別値引き率**
- ・**ランク昇進条件(何円利用で昇進とか)**
- ・**ランク機能 on off**

変更後—バックエンド設計

ユーザアプリ
webページ

○会員情報VIEW

①会員情報ヘッダ

- ・基本情報(氏名、住所)DB

- ・**ランク**

②会員情報明細

- ・支払い情報(クレカ、銀行口座等)DB

- ・認証情報(入店時のユーザ識別のための情報)DB

- ・**累計購入額**

○購入明細DB

- ・会員情報、商品代金、日時

○返品DB

- ・ユーザID、商品ID、日時、返品状況

○店舗マスタDB

- ・店舗ID、店舗名、住所

管理者アプリ
webページ

○在庫管理DB

- ・商品ID、価格、数、店舗ID

○統計

- ・フロントエンドに対応したDB作成

→**ランク別DB追加**

○ランク詳細DB

ランクと値引き率、開始日、終了日

課題

1. 見積もりの甘さ:工数設定が感覚的。また、作業内容を全て列挙しきれていない
2. DBやシステムフロー(モジュール)に関して、店舗毎に実装するのか、本部のみの実装なのかを詰め切れていない
3. バックエンド設計がDB設計のみの記載になっているが、フロントエンドの処理を実行するためにどのようなモジュールが必要なのかを設計できなかった。理想はフロントエンドの処理に対応したモジュールと DBの関係図のようなもの

ご清聴ありがとうございました