**基本設計書**

**ブロックチェーンを用いた**

**酒類在庫共有システム**

**1198 桑田惇平**

**1.3版**

|  |  |  |
| --- | --- | --- |
| 版数 | 年月日 | 内容 |
| 1.0 | 2020/11/30 | 初版 |
| 1.1 | 2020/12/01 | ・2.2節「機能詳細」を追加  ・3章「本演習で用意するデータ」を追加 |
| 1.2 | 2020/12/01 | ・1.3節「システムの最終的なスコープ」を追加  ・2.1節「システム概要図」を追加  ・2.3節「機能詳細」を修正  ・3章「本演習で用意するデータ」を修正 |
| 1.3 | 2020/12/03 | ・2.1節「システム概要図」を更新  ・2.2節「機能一覧」を更新  ・2.3.6節「他店舗在庫の問合せ」を追加  ・3.2節「中央データベース上のデータ」を追加  ・3.3節「データ整合性管理」を追加 |

目次

[1. 要件定義 4](#_Toc57900809)

[1.1 要求概要 4](#_Toc57900810)

[1.2 システムの目的 4](#_Toc57900811)

[1.3 システムの最終的なスコープ 4](#_Toc57900812)

[1.4 要求機能一覧 4](#_Toc57900813)

[1.5 開発環境 5](#_Toc57900814)

[1.6 開発体制 5](#_Toc57900815)

[1.6.1 開発メンバー 5](#_Toc57900816)

[1.6.2 スケジュール 5](#_Toc57900817)

[2. システム機能定義 6](#_Toc57900818)

[2.1 システム概要図 6](#_Toc57900819)

[2.2 機能一覧 6](#_Toc57900820)

[2.3 機能詳細 6](#_Toc57900821)

[2.3.1 在庫の表示 7](#_Toc57900822)

[2.3.2 検索結果表示 7](#_Toc57900823)

[2.3.3 取り置き依頼 7](#_Toc57900824)

[2.3.4 在庫の入力・登録 7](#_Toc57900825)

[2.3.5 在庫の自動登録 8](#_Toc57900826)

[2.3.6 他店舗在庫の問合せ 8](#_Toc57900827)

[2.4 REST API入出力概要 8](#_Toc57900828)

[2.5 画面レイアウト 8](#_Toc57900829)

[2.6 エラーメッセージ一覧 8](#_Toc57900830)

[3. 本演習で用意するデータ 9](#_Toc57900831)

[3.1 ブロックチェーン上のデータ 9](#_Toc57900832)

[3.2 中央データベース上のデータ 9](#_Toc57900833)

[3.3 データ整合性管理 10](#_Toc57900834)

# 要件定義

## 要求概要

現在、通販サイトにおける日用品などの購入が増える一方で、嗜好品を実店舗で買う消費者も依然として存在する。2019年の「店舗を利用する商品ジャンルに関する調査」(予約ラボ)によると、回答者の59％がお酒などの嗜好品を「実店舗で買う」と回答した。

しかし、高級酒類や季節限定の商品はその販売店舗が限られるため、場合によっては消費者が小売店を「はしご」して商品を探す手間が生じる。そこで、今回ブロックチェーンを用いて各小売店が酒類在庫を共有し、「店舗を超えた店舗」を構築するためのシステムを開発することとなった。

## システムの目的

近隣の小売店の在庫の一括検索・取り置き依頼を行うためのUIを提供することで、1.1でも記載した消費者の「はしご時間」をなくす。

また、各小売店が近隣他店舗の在庫を勘案し、他で扱っていない酒類を仕入・販売することも可能にする。(※本演習ではこちらの機能は実装しない)

## システムの最終的なスコープ

・システム適用地域：東京駅より半径５km圏内

・システム利用人数：400～500名

## 要求機能一覧

|  |  |  |
| --- | --- | --- |
| 項番 | 要求機能 | 説明 |
| 1 | 取扱酒類の一覧取得 | 在庫がある酒類の一覧を表示する |
| 2 | 酒類在庫の検索 | 欲しい酒類の在庫があるか検索する |
| 3 | 酒類の取り置き依頼 | 在庫のある酒類の取り置きを依頼する |

※本演習で実装する機能のみを記載

## 開発環境

|  |  |
| --- | --- |
| OS | Windows10 |
| 開発言語 | HTML, JavaScript, Python, Solidity |
| 動作環境 | Google Chrome |

## 開発体制

### 開発メンバー

・桑田 惇平

### スケジュール

・2020年11月30日～2020年12月25日（20日間）

・11/30(月)～12/02(水)　基本設計

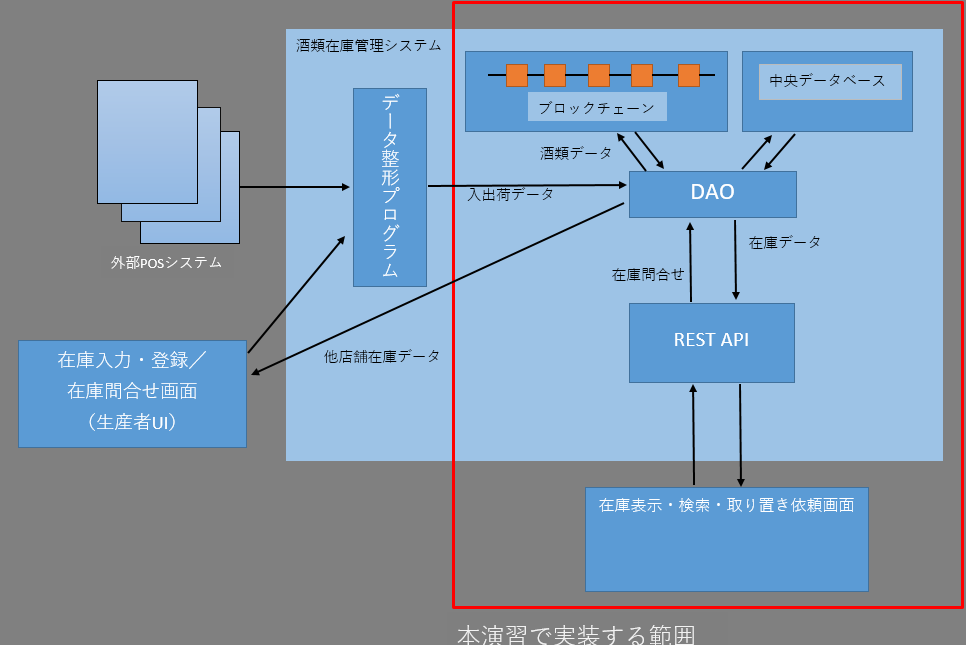
・12/03(木)～12/08(火)　詳細設計

・12/09(水)～12/17(木)　コーディング

・12/18(金)～12/25(金)　テスト

# システム機能定義

## システム概要図



## 機能一覧

|  |  |  |
| --- | --- | --- |
| 項番 | 要求機能 | 説明 |
| 1 | 取扱酒類の一覧取得 | 在庫がある酒類の一覧を表示する |
| 2 | 酒類在庫の検索 | 欲しい酒類の在庫があるか検索する |
| 3 | 酒類の取り置き依頼 | 在庫のある酒類の取り置きを依頼する |
| 4 | 在庫の入力・登録 | 在庫の入力と登録を行う |
| 5 | 在庫の自動登録 | 外部POSデータと連携し、在庫の自動登録を行う |
| 6 | 他店舗在庫の問合せ | 近隣他店舗の酒類在庫を問い合わせる |

※本演習では、項番1～3の機能を実装する

## 機能詳細

以下に示す３機能(2.3.1～2.3.3)は、１つの画面上で動的に処理する。

### 在庫の表示

・URLを入力し、在庫表示画面にアクセスした際に表示される

・REST APIから酒類一覧を取得する

・取得結果は「New Arrive 新着入荷」２点と「Ranking 人気のお酒」3点とに

　分かれて表示される

・画面上部の検索窓(テキストボックス)から酒類の在庫検索ができる

### 検索結果表示

・検索窓に文字列を入力し、「SEARCH」ボタンを押下すると、

「RESULT 検索結果」が表示される

・データベースは各商品ごとのトランザクションIDを保持し、

　REST APIからの問い合わせを受けてブロックチェーン上の

　商品情報を閲覧する

・検索結果は最大5件が横並びで表示される

・検索結果が5件以上に及ぶ場合、６件目以降は表示しない

（※開発スケジュール（実績）に余裕がある場合、5件ごとの

ページネーション機能を実装する）

### 取り置き依頼

・「New Arrive 新着入荷」、「Ranking 人気のお酒」及び「RESULT 検索結果」に

　表示された商品をクリックすると、取り置き依頼ポップアップが起動する

・ポップアップには商品の写真、商品名、取扱店舗名、前日の在庫数が表示される

・ポップアップ下部には「取り置きを依頼しますか？」という確認文が表示され、

　最下部に「取り置きを依頼する」「キャンセル」という２つのボタンが表示される

・「取り置きを依頼する」ボタンを押下し、正常に取り置きが完了した場合「取り置きが完了しました」というメッセージが表示される

・「取り置きを依頼する」ボタンを押下し、正常に取り置きが完了しなかった場合、

　エラーメッセージ101(2.6節参照)が表示される

以下の２機能(2.3.4, 2.3.5)はそれぞれ独立して開発する（本演習では実装しない）。

### 在庫の入力・登録

・ユーザID, パスワードを用いて入力・登録／在庫問合せ画面にログインする

・当該画面では「新規在庫の登録」「既存在庫の変更」「他店舗在庫の問合せ」を行える

### 在庫の自動登録

・外部POSシステムとの連携を行う

・POSデータをもとに、1日ごとに在庫数を計算しデータベースに登録する

### 他店舗在庫の問合せ

・近隣他店舗の在庫問合せを行う

## REST API入出力概要

|  |  |  |
| --- | --- | --- |
| 項番 | リクエスト名 | 説明 |
| 1 | GET | 在庫のある酒類の一覧を取得する |
| 2 | POST | 酒類の取り置き依頼を処理する |
| 3 | PUT | 酒類の在庫の有無、取扱店舗の取得を行う |
| 4 | DELETE | 今回は使用しない |

## 画面レイアウト

同フォルダ内「画面レイアウト.xlsx」を参照

## エラーメッセージ一覧

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 項番 | メッセージID | メッセージ内容 | 分類 | 説明 |
| 1 | 101 | 取り置き依頼に失敗しました。  在庫がないか、既に  取り置きされています | E | 取り置き依頼処理が適切に終了しなかった場合に表示 |

※分類

E: エラーメッセージ

W: 警告メッセージ

I: インフォメーション

# 本演習で用意するデータ

## 3.1 ブロックチェーン上のデータ

本演習ではダミーの酒類在庫データを40～50件、JSON形式でブロックチェーンに

格納する。

属性は以下の通り。

1. 商品名（文字列）
2. インフォメーション（配列）
   1. 取扱人（文字列）
   2. 取り置き依頼が可能か（ブール値）
   3. 入荷日（文字列）

2.4 取り置き実績（数値）

※本演習では簡便化のため、取り置き実績を「人気係数」とみなし

　「人気のお酒」３点の表示に利用する

## 3.2 中央データベース上のデータ

中央データベースは、DAOによる可及的速やかなブロック閲覧を可能にする、およびデータ整合性の確認を行うための最低限の属性を保持する。

具体的な属性は以下の通り（主キーは「◎」で示す）。

1. ◎商品名（文字列）
2. ◎取扱人名（文字列）
3. 残在庫数（数値）
4. ブロックチェーン上のトランザクションID

## 3.3 データ整合性管理

本システムでは、中央データベース上のデータとブロックチェーン上のデータとの間で齟齬が生じるおそれがある。そこで、データ整合性を担保するための手立てとして、以下に示すような処理をDAO上で行う。

1. 中央データベース上の「残在庫数」が0で、ブロックチェーン上の「取り置き依頼が可能か」がTRUEだった場合

・在庫がないにも関わらず、取り置き依頼が可能であることになっているため、DAOは

　当該ブール値をFALSEに書き換えてREST APIに送る。その後、更新後のJSONを

　ブロックチェーンに保存し、新たなトランザクションIDで中央データベースを更新

　する

1. 中央データベース上の「残在庫数」が1以上で、ブロックチェーン上の「取り置き依頼が可能か」がFALSEだった場合

・在庫がないにも関わらず、取り置き依頼が可能であることになっているため、DAOは

　当該ブール値をTRUEに書き換えてREST APIに送る。その後、更新後のJSONを

　ブロックチェーンに保存し、新たなトランザクションIDで中央データベースを更新

　する

上記2ケースの他に、「中央データベース上に存在する商品名と取扱人名との組が、ブロックチェーン上に存在しない場合(またはその逆)」が想定できる。だがこのケースに関しては本演習のスコープ外にあたるため、整合性担保のための処理実装は行わない。