

# 汎用的な端末を利用した小規模店舗での支払い機能の開発 Development of Payment System for Small Stores Using General-purpose Terminals

三井海翔<sup>†</sup> 楠海斗<sup>†</sup> 鳴見智大<sup>†</sup> 伊藤匡祐<sup>†</sup> 中西一貴<sup>†</sup> 兼宗進<sup>†</sup>  
Kaito Mitui, Kaito Kusunoki, Tomohiro Narumi, Kyousuke Ito, Kazuki Nakanishi, Susumu Kanemune

<sup>†</sup>大阪電気通信大学

## 1 はじめに

近年、大型スーパーやアパレルショップなどでは商品の会計、支払いの自動化が進みスムーズな接客を行うことが可能となっている。しかし、デジタル化が進んでいない小規模な商店では大規模な機械を導入することが資金面でも敷地面でも困難という課題がある。そこで本研究ではPCやタブレットのカメラを利用してバーコードを認識することで、小規模で効率的かつ簡潔に会計処理を行うことを目指す。このシステムにより、従来の店員による手作業に依存する会計方法を改善し、小規模店舗の運営効率を向上させることを期待する。

## 2 システムの概要

本システムでは、タブレットを用いてバーコードの読み取りを行う。読み取ったバーコードのJANコードはサーバーに送られ、該当する商品の金額、商品の名前を返す。上記をバーコードスキャンするたびに行い、タブレット上で合計金額を計算し出力する。開発はオンラインプログラミング環境「BitArrow」[1]を利用した。バーコードの認識はQuaggaJS[2]を利用した。

本システムの使用方法としては、起動するとカメラ画面がブラウザ上に表示される。商品の購入者はカメラにバーコードをかざして読み取り、JANコードがカメラ画面の下に表示される。その後、JANコードと該当する商品名、値段が表示される。次々とバーコードをスキャンしていくと、その都度、商品名と値段の表示、それに加え、読み取った商品の値段が加算され、合計金額も表示されるようになっている。

本システムにより出された合計金額を用いることで、レジで行う作業は金銭の受け取りのみとなる。これによりスムーズな対応とレジ作業の効率化が可能となっている。図1に本システムの実行画面を示す。



図1 システムの実行画面

JANコードとそれに対応する標品情報はCSVファイルとしてまとめられている。このCSVファイルを任意のサーバーコンピュータに格納しておくことでオンライン

上から新商品の追加や値段の修正を容易に行うことが可能である。システムを導入する店舗の商品を入力し保存する。

表1 商品をまとめたCSVファイル

JAN_code	Product_name	Price
4902775068819	ベビースター	40
4977821321491	ヤングドーナツ	40
4903013830274	駄菓子屋さんパック	330
4903013032562	日本一ながーいチョコ	40
4903013549503	どらチョコ	40

## 3 BitArrowを利用したシステム開発

BitArrowはブラウザからプログラミングできる教育用の学習環境である。現在はJavaScript、ドリトル、簡易C、DNCL(どんくり)、Python、Tonyuなどの複数の言語に対応している。プログラム実行方式には、ブラウザ実行とサーバ実行の二種類がある。教育用JavaScriptはブラウザ実行のみに対応しており、Pythonはブラウザ実行とサーバ実行のどちらも利用可能である。図2にBit Arrowのプログラム記述画面を示す。

BitArrowでは、クライアント側 (HTML、JavaScript) とサーバー側 (Python) の間でデータのやり取りが可能である。サーバー側の Pythonプログラムをクライアントから呼び出す際は、あらかじめサーバー側でhttp通信からプログラムを実行するためのURLを生成しておく。その後、クライアント側からは、GETパラメータを使ってデータをURLに含めてリクエストを送信する。サーバー側のPythonは、GETパラメータのデータをinput関数で受け取り、プログラムを実行する。結果はprint関数によって標準出力に出力され、GETリクエストのレスポンスとして返される。図3に、Bit Arrow の情報システムの概要図を示す

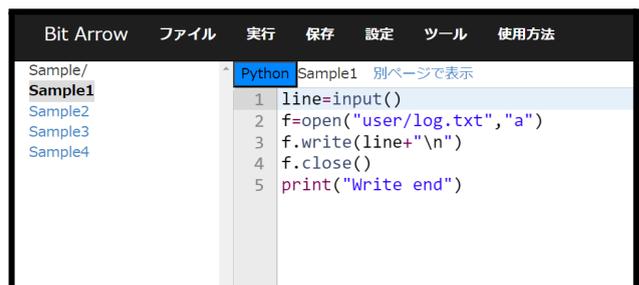


図2 Bit Arrow のプログラム記述画面

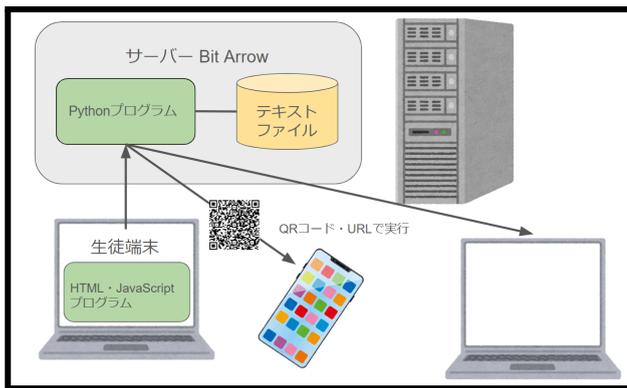


図3 Bit Arrow のサーバープログラム実行方式

BitArrowによるクライアント側プログラムとサーバー側プログラムの記述について説明する。クライアントからの通信として送られたメッセージの文字列を、Pythonのinput() 関数で受け取り、print() 関数にて文字列を送り返すことができるという点で、これにより、サーバー側で複雑なプログラムを記述することなく、Pythonの記述方法と大きく異なることもなく、違和感の出づらい作りになっている。BitArrowでは、素材管理と呼ばれる領域に、画像データやテキストデータ等の各種のデータを置くことができる。素材管理には、ユーザ本人だけが読み書きできるアクセスできるユーザエリア(user)と、受講するクラスに 所属する全員がアクセスできるクラスエリア(class)がある。図4に、素材管理のファイル一覧ダイアログを示す。

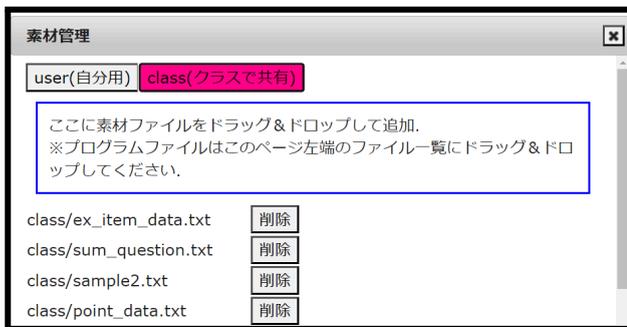


図4 Bit Arrow の素材管理ダイアログ

クライアント側のプログラムでは、HTML/CSS と教育用 JavaScript を使用してサーバーと通信する。図4には、メッセージの入力と送信を行う input タグとボタンがあり、サーバーからの実行結果および読み込んだデータを表示する result エリアとhistエリアがある。JavaScriptプログラムでは、read関数でプログラムを呼び出し、サーバー 側で実行結果を histエリアに表示する。write 関数では、inputタグで入力されたデータをに送信し、サーバー側でテキストファイルに書き込む。resultエリアには、サーバーへのリクエスト前に「RunningCGI...」と、書き込み成功時に「Write end」が表示される。

#### 4 まとめと今後の展望

実用的に使えるシステムを作れたため、学校の文化祭や大学祭などで利用して利便性を評価していきたい。また、機会があれば小中学校の授業で使ってもらえるように準備を勧めたい。

また、これをスーパーやショッピングモールにある買い物かごに取り付け、レジがなくても買い物かごだけで完結するレジ搭載買い物かごとして発展させたいと思った。

クレジットカードや電子マネーなどのキャッシュレス決済を用いれば、レジもいらず、このシステムのみで会計を済ませることができる。

現状では商品情報の編集がCSVを直接編集する必要があるが、店側が容易に編集することができる画面を用意していきたい。

#### 参考文献

- [1] オンラインプログラミング環境 ビットアロー (Bit Arrow), <https://bitarrow.eplang.jp/>, 2024-01-05
- [2] QuaggaJS, <https://serratus.github.io/quaggaJS/>, 2024-01-05