Vol.2013-CSEC-61 No.23
Vol.2013-IOT-21 No.23
2013/5/10

ポイントカードシステムのデータベース開発と評価

岡崎　功[†1]

携帯電話端末向けの低コストなポイントカードシステムの開発に関して，データベースの設計・開発を行った．ポイントカードシステムは加盟店の各店舗から与えられるポイントを，顧客が各店舗の電子的なカードに記録し利用するものである．システムはクラウド上のサーバと携帯電話端末から成る．データベースは MySQL 5.5 を使用し，データ整合性やアクセス制限に配慮した設計を行った．さらに構築したデータベースについてパフォーマンスの評価を行った．

Database development and evaluation of a point card system

ISAO　OKAZAKI[†1]

For the low-cost point-card system using high-function cell phones, a database of the system was designed and developed. The system uses electronic cards in stead of the traditional paper cards. Points recorded on the card are stored into the database. The point-card system consists of cell phones and a server computer in the cloud internetwork. The database was implemented using MySQL 5.5 with due consideration for the data consistency and security. We also checked the performance of the developed database.

1. はじめに

飲食店，美容院など，多くのお店でスタンプカードもしくはポイントカードと呼ばれるカードが販売促進の手段として利用されている．これは，来店するお客にカードを配布し，商品購入時に金額に応じてスタンプを押印，スタンプが幾つか集まったら何らかのサービス商品を与えるというものである．お店側としてはリピータとしてのお客を確保しすることで，売り上げ向上を目的としている．このようなカードは，紙カードであったり，磁気的な記録ができるカードであったりする．消費者であるお客は，お店ごとに配布されているカード類をいちいち持ち歩かねばならない. たくさんのカードが財布やバックの中でかさばったり，使用するときに目的のカードが見つからない，自宅に忘れてきた，時には紛失するなど，多数のカードを利用するうえでは不便極まりない．ストレスを感じる消費者が多い．そのような中で，お店側としても，お客によるカードの紛失・忘却などのために生じるリピータ減少を抑えるべく，もっと有効にカードを利用したい，又はカードを媒介として広告を出すなどの販売促進を望むところである．

近年, 多機能携帯電話端末の普及がめざましい. 我々は，合資会社インターゾーン[1] の企画・立案のもとで，携帯電話端末に着目し，ポイントカードを電子化（クラウドデータ化）することで，携帯電話端末で一括管理を可能にする以下のようなスマートポイントシステム（スマッピーと呼ぶ）を開発している．

- 消費者の店舗別ポイントカード情報を電子化，クラウド上に保存・管理し，端末アプリケーションからクラウドサービスへ接続することで，ポイントカードのデータ，及び店舗が発信するセール情報等を利用する．
- 消費者はポイントカードの煩わしさやカードの紛失を解消でき，お店からの情報を確認できるメリットがある．
- 店舗側のメリットは，サービス利用者の増加によるリピーターの増加，及び当サービス利用者に対するセール情報等の情報配信である．

スマッピーは携帯電話端末の利点を生かして，消費者のカード利用時の不便解消を目指しつつ，お店によるカードの有効活用ができるシステムである．また，地域経済活性化のために低価格で導入・利用ができることを念頭においている．

類似したシステムとしては, 次のようなものが存在する．(1) NTT ドコモによる ショッぷらっと[2]：iPhone と Android に対応, 2013 年 4 月現在，首都圏の加盟店が約 600 店舗登録されている．店舗にチェックインすることでポイント(start と呼ばれる)を集め，商品券やクーポンと交換できる．(2) アップルによる Passbook[3]：iPhone (iOS6 以上)であり，大手家電販売店，コンビニエンスストア，航空会社などで利用がなされている．クーポンや，ショップカード，イベントチケット，搭乗券などを一括管理できる．このように，首都圏限定のサービスであったり，携帯電話端末の機種が限定されている．さらに年間登録料や Passbook においてはアプリ開発同様の作業が必要であるなど，小規

†1 弘前大学大学院理工学研究科
Graduate School of Science and Technology, Hirosaki University

Vol.2013-CSEC-61 No.23
Vol.2013-IOT-21 No.23
2013/5/10

模小売店については参加のための敷居が小さくはない．

本稿では，低価格で導入・利用ができることを念頭に開発したシステム，スマッピーについて，そのデータベース部分の設計・開発を述べる．

2. スマッピー（スマートポイントシステム）

我々が提案するシステムについて，その概要とユースケース，ソフトウェア・ハードウェア構成について順次述べる．

2.1 システム概要

顧客と店舗は多機能携帯電話端末，スマートフォンを使用する．顧客についてはスマートフォンの替りにフィーチャーフォンによる利用も可能とする．図１に示すように，インターネット上に設置するスマッピーサーバーに，顧客情報，店舗情報，ポイントカード情報等を保持し，携帯電話端末へと情報提供をする．顧客が店舗で買い物をした場合，会計時に，ポイントカードの情報を QR コードで携帯電話の画面として提示する．店舗側はスマートフォンのカメラ機能によってこれを読み込み，カード情報を更新することでポイント追加をする(図１)．ポイント獲得に加えて，クーポン使用時にもこのように QR コードを利用する．

初期の利用者数として，消費者である顧客は 2,000 人，利用できる加盟店として店舗は 500 店を想定する．これは NTT タウンページよると青森県内飲食店の店舗数の約５％に相当する数である．

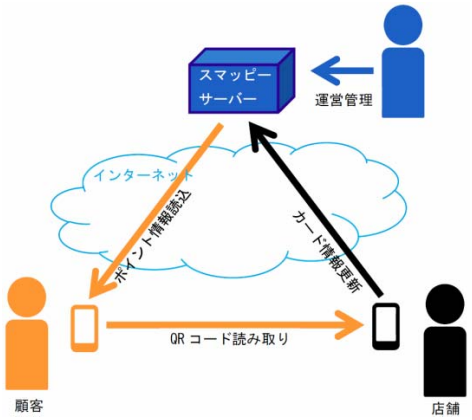

図 1　スマッピーシステムの構成
Figure 1　Overview of Smappy System.

2.2 ユースケース

主なユースケースを図２に示す．図中には説明の都合上，c1～c4 などの記号を付けた．利用登録 (図２ c1 参照) や店舗登録 (s1) は Web ホームページから行う．利用登録後，顧客は携帯電話用に独自開発したアプリからログインしたのちメニュー操作をする．例えば，ポイントカード管理 (c4) から店舗のポイントカードを発行する等を行う．ポイントカード管理 (c4) のサブメニューとしては次の項目がある：ポイントカードの発行・検索・一覧・利用履歴，最近使用したポイントカードの一覧表示，ポイント上限達成一覧，ポイント期限切れ間近一覧，店舗情報の検索・一覧，お気に入り登録．

一方，店舗登録済みの店舗利用者はログイン後，店舗情報を更新 (s2) する（ポイントカード，クーポン，お知らせ情報等の登録がサブメニューにある）．店舗が登録したポイントカードは顧客によっていつでも発行できる (c4 のサブメニューから) ことができる．ポイントを幾つ集めるとどんなクーポンを受け取れるのかという情報は，店舗が登録したポイントカードから知ることができる．その他，店舗はポイント付与数一覧など利用者統計を見る (s3) ことができる．

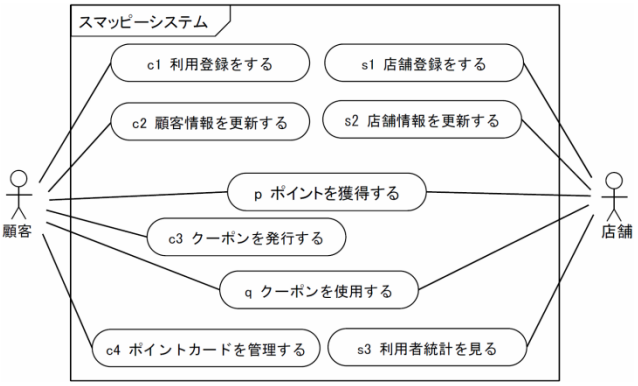

図 2　スマッピーシステムの主なユースケース
Figure 2　Use cases diagram of Smappy System.

2.3 ソフトウェア・ハードウェア構成

システムのソフトウェアは，顧客が使う携帯電話端末用のアプリ，店舗の携帯電話端末用のアプリ，Web サーバーとデータベースサーバーから成る．アプリはいずれも下層で Web サーバーにアクセスすることで php コードを実行させ，データベースと情報をやり取りする．携帯電話端末のアプリ及び php プログラムは，弘前大学理工学研究科深瀬らが設計・開発を行った[4].

スマッピーサーバーは，株式会社ビジネスサービス(KBS)[5]によるレンタルサーバー（仮想，CentOS 6.3, HDD 100GB，メモリ 2GB）を使用し，無料で利用できる Apache 2.2, php 5.3, MySQL Community Server 5.5 [5]を使用した．レンタルサーバーでは ftp，http，https の各プロトコルポートが利用可能である．mysql のポートは開発者側の IP アドレスからのみアクセス許可をし，開発時に利用した．

実質上，低コストなレンタルサーバー代と運営管理のための人件費でサービスを運用できる．

3. データベース

スマッピーシステムが使用するデータベースについて行った設計と開発，使用例について述べる．

3.1 設計・開発

ユースケースを検討するもとに必要なデータを洗い出して ER 図を作成した．とりわけ次の事柄に配慮をした．なじみやすさ・理解のしやすさを考慮し，従来の紙製ポイントカードと同様のポイントカードを利用できること．ポイントカードには，何ポイントで何のクーポンが発行できるかを自由に設定できること．不特定多数を対象としたクーポンは取り扱わないが，0 ポイントでクーポンを発行できること．また，ひとつの店舗が複数種類のポイントカードを発行できること（旧ポイントカードから新ポイントカードへの変更もこれで実現する）．ポイントを顧客に与えた日時（店舗利用時間に対応する）やクーポンの発行・使用日時を記録すること（情報は集計して店舗の経営戦略に利用）．一方では，電子的なメリットを生かして，ポイントカードの拡張である "ポイントキュー" を利用できるようにした．従来のポイントカードはカード毎に期限切れが設けられている場合が多い．ポイントキューは記録できるポイント数に上限を設けない無期限のポイントカードである．あまり利用しない顧客にとってはポイントが無駄にならずにリピート率が高くなることが期待される．

ER 図を作成後，データベースに載せる第三正規形のテーブル（いわゆるリレーションスキーマ）に調整した．最終的なテーブルを ER 図として図 3 に示す．

データの整合性を保つためには，主キー制約，外部キー制約はもちろんのこと，チェック制約（テーブルに入る値が定義域を満たすか否かのチェック）も考慮した．主キーは図 3 の実体記号内にある一つまたは複数項目の組で示す（主キーとそれ以外は横線で区切られている．横線の上にある項目が主キー）．外部キーは項目名に "(FK)" を付けて示す．チェック制約は MySQL 5.5 では使用できないがトリガーを利用して実現した．トリガーは，テーブル操作のイベントで実行する一連の SQL 文である．チェック制約を含めて，テーブルの各項目について行ったコード設計の一部を表 1 に示す．

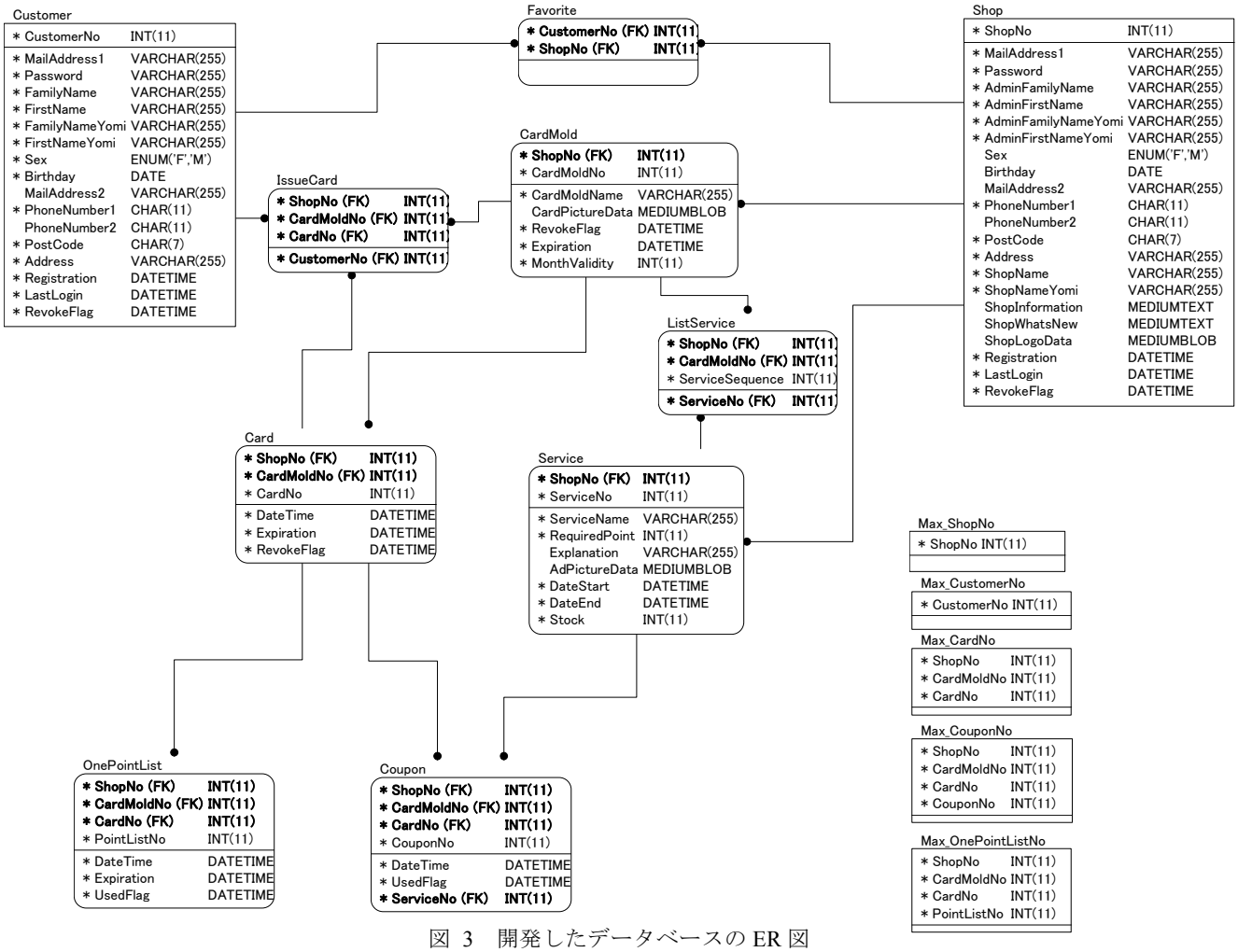

図 3　開発したデータベースの ER 図

Figure 3　ER diagram of developed database.

表 1　テーブルついてのコード設計（一部）

Table 1　Code design for some tables of developed database.

テーブル（意味）	項目[#1]	データ型	長さ	必須	チェック制約	備考
Card（ポイントカード）	**ShopNo**	INT	11	Y		主キーかつ外部キー
	CardMoldNo	INT	11	Y		主キーかつ外部キー
	CardNo	INT	11	Y	正数６桁まで	主キー
	DateTime	DATETIME	-	Y		発効日
	Expiration	DATETIME	-	Y		0 は期限無し
	RevokeFlag	DATETIME	-	Y		破棄(削除)日時．0 は利用中
CardMold（ポイントカードの鋳型）	**ShopNo**	INT	11	Y		主キーかつ外部キー
	CardMoldNo	INT	11	Y	正数２桁まで又は 0	主キー．店舗が自由につける．0 はポイントキュー
	CardMoldName	VARCHAR	255	Y		
	CardPictureData	MEDIUMBLOB	-	N		ポイントカードの背景画像
	RevokeFlag	DATETIME	-	Y		破棄(削除)日時．0 は利用中
	Expiration	DATETIME	-	Y	0 又は日時．MonthValidity が 0 でない時は 0	カード発行時に利用するデータ．カードの失効日または有効月数を保存．両方 0 は期限無し
	MonthValidity	INT	11	Y	0 又は正数．Expiration が 0 でない時は 0	
ListServer（ポイントカードの鋳型とサービスの対応）	**ShopNo**	INT	11	Y		
	CardMoldNo	INT	11	Y		
	ServiceNo	INT	11	Y		
	ServiceSequence	INT	11	Y	正数６桁まで	1,2,3,…N までの順番（サービスで与えるクーポンの順番を指定）．ポイントキューの場合は便宜的な順番．

[#1] 下線は主キー，太字は外部キーを示す．

データベース内のデータに対するセキュリティー保護は次のようにした．セキュリティーを高めるために，Web 上で実行する php コードで疑似的な個人認証（全権を持つ php コードを使ってログイン認証した利用者に応じて実行制限を課す）は行わないこととした．これは将来的なコードの拡張により脆弱性が現れることを避けるためである．我々は利用者毎にデーターベースユーザーを作成することで，データベースサーバー自体によりデータに対するアクセス制限を行った．ただし，MySQL ではデータベースユーザー毎にテーブル毎のアクセス制限を設定する事はできるが，これでは不十分である．目的のアクセス制限を実現するために，データベース上のテーブルから，各データベースユーザーだけが利用する許された行や項目だけから成るビュー表（一つまたは複数のテーブルから作られる仮想テーブル）を作った．データベースユーザーには自分のビュー表だけをアクセスする権限のみを与えて，ビュー表を通してデータの挿入・更新等を行えるようにした．ビュー表により粒度の細かい指定が可能となった．

顧客または店舗が，利用登録または店舗登録をする時に，利用者のデータベースユーザーとビュー表を作成する．データベースユーザーは 10 桁の数字に，顧客の場合は"uc"，店舗の場合は"us"の文字を先頭に付けたユーザー名とした．ビュー表は usXXXXYYZZZZ_Customer のように，ユーザー名に元のテーブル名を付けた名称とした（XXXXYYZZZZ は実際には数字の並び）．利用登録・店舗登録は，登録用のデータベースユーザーをひとつ用意し，Web ページの要求により登録用 php コードが実行されることで遂行される．

利用登録・店舗登録で必要になる一連の SQL 文は決まり決まったものであるため，ストアドプロシージャという形でデータベース内に SQL コードを保存し，データベースを使うクライアントプログラムからは call 文で実行できるようにした．また，クライアントプログラムの利便性のために，新規データ挿入時に必要な番号（顧客番号，店舗番号，カード番号，クーポン番号，ポイント番号）を自動生成するストアドプロシージャも作成した．図３で右下に孤立している４個のテーブルはこのストアドプロシージャが利用するものである．これらストアドプロシージャの実行制限も適切に設定した．

以上の事柄は，ソフトウェア SI Object Browser ER version 7.2 [7]と mysql コマンド（MySQL のフロントエンド）を使用してコード化した．SI Object Browser ER でテーブル（データ型，主キー制約，外部キー制約，及び空値の許可・不許可の指定含む）を作成し，フォワードエンジニアリングにより MySQL サーバ上にテーブルを構築した．フォワードエンジニアリングにより約 250 行の SQL 文が実行された．

トリガーとストアドプロシージャは mysql コマンドから SQL 文を直接実行することで定義した．トリガーは各テーブル（合計 10）のデータ挿入（insert）とデータ更新（update）に対応して，合計 20 個定義した（SQL コードは合計約 400 行）．クライアントプログラムが利用するストアドプロシージャの SQL コードは合計約 350 行となった．

3.2 使用例

開発したデータベースを使用した例として，顧客が店舗でポイントを獲得する場合を載せる．顧客および店舗により実行されるクライアントプログラムは次のステップをふむ．必要な情報はビュー表を使った演算で求めることができる（斜体で SQL コード例を示した）．

1) 顧客がスマッピーサーバーにログイン認証して接続
2) ポイントカードカード一覧を調べる

select ShopNo, CardMoldNo, CardNo from 顧客ユーザ名_Card where RevokeFlag = "0000-00-00 00:00:00" and (Expiration > 現在日時 or Expiration = "0000-00-00 00:00:00");

3) 一覧からひとつ選択したのち，ShopNo，CardMoldNo, CardNo で構成される QR コードを表示
4) 店舗がスマッピーサーバにログイン認証して接続
5) トランザクション開始
6) ポイントカードに記録可能なポイント数上限を確認（読取った QR コードを使用(以下の値１と値２)）

select sum(RequiredPoint) from 店舗ユーザ名_Card natural join 店舗ユーザ名_ListService natural join 店舗ユーザ名_Service where CardMoldNo=値１ and CardNo = 値２ group by ShopNo, CardMoldNo, CardNo;

7) ポイントカードの記録済みポイント数を調べる

select count(*) from 店舗ユーザ名_OnePointList where CardMoldNo = 値１ and CardNo = 値２ and (Expiration = '0000-00-00 00:00:00' or Expiration >= 現在日時);

8) 記録可能な場合に１ポイントを挿入する（２ポイント以上の場合は繰り返す）

call GET_OnePointListNo(店舗番号, 値１, 値２, 返値);

図 4　スマッピー利用中のスマートフォン
Figure 4　Smartphone using the developed system.

insert into 店舗ユーザ名_OnePointList (ShopNo, CardMoldNo, CardNo, PointListNo, DateTime, Expiration, UsedFlag) values (店舗番号, 値１, 値２, 上記返値, 現在日時, '0000-00-00 00:00:00', '0000-00-00 00:00:00');

9) トランザクションコミット

3.3 全体の動作検証

システム全体の動作を検証するため，スマートフォン上のアプリ[4]を使用し，ユースケースについて動作を確認した．スマートフォンの画面を図４に示す．

4. 開発したデータベースの評価

開発したデータベースは，セキュリティー上の考慮から多数のビュー表を使用している．想定する利用者数（顧客 2,000，店舗 500）の場合，データベースユーザーが 2,500 となる．各ユーザに 10 個のビュー表が作られるため，合計 25,000 のビュー表が作成されることになる．また仮に，各顧客が 10 枚のポイントカードを作成したとし，各カードに 10 ポイントが記録されている状況では，20 万（2,000×10×10）のポイントとしてのデータが OnePointList テーブルに収められ，これを元に各ユーザのビュー表が構成されている．この様なビュー表を使ってデータの挿入・更新・検索等が行われることになる．時には複数のビュー表が結合演算されて使われうる．

開発したデータベースの性能評価のために，次の主要な４つのユースケースについて動作時間を調査した．

・ポイントカードを管理する (図２ c4) のサブメニューから ポイントカードを発行する
・ポイントを獲得する (図 2 p)
・クーポンを発行する (図 2 c3)
・クーポンを使用する (図 2 q)

データベースには，顧客 200 人 店舗 50 店から，顧客 20,000 人 店舗 5,000 店（初期想定利用者の 10 倍）までのテストデータを，顧客/店舗の比を保ったまま，最初は顧客 200 人毎，続いて 400 人毎，4,000 人毎にデータを入れた状態で動作時間を測定した．

テストデータ値は次のようにした．各店舗は 1 件のポイントカード鋳型（CardMold テーブルに登録される）を持つ．このカード鋳型は 5 件のクーポンが発行できるとする（最初の 5 ポイントで一つ目，続いて 10 ポイント，10 ポイント，10 ポイント，15 ポイントでそれぞれクーポンを発行）．クーポンと引き換える商品やクーポンの発行順序等は Service と ListService テーブルに登録される（図３）．一方、顧客は登録されている店舗のうち一様乱数で選んだ 10 店舗からそれぞれ 1 枚ずつ，合計 10 枚のポイントカードを持つとする（Card と IssueCard テーブルに登録）．各ポイントカードには既に 25 ポイント集まっており（OnePointList テ

ーブルに登録），1 の件クーポンを発行（Coupon テーブルに登録）した状態とした．なお，顧客名や住所，電話番号などは適当な長さのランダムな文字列を使った．

テストデータ入力とユースケースの実行には簡単のために全て perl スクリプトにより行った．各ユースケースでデータベースを使用するときの手順を以下に記す．

ポイントカードを発行する

1) 顧客がスマッピーサーバーにログイン認証して接続
2) トランザクション開始
3) ポイントカード鋳型の有効期限を確認
   select RevokeFlag, Expiration, MonthValidity from 顧客ユーザ名_CardMold where ShopNo = 店舗番号 and CardMoldNo = カード鋳型番号;
4) ポイントカードを発行する
   call GET_CardNo(店舗番号, カード鋳型番号, 返値);
   insert into 顧客ユーザ名_Card (ShopNo, CardMoldNo, CardNo, DateTime, Expiration, RevokeFlag) values (店舗番号, 1, 上記返値, 現在日時, 有効期限, '0000-00-00 00:00:00');
   insert into 顧客ユーザ名_IssueCard (ShopNo, CardMoldNo, CardNo, CustomerNo) values (店舗番号, 1, 上記返値, 顧客番号);
5) トランザクションコミット

ポイントを獲得する

3.2 節 使用例 で示した．

クーポンを発行する

紙面の都合上割愛する．

クーポンを使用する

1) 顧客がスマッピーサーバーにログイン認証して接続
2) 所持するクーポン一覧を調べる
   select ShopNo, CardMoldNo, CardNo, CouponNo, ServiceNo from 顧客ユーザ名_Coupon where UsedFlag = "0000-00-00 00:00:00";
3) 一覧からひとつ選択（測定時は一様乱数で選択した．以下の値１～値４．なお，実際の携帯電話端末使用時は QR コードにより店舗は選択情報を知る）
4) 店舗がスマッピーサーバーにログイン認証して接続
5) トランザクション開始
6) 利用期間を確認
   select DateStart, DateEnd, Stock from 店舗ユーザ名__Service where ServiceNo = 値１;
7) クーポンを使用済みにする
   update 店舗ユーザ名_Coupon set UsedFlag = 現在日時 where CardMoldNo = 値２ and CardNo = 値３ and CouponNo = 値４;
8) トランザクションコミット

動作時間の測定結果を図５に示す．結果は各ユースケースにおいて，スマッピーサーバーへの接続（データベースへのログイン接続）からひと通りの処理を行い，接続切断までの全実行時間である．それぞれ 10 回の平均値をグラフで示した．顧客数の増加による実行時間の増加は認められない．初期想定利用者の 10 倍でも開発したデータベースの使用において実行時間の問題は生じない．平均前の個々の実行時間からは，しばしばサーバーへの接続に時間を要する場合があった．例えば，顧客数 1,000 人のとき，ポイントを獲得する場合のユースケースは 0.6 秒程で処理が終わるが，接続時に約 1.0 秒を待たされる場合が 2 例ほどあった．

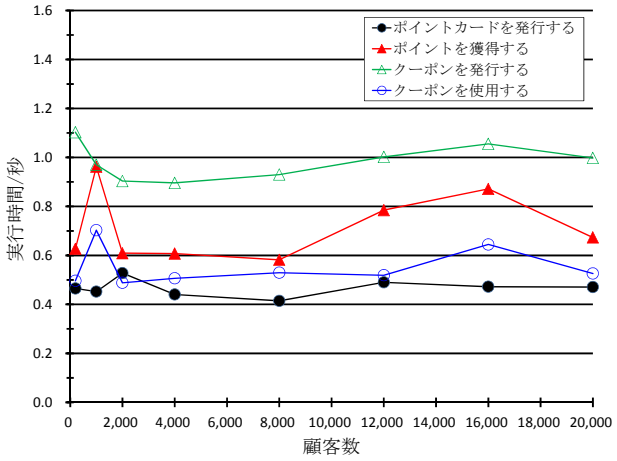

図 5 ユースケースの実行時間

Figure 5 Elapsed time of use case.

5. まとめ

ポイントカードシステムを電子化したスマッピーシステムで使用するデータベースを設計・開発した．設計にはデータ整合性やセキュリティーに配慮したアクセス制限を十分にほどこすために，トリガーやビュー表を有効に活用した．動作時間の面からも十分に，一般の利用に供することができる．スマッピーシステムとしては，実際の店舗現場での運用試験を残すのみである．

謝辞 本研究は合資会社インターゾーンにより企画・立案された，あおもり産業総合支援センターによる 24 年度上期助成事業の一環で行われた．

参考文献

1) 合資会社インターゾーン, http://www.inter-zone.co.jp/
2) 株式会社 NTT ドコモ ショッぷらっと, http://shoplat.net/
3) Apple Inc. Passbook, http://www.apple.com/jp/ios/whats-new/#passbook
4) 深瀬政秋ら: 発表予定
5) 株式会社ビジネスサービス, http://www.kbs-web.com/
6) Oracle Corp. MySQL, http://dev.mysql.com/
7) 株式会社システムインテグレータ SI Object Browser ER, http://siob.sint.co.jp/er/