Implementing a loyalty program using Windows Phone 8 Wallet

This article explains how to Implement a Loyalty Program using Windows Phone Wallet.

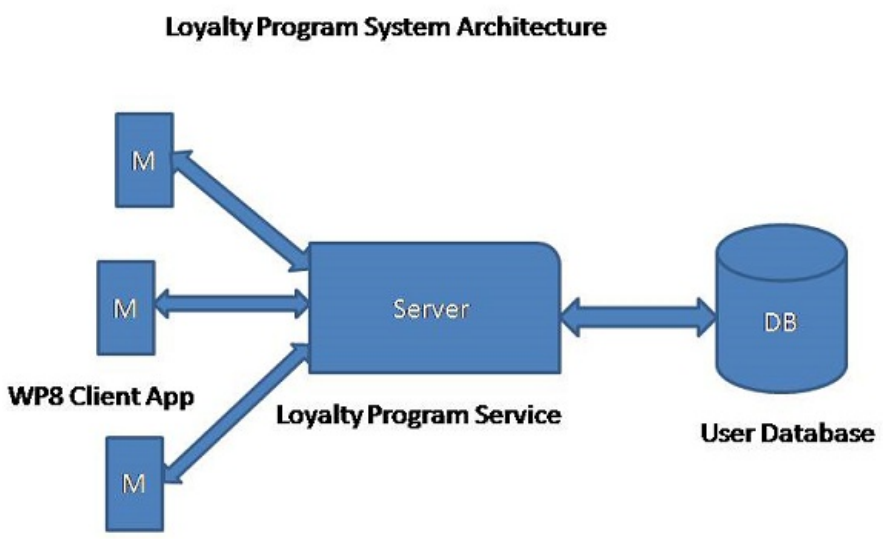
Introduction

On Windows Phone 8 (WP8) we can very well integrate our own apps dealing with financial instruments and aspects. One of the wide use of Wallet will be seen in implementing loyalty programs by various service providers including Hotels restaurants to book shops and coffee shops where we visit regularly. Even online shopping will see various possibilities in using Wallet.

One of the best thing about WP8 wallet is you can make your physical wallet lighter and stop carrying coupons, membership cards or loyalty cards. Currently what we see is Loyalty Cards are implemented with the help of something like a smart card which you have to carry or use the details like your ID etc while making purchase. Remembering or knowing the loyalty points is not easy process.

Typically Loyalty Cards work like this, user makes a purchase and some points are added to his account, these points can be used for future purchase discounts or sole purchase if allowed. Same can be implemented with the help of the third party service along with a WP8 app which is Wallet enabled. This article explains how a Loyalty Program can be implemented for an online shop like eBay or Amazon.

Solution



The overall solution consists of three major components

1. *Windows Phone 8 app*
2. *Loyalty Program Service*
3. *Loyalty Program Database*

1. Windows Phone 8 app

This is the front end with which user will interact. WP8 app for loyalty Program will have to provide few mandatory features.

1. *User authentication using Loyalty Program service : The loyalty program will be associated with an online store. User will have to register with them in order to track the points. App will let the user make online purchases using either a credit card or redeem the Loyalty Program points from Wallet. To protect app from illegal access the app must use password protection.*
2. *Integration with wallet to use it for payments.*
3. *Let the user browse the items and purchase them.*
4. *Add the loyalty points to the membership in wallet.*

We will discuss the complete app design in a section below

2. Loyalty Program Service

(C) Copyright Nokia 2014. All rights reserved.

Loyalty program service will reside on some server backed by a database. The service can be implemented using any web service technique. Simplest possible implementation will be a set of PHP webscripts running on Apache server, https is optional. The Loyalty Program Service has to provide minimum following functions

1. *Save loyalty points based on users purchases*
2. *Get the status of user loyalty points*
3. *Let use the user loyalty points for future purchases along side other instruments.*

The Loyalty Program Service will have wider scope based on the actual and total feature requests. Above are few representative ones and important.

3. Loyalty Program Database

The database will store user information along with the earned and redeemed loyalty points per user.

Windows Phone 8 App design requirements

The app to track and use Loyalty Points earned through program needs to implement some basic Wallet operations.

1. *Add the Loyalty Program membership to Wallet in the form of a Loyalty Card once user logs in using his credentials first time*
2. *Query the service for current status of Loyalty Points for the user.*
3. *Initialize the Loyalty Points to the Loyalty Card based on the details sent by the service.*

To do this following code can be used

Below code shows how a loyalty card item can be created in Wallet with Loyalty Points Balance. Before the Loyalty Card can be added app will have to get the user ID and query for the Loyalty Points with service by connecting it with appropriate URL and provide necessary parameters, a simple service will ask for user ID and provide the details in response

```
         try
            {
         LoyaltyCard loyaltyCard = GetLoyaltyStatus(id);
                WalletTransactionItem lCardItem;
                lCardItem = new WalletTransactionItem("lcard");
                lCardItem.IssuerName = “ XYZ Loyalty Program”;
                lCardItem.DisplayName = " XYZ Loyalty Program";
                lCardItem.IssuerPhone.Business = MockWebService.WebService.IssuerPhone;
                lCardItem.CustomerName = loyaltyCard.customerName;
                lCardItem.AccountNumber = id;

                lCardItem.IssuerWebsite = new Uri(“xyz.com”);
                lCardItem.DisplayAvailableBalance = loyaltyCard.balance;
                BitmapImage bmp = new BitmapImage();
                using (System.IO.Stream stream =
Assembly.GetExecutingAssembly().GetManifestResourceStream("LoyaltyProgramLogo.jpg"))
                    bmp.SetSource(stream);

                lCardItem.Logo99x99 = bmp;
                lCardItem.Logo159x159 = bmp;
                lCardItem.Logo336x336 = bmp;
  AddItemWalletItemTask addWalletItemTask = new AddItemWalletTask();
                addWalletItemTask.Item = lCardItem;
                addWalletItemTask.Show();
            }
            catch (Exception ex)
            {
                MessageBox.Show("There were the following errors when saving your
membership to the wallet: " + ex.Message);
            }
```


(C) Copyright Nokia 2014. All rights reserved.

While adding the membership to wallet you will also need to provide a Logo image of service with various resolutions to suit various screen types. Once the membership is created successfully user can see it added in the Wallet. In wallet the Loyalty Card will show up as a membership in “All” section with available Loyalty Points. If user clicks on the membership the related App will be launched by Wallet.

As now the membership is added to Wallet the status of Loyalty Points will be updated in Wallet anytime user makes purchase using app.

Use following code to load the loyalty card details,update and save back to wallet

```
         WalletTransactionItem item = Wallet.FindItem(“lcard”);
         int newBalance  = int.Parse(item.DipslayAvaiableBalance)  + earnedPoints;
         Item.DisplayAvailabelBalance = newBalance; // after purchase service will
return this
         Item.SaveAsync(); // save the current card to wallet after purchase is over
         LoyaltyCard loyaltyCard = GetLoyaltyCard(id) // user service to get the current
users card
         lolaytyCard.balance = newBalance;
         UpdateLoyaltyCard(loyaltyCard);
```

Above code explains how the Loyalty Point balance can be updated in Wallet. Functions like *GetLoyaltyCard, UpdateLoyaltyCard and class LoyaltyCard* are a way to talk to WebService and needs to be implemented and are specific to app context, here references are being added to complete the example code.

The WP8 app needs to provide some additional intelligence, while purchasing an item using app and backend online store services, app must check for Loyalty Card balance in Wallet then if the balance qualify for some discount or free purchase through redemption of points then notify the user about the situation.

Application will show two options mainly while purchasing an item online

1. Buy with Credit Card from Wallet”
2. Buy with Credit Card and Redeem Loyalty Points

If selected 1 app will complete the transaction using available credit card in wallet as follows Load all the cards from Wallet using async method GetItemsAsync from Wallet class, this method will return all the PaymentInstrument types of items if called as follows

```
        GetItemsAsync() as PaymentInstrument;
```

Above method asynchronously reads all credit and debit cards and bank accounts from wallet, App will show list of all returned cards and let the user select a card to process payment. Call a WebService function to process payment . This is again specific to the implementation.

If selected 2 app will show the available Loyalty Points and takes further input from user about purchase i.e. how many points user wants to redeem, also shows all payment instruments available including credit cards, debit cards, bank accounts as discussed above using Wallet API and then processes payment using WebService function.

At this stage the Loyalty Card must be updated in Wallet upon confirmation on payment from WebService. The code is same as discussed above.

Conclusion

There are various use cases possible with windows phone 8 wallet, one such use case has been discusses in this article. *Windows Phone app plays an important role in above example letting user see all his loyalty points at one place and redeem them from app, no need to remember user ID and visit the website to redeem them.*

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.