

COUPON REDEMPTION SYSTEM

A THESIS IN
Computer Science

Presented to the Faculty of the University
Of Missouri Kansas City In partial fulfillment
Of the requirements for the degree

MASTER OF SCIENCE

By
RAM SHASANK PATHATH

B.Tech. GITAM UNIVERSITY, INDIA 2014

Kansas City, Missouri
2015

©2015

RAM SHASANK PATHATH

ALL RIGHTS RESERVED

COUPON REDEMPTION SYSTEM

Ram Shasank Pathath, Candidate for the Master of Science Degree

University of Missouri – Kansas City, 2015

ABSTRACT

A number of trends in e-commerce have become evident in recent years. Online storefronts have proliferated, and advertisements for these storefronts in the form of click through banners have become popular. Recently, the Internet has also become a distribution medium for conventional coupons. The customer prints out such coupons with his or her printer and redeems them like any other coupon. Although coupons are one of the most widely used forms of sales promotion, they suffer from a number of disadvantages. First, conventional distribution channels are slow and have long lead-times. Hence, distributing coupons requires much advance planning. Second, coupons were being redeemed multiple times by multiple people—sometimes well after the expiration dates. Coupons could be possibly redeemed by employees. Third, lower cost distribution channels such as newspapers do not lend themselves to precise targeting. . A new approach to redemption, however, offers a simple solution for ensuring that each offer is redeemed according to plan.

According to our approach chain store owner can choose to generate coupons for a specific branch or branches. When customer prints the coupons, the customer's name can be affixed to the coupon such that others cannot use the coupon. This way prevents store employees from redeeming the coupons.

Store owner can choose to provision a limited number of coupons or to set the number of coupons to be distributed as unlimited. He or she can terminate the provisional coupon issuing instantly.

In this new scenario, each promotional offer is assigned an authenticated coupon code. The code can be assigned specific parameters, such as one-time use, an expiration date and future activation. When a coupon or other promotion is redeemed, the coupon code is moved from active state to used state. Whenever a user tries to redeem the same coupon again, it checks in the database and whether the state of coupon is active or used.

APPROVAL

The faculty listed below, appointed by the Dean of School of Computing and Engineering, have examined a thesis titled “Coupon Redemption System” presented by Ram Shasank Pathath, candidate for the Master of Science degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Yijie Han, Ph.D., Chair
School of Computing and Engineering

Xiaojun Shen, Ph.D.
School of Computing and Engineering

Praveen Rao, Ph.D.
School of Computing and Engineering

CONTENTS

ABSTRACT	iii
ILLUSTRATIONS	viii
ACKNOWLEDGEMENTS	x
CHAPTER	
1. INTRODUCTION.....	1
1.1. History	1
1.2. Delivery methods	1
1.2.1. Print media	2
1.2.2. Internet coupons	2
1.2.3. Mobile coupons	3
1.2.4. Mobile app coupons	4
1.3. Why coupons	4
1.4. Mobile coupon/Rebate campaign best practices	4
1.5. How to redeem coupons	5
2. OUR APPROACH	
2.1. Introduction to coupon redemption	8
2.2. Two key attributes	9
2.3. How the solution works	9
3. ANALYSIS AND WORKING	
3.1. Application overview	11
3.2. Working	13
3.2.1. Store owner	13

3.2.2. Customer	15
3.2.3. Employee.....	18
4. CODE	
4.1. Storeowner	20
4.2. Customer	24
4.3. Employee	29
5. CONCLUSION.....	32
6. FUTURE WORK.....	33
REFERENCES	34
VITA	36

ILLUSTRATIONS

Figure	Page
1. Coupon redemption system	5
2. Barcode redemption	6
3. QR code scanner	7
4. Store owner registration form	13
5. coupon provision system for store owner.	14
6. Dialogue showing coupon generation complete	14
7. Customer registration form	15
8. Customer login form	16
9. Coupon dropdown box for the customers	16
10. Barcode generation	17
11. customer registration storage database table	17
12. Coupon details storage database	17
13. User and coupon connectivity to barcode form	18
14. Android application start page.	19
15. Barcode scanner.	19
16. Store owner registration page	20
17. html code for coupon generation by store owner	21
18. Perl code for storing registration data into database	22
19. Perl code for coupon generation by store owner	23
20. html code for customer registration	24
21. html code for coupon generation by the customer	25

22. Perl code for storing customer data into database	26
23. Perl code for generation of Unique barcode	27
24. Android application code for connecting to database	29
25. code for connecting the server with the android application	30

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank the following people who have directly or indirectly helped me in academic achievements. Firstly, I would like to thank Dr. Yijie Han, my mentor and advisor, for his continuous support and guidance throughout my master's program in computer science. Without his tireless effort and advice, this personal achievement wouldn't be possible. I sincerely thank Dr. Xiaojun Shen and Dr. Praveen Rao for accepting to be a part of my thesis committee and making time for me from their busy schedule.

I would like to dedicate my thesis to my parents who constantly inspired me to pursue higher studies. I would like to thank my family, my dad Mr. Krishna Mohan Rao Pathath, my mom Mrs. Subba Lakshmi Pathath, my uncle Mr. Sudhakar rao Pathath, my aunt Mrs. Lakshmi Pathath and my brother Mr. Hemanth Kumar Pathath, who stood behind me all these years during my degree.

Finally I would like to thank all my teachers, educational administrators, present and past and all who helped me achieve this academic goal.

CHAPTER 1

INTRODUCTION

1.1 History

In 1886 the Coca-Cola Company was incorporated in Atlanta, Georgia, with Asa Candler as one of the partners. He transformed Coca-Cola from an insignificant tonic into a profitable business by using innovative advertising techniques. The key to this growth was Candler's ingenious marketing including having the company's employees and sales representatives distribute complimentary coupons for Coca-Cola.¹ Coupons first saw widespread use in the United States in 1909 when C. W. Post conceived the idea to help sell breakfast cereals and other products. Today more than 2,800 consumer packaged goods companies offer coupons for discounts on products. In 2011 U.S. consumers used coupons to save 4.6 billion dollars on their purchases of consumer packaged goods.

1.2 Delivery methods

Customers may get these coupons from various sources, including national newspapers and the Internet, with web sites offering free printable grocery coupons can be printed at home and use them at retail store. Some major grocery chains also produce digital coupons that may be loaded onto the retailer's loyalty-card at home, or at a coupon dispensing machine located in store [1]. In 2011, the top five vehicles for distributing consumer packaged goods coupons in the U.S. were: the Free Standing Insert, a coupon booklet distributed through newspapers and other sources (89.4%); in-store distribution (4.2.%); direct mail (2.3%); magazines (1.5%); and coupons distributed on or in product packaging (1.3%). Other distribution methods together accounted for less than 2% of all coupons distributed. There are coupon-providing

websites that provide customers with coupons of various stores. These sites accumulate coupons from various sources.

1.2.1 Print media

Clipping coupons from newspapers has been the most popular way to obtain coupons, though Internet and Mobile Phone coupons are gaining wide popularity. Based on its annual Redplum Purse String Study surveying more than 23,000 shoppers, a coupon provider, Redplum, reports 76% of coupon-seekers utilize newspapers as their primary source for coupons and deals; 59% find coupons in e-mails and coupon alerts, a 29% increase from 2010; and 33% use Internet searches to find coupons.

Some retailers and companies use verification methods such as unique barcodes, coupon ID numbers, holographic seals, and watermarked paper as protection from unauthorized copying or use.

Other than newspaper, there are also coupon book publishers and retailers who compile vouchers and coupons into books, either for sale or free.

1.2.2 Internet coupons

Online retailers often refer to coupons as "coupon codes", "promotional codes", "promotion codes", "discount codes", "key codes", "promo codes", "surplus codes", "portable codes", "shopping codes", "voucher codes", "reward codes", "discount vouchers", "referral codes" or "source codes". Internet coupons typically provide reduced cost or free shipping, a specific dollar or percentage discount, or some other offer to encourage consumers to purchase specific products or to purchase from specific retailers. Because paper coupons would be difficult to distribute and redeem, typically secret words or codes are distributed for consumers

to type in at checkout [4]. Marketers can use different codes for different channels or groups in order to differentiate response rates.

1.2.3 Mobile coupons

A mobile coupon is an electronic ticket solicited and or delivered to a mobile phone that can be exchanged for financial discount or rebate when purchasing product or service [7]. Coupons are usually issued by manufacturers of consumer packaged goods or retailers, to be used in retail stores as part of a sales promotion. They are often distributed through WAP Push over SMS or MMS, through GEO Fencing technology or other mobile means. The customer redeems the coupon at store or online. In some cases, customers may redeem the mobile coupon at the point of sale. Some retailers may choose to forward the redemption to a clearinghouse for final processing.

What is unique about mobile coupons is the memory of information in the coupons often outlast the expiration dates of the coupons themselves, triggering actual purchases at later dates. Researchers suspect it is driven by the engagement generated by the mobile device. Mobile coupons are popular among U.S. fast-food customers. The primary success factors for the SMS campaigns are discount size, how the discount value is framed (as a gift or percent off) and the timing of the campaign.

Many retailers support the redemption of email and paper coupons via mobile devices. In addition to distributing such offers via their own email lists, SMS subscriptions, and apps, they are also often made available through coupon applications [2].

1.2.4 Mobile app coupons

A mobile app coupon can be either a regular coupon for redemption (discount value or discount percentage) used at checkout distributed by the app-makers or unique codes owned by each user to recruit new users with referral benefits (e.g. Airbnb, Uber). The latter form requires personal sharing on behalf of users to their personal networks.

1.3 Why coupons?

- Coupons serve as both an advertisement for your business or service and an incentive for subscribers to use the business or service you are advertising.
- Improve ease and efficiency of distribution by making it available to only legible users. The users can print coupons only when needed [7].

1.4 Mobile Coupon/Rebate campaign best practices-principles

The following five principles are intended to apply broadly across wireless platforms in which coupons may be distributed [6]. They are not meant to be overly restrictive or prohibitive, but to only provide a framework to those that distribute, develop, and publish coupons on wireless platforms. They are meant to help protect both consumers and businesses as well as help maintain the reliability and integrity across the coupon distribution industry.

1. Honest, True and Fair: All coupons placed in the market for distribution must be honest and in good taste. Coupons should be conveyed in a truthful and not deceptive or misleading manner.

2. Content and Placement: Good taste must always govern the content, placement and presentation of the coupon. All coupons should be prepared with proper consideration of the type of product being advertised and the audience to whom the advertising or promotions are directed.

3. Transparency: The corporate name, product name, or trade name should be identified, clearly, prominently and visually. Developers, publishers or the advertised products corporate entity should have a mechanism in place for consumer feedback on the coupon [3]. All rules that govern the promotion, i.e., expiration dates, limits, etc., should also be made clear to the consumer.

4. Price and Value: All prices and values must be confined to specific facts and presented in a prominent, fair and accurate manner.

5. Privacy and Security: Privacy should be respected and user consent, opt in, opt out and other.

1.5 How to redeem coupons

1. Coupon code redemption

The moment a coupon is claimed, a unique cookie ID is created and stored on the client's mobile device [11]. Once claimed, every **digital** or printed coupon carries a unique 6 digit coupon code. Just show the sales clerk the coupon code on your phone to get the discount. The unique 6 digit coupon code, that is embedded on every coupon, or is presented via print. A valid coupon code triggers the redemption, validation of the coupon.

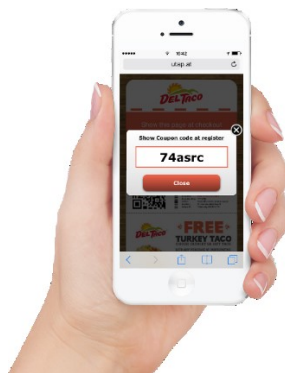


Figure 1: coupon redemption system

2. Barcode redemption

This system is ideal for Enterprises who use a printed coupon system and want to evolve digitally [11]. While our coupons have a unique barcodes digitally embedded on the coupon, vouchers can also be printed with the unique bar code embedded on paper, where validation takes place by scanning the Barcode off the coupon. By scanning our coupons with embedded barcodes you will be able to redeem and track mobile coupons in real-time, and integrate our API to your Point of Sale.

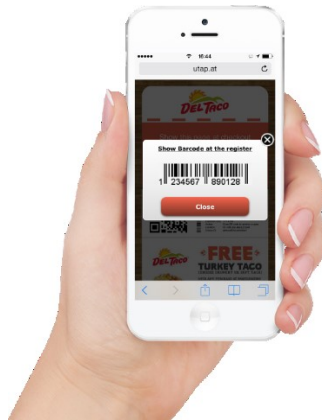


Figure 2: Barcode redemption

3. QR code redemption

This platform lets you turn mobile devices into enterprise-grade Coupon Scanners. Install a QR Code reader on any smartphone, and validate your client coupons at check out. At validation, the cashier scans the QR code on the client's phone. This connection generates a pin coded validation option on the cashier's mobile device. The correct pin code, releases the coupon validation [11]. At the same time, the client's mobile phone shows the 'processing' of the validation, and when the offer has been redeemed, the coupon shows a 'verified' mark, and it will no longer be valid.



Figure 3: QR code scanner

CHAPTER 2

OUR APPROACH

2.1 Introduction to coupon redemption

The stories of unexpected losses related to discounts and promotions seem to pervade every retail category and affect merchants of all sizes. A regional retailer issues a \$10 coupon and budgets \$450,000 for the campaign. The coupon—which was intended for specific customers—goes viral and the merchant is flooded with people redeeming the offer. The volume of redemption consumes the initial budget in four hours and wipes out the marketing budget for the remainder of the year.

The coupons were being redeemed multiple times by multiple people—sometimes well after the expiration dates. Big box retailers abandoned promotions because they could not find affordable and easy-to-add authentication.

Fraudulent redemption of coupons, daily deals and other offers catches merchants by surprise and turn promising promotions into painful, costly experiences [7]. While specifics vary from one merchant to another, the essential problem is an inability to authenticate these discounts and promotions at the point of sale. Complex authentication solutions that require extensive programming, integration or training would defeat the purpose of running promotions: driving incremental profits. A new approach to redemption, however, offers a simple solution for ensuring that each offer is redeemed according to plan.

In this new scenario, each promotional offer is assigned an authenticated coupon code. The code can be assigned specific parameters, such as one-time use, an expiration date and future activation. When a coupon or other promotion is redeemed, the coupon code is moved

from active state to used state. Whenever a user tries to redeem the same coupon again, it checks in the database and whether the state of coupon is active or used.

2.2 Two key attributes: transparency and accountability

Employee misuse of redeemed offers is a widespread problem for merchants in a variety of markets. This new authentication solution mitigates this risk by obscuring the redemption data from multiple users: coupon codes and other information travel between the database system and the host without being displayed [1, 3].

Furthermore, data related to specific promotions can be captured, sorted and compiled in a variety of standard or customized reports. This helps marketers and senior managers track the performance of specific promotions and make future investment decisions for their marketing dollars. It also provides a real-time view of costs while the promotion is in progress, which is clearly considered critical knowledge after the troubles experienced by merchants in so many campaigns.

2.3 How the solution works

Set-up for the authentication solutions is relatively simple. Merchants define approved coupon codes and assign them to specific promotions, with a unique identifier issued for each individual coupon. The processor receives these codes and downloads them to the host. When customers redeem coupons or other offers at the point of sale, the unique identifier is transmitted to the server for authentication [4]. If the offer has not yet been redeemed—and it is being redeemed within an approved period of time—the host applies the discount to the transaction and returns the coupon code to the merchant's system for use in tracking and analytics. A retailer's sales clerks are no longer required to make decisions regarding authenticity or expiration dates. They also have no access to promotion details and cannot

“recycle” offers. Customers are limited to the intended use of the offer—one time, in most cases. They also cannot distribute the offer to other consumers for multiple redemptions [5]. This same process is applied to both physical (paper-based) and virtual (mobile or card-based) programs.

Mitigating risk in marketing promotions This simple authentication solution addresses the long list of problems experienced by merchants—especially in recent years—as they have attempted to capitalize on the revenue generating benefits and popularity of daily deals, e-rewards, electronic coupons and other emerging marketing tactics[10]. It brings a new level of predictability and accountability to the marketing process and mitigates the risk of substantial losses caused by fraudulent use of coupons and other promotions.

CHAPTER 3

ANALYSIS AND WORKING

3.1 Application overview

Coupon redemption system is an application that helps the owners to create and publish the coupons in their database such that the coupons are visible to the customers. This application helps the customers to view and download the coupons

Before downloading or printing a coupon the customer has to register with the website customer registration, which means the customer enters his personal information to register before he/she will be able to download or print the coupon. Initially all the coupons created by the store owner are stored into the database. The coupons are visible to the customers after they log into their accounts. If the customer print a particular coupon present in the database then he will not be able to print the coupon again.

The employees in the store are provided with an android application which when run the barcode scanner scans the barcode present on the coupon and activates the coupon. Once the coupon is activated the offer is applied to the customer's payment. The barcode present on the coupon is generated for every customer separately and therefore once the coupon is activated the barcode is marked as used and therefore the customer will not be able to use it again.

The application consists of mainly three users:

Store Owner:

The main functionality of the store owner is that he is able to create and control the coupons generated. He is able to create a coupon with a start date and end date based on which the coupon can be differentiated whether the coupon is active, available or expired. Once the

coupon is created the coupon code is stored into the database with the coupon name, start date and end date.

Customer:

The customer or the user has to first register into the customer registration with all his login credentials. Once the login credentials are given and the user creates an account he can login to the system using his user name and password. The customer's home page consists of a dropdown box where the coupons which are all available for him are visible. The customer is able to print the selected coupon once. Once the coupon is printed the customer gets the coupon with a unique barcode on it. The customers have to be careful such that they don't lose the coupon. If the coupon is once printed by the customer then he will not be able to view the coupon again.

Employee:

The employee is provided with the android application developed which has the barcode scanner that can be used to scan the barcodes present on the coupons provided by the customers. The barcode once finished scanning compares it with the barcodes which are generated when the customers print their coupons. If the barcode matches with the barcode present in the database then the employee can activate the coupon. Once the barcode is activated the coupon cannot be used again.

3.2 Working

3.2.1 Store owner

The store owner has a registration page through which he has to submit all the data. The registration page is as appears on the image below. The store owner has to provide the details such as name, company name, address, email id. He has to also choose a username and a password as his login credentials.

Registration form

*Required Field	Enter Password
Enter Name	Re-type Password
Enter Company Name	Submit
Enter Address	
Enter Email	
Enter Username	

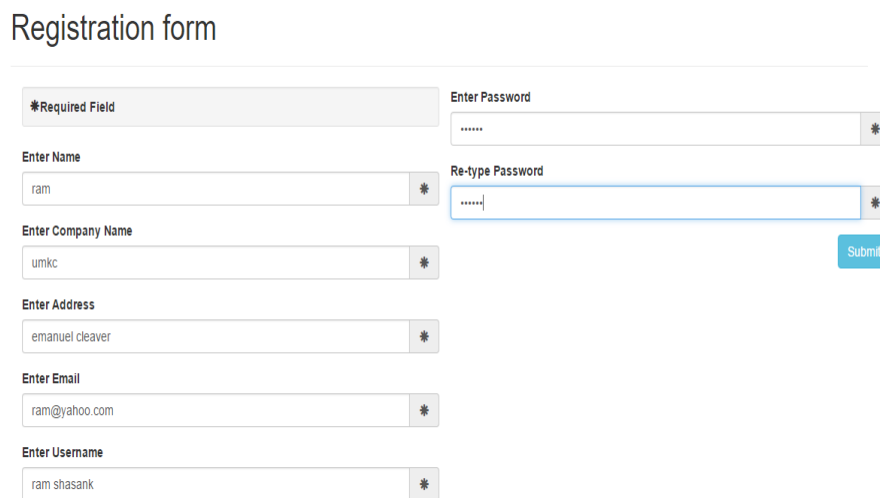


Figure 4: Store owner registration form

Once the registration part is done the store owner is directed to the coupon provision system where the store owner can create a coupon by entering the coupon name, coupon start date and coupon end date. The screen shot of the coupon provision system is provided below. The store owner can create limited number of coupons or unlimited number of coupons based on the requirement. If the owner want to generate unlimited coupons then he can leave it blank so that it generates unlimited coupons by default.

Coupon Provision system

*Required Field

Coupon Name
 *

Coupon Start Date
 *

Coupon End Date
 *

Available count for users (default will be unlimited)
 *

Figure 5: coupon provision system for store owner

Once the coupon is generated the store owner receives a dialogue box saying that the coupon is generated successfully.

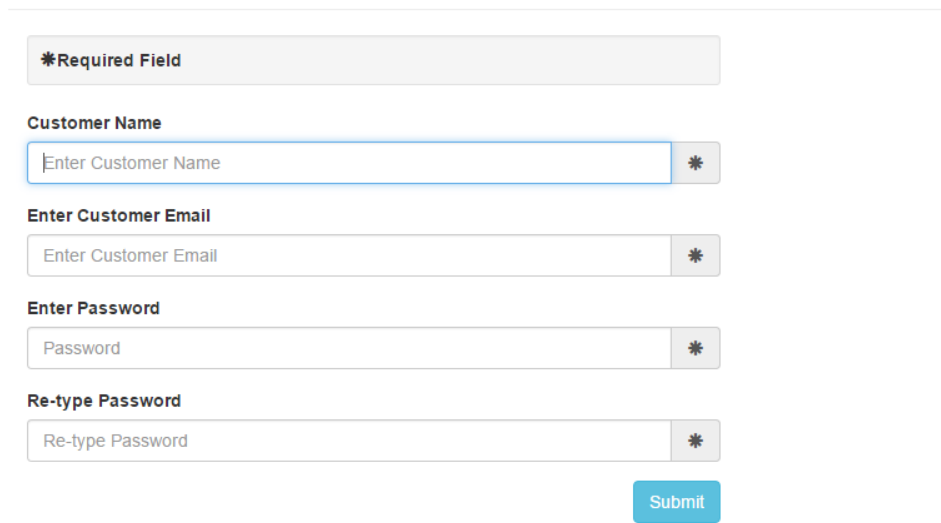
Coupon generated Successfully

Figure 6: Dialogue showing coupon generation complete

3.2.2 Customer

The customer has to fill a registration form for them to access the coupons the store is providing. In the process of registration the customer has to provide with his name, email and password. The data provided by the customer is stored into the database for the user to be able to login into the site. Once the registration is done the customer is redirected to the login page where he has to enter his login credentials.

Customer Registration Form



The image shows a customer registration form with the following fields and a submit button:

- A header box containing the text ***Required Field**.
- A label **Customer Name** above a text input field with the placeholder text "Enter Customer Name" and an asterisk icon on the right.
- A label **Enter Customer Email** above a text input field with the placeholder text "Enter Customer Email" and an asterisk icon on the right.
- A label **Enter Password** above a text input field with the placeholder text "Password" and an asterisk icon on the right.
- A label **Re-type Password** above a text input field with the placeholder text "Re-type Password" and an asterisk icon on the right.
- A blue **Submit** button located below the password fields.

Figure 7: Customer registration form

The Login page appears as follows. The customer enters his login credentials i.e his email id and the password to login. The customer also can be directed to the registration page by clicking the register button present at the bottom for new users. If the customers logs into his account then he is redirected into the generate coupon.

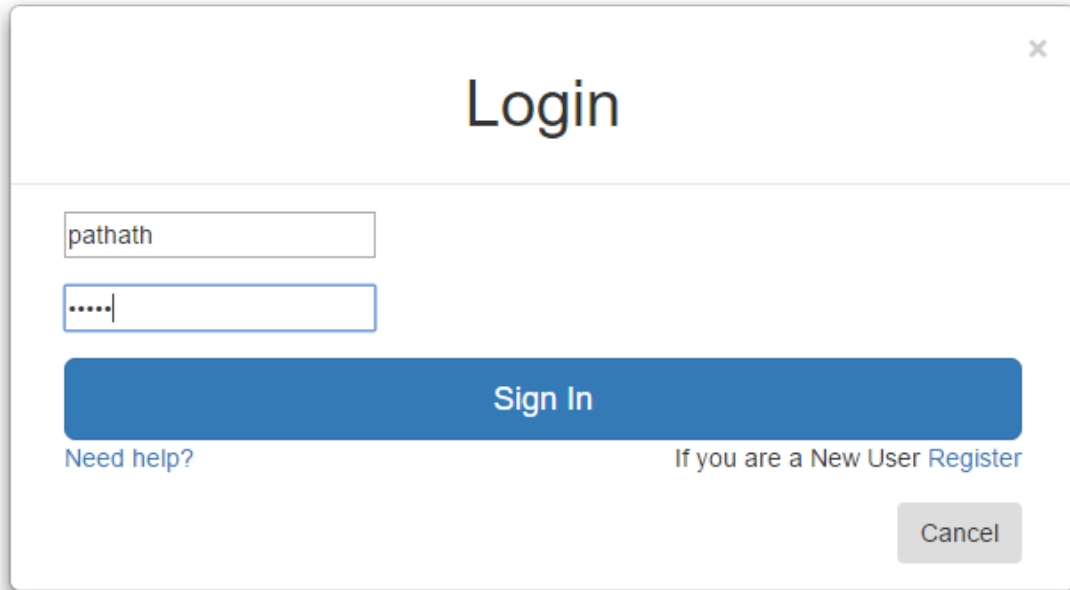
A login form titled "Login" with a close button (x) in the top right corner. It features two input fields: the first contains the text "pathath" and the second contains five dots. Below the fields is a large blue "Sign In" button. At the bottom left, there is a link "Need help?". At the bottom right, there is a link "If you are a New User Register" and a grey "Cancel" button.

Figure 8: Customer login form

The generate coupon page consists of a drop down box where all the coupons generated are visible and the customer can print the coupon. Once he prints the coupon then a unique barcode is generated with which the customer can use that particular offer.

A screenshot of the "Generate Coupon" page. In the top right corner, it says "Welcome pathath" with a "logout" button. The main heading is "Generate Coupon". Below it, there is a "Select a Coupon" dropdown menu with a list of options: "Select A Coupon", "Select A Coupon", "flat20", and "thanksgiving". To the right of the dropdown is a "Print a Coupon" button.

Figure 9: Coupon dropdown box for the customers.

The users table that is being stored into the database is as follows. The coupons once printed by the user are moved to the used Coupons table with which the system knows whether the coupon is available for the user or not. Once the coupon is printed by the customer it is moved to this table and the coupon is not found in the dropdown box. The barcode is generated on the coupon so that the barcode when scanned the user details can be obtained.

You have successfully printed the coupon. Your Coupon code is 459103228962. [Back](#) [logout](#)



Figure 10: Barcode generation

+ Options										
				oid	username	email	password			
<input type="checkbox"/>		Edit		Copy		Delete	867064	rams	rak@gmail.com	abc
<input type="checkbox"/>		Edit		Copy		Delete	332560	ramesh	ram@gmail.com	munna
<input type="checkbox"/>		Edit		Copy		Delete	230896	shasank	shasank@gmail.com	munna

Check All With selected: Change Delete Export

Figure 11: customer registration storage database table

The barcodes that are generated are stored into the coupons table under the allBarCodes column.

oid	couponName	startdate	enddate	allBarCodes	availableCount
995197	thanksgiving	2015-02-11	2016-02-11		-1
919318	christmas	2015-02-11	2016-02-11		-1
359763	thank50	0015-02-12	2016-02-11		-1
953345	christmassale	2015-12-25	2016-12-25		-1

Figure 12: Coupon details storage database

The users and coupons are connected to the barcode so that the user id and the coupon id can be compared to obtain the details of the customers when the barcode is scanned.

user oid	coupon oid	barCode
230896	995197	362956363750
		334867178365
332560	359763	286725893519
332560	919318	238944468922
332560	919318	361335798628
867064	953345	909797268146

Figure 13: User and coupon connectivity to barcode

3.2.3 Employee

The employees are provided with barcode scanner app for android mobiles. The barcode scanner app can be used to scan the barcode generated on the coupon. The android application is connected to the database through internet. The barcode once scanned the application compares the barcode with the barcodes which are generated in the database and checks whether the coupon is present in the used column. If the coupon is found in the used column it displays a dialogue box showing that the coupon is already used and cannot be used again. If the barcode is not in the used column then the application checks the start date and end date of the coupon and compares it with the current date. If the end date is prior to the current date then a dialogue box appears showing that the coupon is already expired or if the current date is prior to the start date it displays a message saying that the coupon is not yet active. The coupon can be activated only when the current date is between the start date and the end date.

The screen shots of the android application are as shown below.

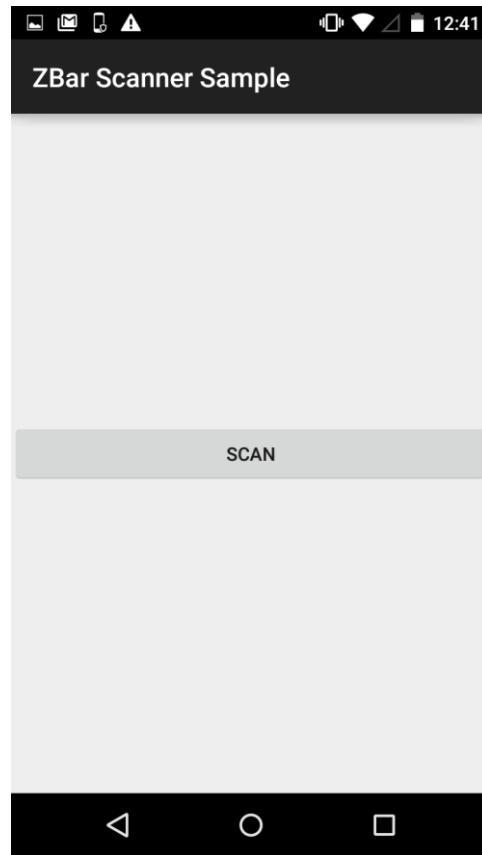


Figure 14: Android app start page

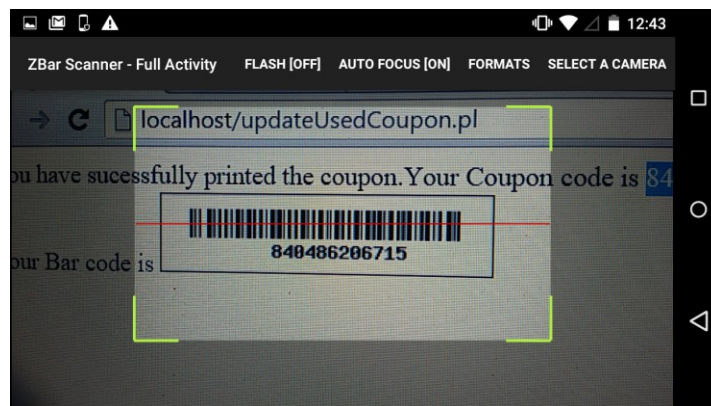


Figure 15: barcode scanner app

CHAPTER 4

CODE

4.1 Store owner

a) HTML:

i) Registration:

```
<div class="form-group">
  <label for="InputName">Enter Name</label>
  <div class="input-group">
    <input type="text" class="form-control"
name="InputName" id="InputName" placeholder="Enter Name" required>
    <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
  </div>
</div>

<div class="form-group">
  <label for="InputCompanyName">Enter Company Name</label>
  <div class="input-group">
    <input type="text" class="form-control"
name="InputCompanyName" id="InputCompanyName" placeholder="Enter Company
Name" required>
    <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
  </div>
</div>

<div class="form-group">
  <label for="InputAddress">Enter Address</label>
  <div class="input-group">
    <input type="text" class="form-control"
name="InputAddress" id="InputAddress" placeholder="Enter Address" required>
    <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
  </div>
</div>

<div class="form-group">
  <label for="InputEmail">Enter Email</label>
  <div class="input-group">
    <input type="email" class="form-control"
id="InputEmail" name="InputEmail" placeholder="Enter Email" required>
    <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
  </div>
</div>

<div class="form-group">
  <label for="Inputusername">Enter Username</label>
  <div class="input-group">
    <input type="text" class="form-control"
name="Inputusername" id="Inputusername" placeholder="Enter Username"
required>
    <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
  </div>
</div>

<div class="form-group">
  <label for="Password">Enter Password</label>
  <div class="input-group">
```

Figure 16: Store owner registration page

ii) Coupon generation:

```
<div class="well well-sm"><strong><span class="glyphicon glyphicon-
asterisk"></span>Required Field</strong></div>
  <div class="form-group">
    <label for="InputName">Coupon Name</label>
    <div class="input-group">
      <input type="text" class="form-control"
name="InputcouponName" id="InputcouponName" placeholder="Enter Coupon Name"
required>
      <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
    </div>
  </div>

  <div class="form-group">
    <label for="InputName">Coupon Start Date</label>
    <div class="input-group">
      <input type="date" class="form-control"
name="InputcouponSdate" id="InputcouponSdate" placeholder="Enter Coupon start
date" required>
      <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
    </div>
  </div>

  <div class="form-group">
    <label for="InputName">Coupon End Date</label>
    <div class="input-group">
      <input type="date" class="form-control"
name="InputcouponEdate" id="InputcouponEdate" placeholder="Enter Coupon End
date" required>
      <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
    </div>
  </div>
```

Figure 17: html code for coupon generation by store owner

b) PERL:

i) Storing registration data into database:

```
# Config DB variables
our $platform = "mysql";
our $database = "registration";
our $host = "localhost";
our $port = "3306";
our $tablename = "storeowners";
our $user = "test";
our $pw = "test";
our $q = new CGI;

# DATA SOURCE NAME
$dsn = "dbi:mysql:$database:localhost:3306";

# PERL DBI CONNECT
$connect = DBI->connect($dsn, $user, $pw);

#Get the parameter from your html form.
$name=$q->param('InputName');
$name=$q->param('InputCompanyName');
$address=$q->param('InputAddress');
$email=$q->param('InputEmail');
$username=$q->param('Inputusername');
$password=$q->param('Password');

print $q->header;

$sql="INSERT INTO
registration.storeowners(Name,companyName,Address,Email,InputName>Password)
values('$name','$cname','$address','$email','$username','$password)";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..

$rv = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";
#execute your query
```

Figure 18: PERL code for storing registration data into database

ii) Coupon generation:

```
# Config DB variables
our $platform = "mysql";
our $database = "registration";
our $host = "localhost";
our $port = "3306";
our $tablename = "coupons";
our $user = "test";
our $pw = "test";
our $q = new CGI;

# DATA SOURCE NAME
$dsn = "dbi:mysql:$database:localhost:3306";

# PERL DBI CONNECT
$connect = DBI->connect($dsn, $user, $pw);

#Get the parameter from your html form.
my $coupon=$q->param('InputcouponName');

my $startdate=$q->param('InputcouponSdate');

my $enddate=$q->param('InputcouponEdate');

print $q->header;

$sql="INSERT INTO registration.coupons(couponName,startdate,enddate)
values('$coupon','$startdate','$enddate)";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..

$rv = $sth->execute

or die "can 't execute the query $sql : $sth->errstrn";
#execute your query
```

Figure 19: Perl code for coupon generation by store owner

4.2 Customer

a) Html:

i) Customer registration:

```
<div class="well well-sm"><strong><span class="glyphicon glyphicon-
asterisk"></span>Required Field</strong></div>
  <div class="form-group">
    <label for="InputName">Customer Name</label>
    <div class="input-group">
      <input type="text" class="form-control"
name="InputCustName" id="InputName" placeholder="Enter Customer Name"
required>
        <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
      </div>
    </div>

    <div class="form-group">
      <label for="InputEmail">Enter Customer Email</label>
      <div class="input-group">
        <input type="email" class="form-control"
id="InputCustEmail" name="InputEmail" placeholder="Enter Customer Email"
required>
          <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
        </div>
      </div>

      <div class="form-group">
        <label for="InputPassword">Enter Password</label>
        <div class="input-group">
          <input type="password" class="form-control"
name="Password" id="CustNewPassword" placeholder="Password" required />
          <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
        </div>
      </div>

      <div class="form-group">
        <label for="InputPassword">Re-type Password</label>
        <div class="input-group">
          <input type="password" class="form-control"
name="Password" id="CustRePassword" placeholder="Re-type Password" required
/>
          <span class="input-group-addon"><span
class="glyphicon glyphicon-asterisk"></span></span>
        </div>
      </div>
    </div>
```

Figure 20: html code for customer registration

ii) Select coupon:

```
<div class="container">
  <div id="userNameDiv"></div>
  <input type="button" value="logout" id="logoutbtn"
onclick="logout()"></input>
  <div class="page-header">
    <h1>Generate Coupon </h1>
  </div>
  <form method="post" action="./updateUsedCoupon.pl">
    <div id="selectCoupon">Select a Coupon <select
id="selectACoupon" onchange="changedCoupon()"></select></div >
    <input type="hidden" id="secertCarrier"
name="couponCarrier"></input>
    <input type="hidden" id="secertuser" name="secretusername"
></input>
    <div id="printCoupobBtn"><input type="submit"
id="couponUpdateButton" value="Print a Coupon"> </input></div>
  </form>
</div>
<script src="./js/cookie.js"></script>
<script
src="//ajax.googleapis.com/ajax/libs/jquery/2.0.2/jquery.min.js">
</script>
<script src="js/bootstrap.min.js"></script>
```

Figure 21: html code for coupon generation by the customer

b) PERL:

i) Storing registration data in database using PERL:

```
$sql="INSERT INTO registration.users(username,email,password)
values('$username','$email','$password)";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..

$rv = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";

$sql="select barcode from registration.users where username='$username'";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..
$rv = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";

my @barcode;
while (@barcode = $sth->fetchrow_array) { # retrieve one row
}

$rv = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";

# get all the coupons details
$sql="SELECT couponName FROM registration.coupons";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..

$rv = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";

my @row;
while (@row = $sth->fetchrow_array) { # retrieve one row

foreach my $column (@row) {
push(@coupons, $column);
}
}
}
```

Figure 22: Perl code for storing customer data into database

ii) Coupon search and used coupon updating:

```
# DATA SOURCE NAME
$dsn = "dbi:mysql:$database:localhost:3306";

# PERL DBI CONNECT
$connect = DBI->connect($dsn, $user, $pw);

#Get the parameter from your html form.

$usedCoupon=$q->param('couponCarrier');
$username = $q->param('secretusername');

print $q->header;

$sql="SELECT usedCoupons FROM registration.users WHERE username =
'$username' ";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..

$rval = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";

my @row;
while (@row = $sth->fetchrow_array) { # retrieve one row

push(@usedCoupons, @row);
}

push(@usedCoupons, $usedCoupon);

#updating barcode to users and coupon tables

$sql="SELECT usedBarCodes FROM registration.users ";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..

$rval = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";

my @row;
while (@row = $sth->fetchrow_array) { # retrieve one row

push(@usedBarCodes, @row);
}

my $flag= true;

my $minimum = 100000000000;
my $maximum = 999999999999;

my $newBarCode ;
```

```

$newBarCode= $minimum + int(rand($maximum - $minimum));
my @matches = grep { /$newBarCode/ } @usedBarCodes;
while(@matches){

if( @matches!= "" && @matches!= null ){

$newBarCode= $minimum + int(rand($maximum - $minimum));
@matches = grep { /$newBarCode/ } @usedBarCodes;
}
}

# update user table with the generated barcodes

$sql="SELECT usedBarCodes FROM registration.users WHERE username =
'$username' " ;
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..

$rval = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";

my @row;
while (@row = $sth->fetchrow_array) { # retrieve one row

push(@usedBarCodesOfaUser, @row);
}

push(@usedBarCodesOfaUser, $newBarCode);

}

}

$sql="Select allBarCodes from registration.coupons WHERE couponName =
'$usedCoupon'";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";
#pass sql query to database handle..

$rvalue = $sth->execute
or die "can 't execute the query $sql : $sth->errstrn";
#execute your query

my @row;
while (@row = $sth->fetchrow_array) { # retrieve one row

push(@allBarCodesOfACoupon, @row);
}

push(@allBarCodesOfACoupon, $newBarCode);

}

}

$sql="update registration.users set usedCoupons = '@usedCoupons' ,
usedBarCodes = '@usedBarCodesOfaUser' WHERE username = '$username'";
$sth = $connect->prepare($sql)
or die "Can't prepare $sql: $connect->errstrn";

```

Figure 23: Perl code for generation of unique barcode

4.3 Employee

a) Android application code for scanning and connecting to database:

```
public void handleResult(Result rawResult) {

    try
    {
        Uri notification =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        Ringtone r = RingtoneManager.getRingtone(getApplicationContext(),
notification);
        r.play();
        showMessageDialog("Contents = " + rawResult.getContents() + ",
Format = " + rawResult.getBarcodeFormat().getName());
        String result = "";
        InputStream isr = null;
        try{
            HttpClient httpclient = new DefaultHttpClient();
            HttpPost httppost = new
HttpPost("http://localhost/getAllCustomers.php"); //YOUR PHP SCRIPT ADDRESS
            HttpResponse response = httpclient.execute(httppost);
            HttpEntity entity = response.getEntity();
            isr = entity.getContent();
        }
        catch(Exception e){
            Log.e("log_tag", "Error in http connection " + e.toString());
            // resultView.setText("Couldnt connect to database");
        }
        //convert response to string
        try{
            BufferedReader reader = new BufferedReader(new
InputStreamReader(isr,"iso-8859-1"),8);
            StringBuilder sb = new StringBuilder();
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            isr.close();

            result=sb.toString();
        }
        catch(Exception e){
            Log.e("log_tag", "Error converting result "+e.toString());
        }
    }
}
```

```

//parse json data
String Message = rawResult.getContents();
try {
    String s = "";
    JSONArray jArray = new JSONArray(result);
    Date curDate = new Date();
    SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy");
    String DateToStr = format.format(curDate);

    for(int i=0; i<jArray.length();i++){
        JSONObject json = jArray.getJSONObject(i);
        /*s = s +
            "coupon name : "+json.getString("couponName")+
            "Start date : "+json.getString("startdate")+"\n"+
            "End date : "+json.getString("enddate")+"\n"+
            "Barcode: "+json.getString("barcode")+"\n"+
            "All Barcodes:
"+json.getString("allBarCodes")+"\n\n";*/
        String barcode = json.getString("allBarCodes");
        String startdate = json.getString("startdate");
        String enddate = json.getString("enddate");

        if(barcode.equals(Message))
        {
            Date date1 = new
SimpleDateFormat("MM/dd/yyyy").parse(DateToStr);
            Date date2 = new
SimpleDateFormat("MM/dd/yyyy").parse(startdate);
            Date date3 = new
SimpleDateFormat("MM/dd/yyyy").parse(enddate);
            if((date1.after(date2)) && date1.before(date3))
            {
                showMessageDialog("Still active");
            }
            else if(date1.before(date2))
            {
                showMessageDialog("Not active Yet");
            }
            else if(date1.after(date3))
            {
                showMessageDialog("Expired");
            }
        }
    }

    //resultView.setText(s);
} catch (Exception e) {
    // TODO: handle exception
    Log.e("log_tag", "Error Parsing Data "+e.toString());
}

} catch (Exception e) {}

```

Figure 24: Android application code for connecting to database

b) Php code for android mobile accessing code from database:

```
<?php

    if(isset($_REQUEST['user']))
    {
        $con = mysql_connect("localhost","root","password");
        if (!$con)
        {
            die('Could not connect: ' . mysql_error());
        }
        mysql_select_db("TestDatabase", $con);

        $name = $_REQUEST['user'];

        $result = mysql_query("SELECT * FROM Customer WHERE FirstName =
'$name' ") or die('Errant query:');

        while($row = mysql_fetch_assoc($result))
        {
            $output[]=$row;
        }

        print(json_encode($output));

        mysql_close($con);
    }
    else
    {
        $output = "not found";
        print(json_encode($output));
    }

?>
```

Figure 25: code for connecting the server with the android application

CHAPTER 5

CONCLUSION

The present invention comprises a coupon redemption system and method for redeeming both paper and electronic coupons that eliminates fraud. The system and method provides verification, validation and authentication of coupon redemption transactions in a third party environment while eliminating the need to ship coupons to manufacturers for reimbursement. Moreover, the system and method of the present invention also advantageously addresses the increasingly complex identification of coupons by removing the encoded redemption requirements from the coupon by providing the ability to validate coupons based on the coupon issuer's registered coupon redemption requirement data.

The present invention comprises a universal coupon redemption system and method for redeeming both paper and electronic coupons that incorporates the ability to audit each coupon redemption record processed for payment. The system and method of the present invention addresses the two major origins of fraud in prior art coupon redemption systems: it eliminates the opportunity for consumers to create and redeem counterfeit coupons, and it eliminates the need to ship coupons to clearinghouses for reimbursement.

CHAPTER 6

FUTURE WORK

Coupons continue to be one of the most favorite promotional tools for the marketers to influence, incentivize, and encourages customers towards trying the product. Mobile-coupon (mcoupon), an extension of the traditional coupons which is accessible on mobile device is a highly personalized medium, it enhances the opportunity for the marketers to reach out to the potential customers in an effective manner.

REFERENCES

- [1] Oliver, Richard I., and Mikhael Shor. (2003). “Digital Redemption of Coupons: Satisfying and Dissatisfying Effects of Promotion Codes.” *Journal of Product & Brand Management*, 12(2): 121-134.
- [2] Araújo, Jorge, Gonçalo Paiva Dias, Hélder Gomes, Jorge Gonçalves, Daniel Magueta, Fábio Marques, Ciro Martins, and Mário Rodrigues. (2015). “An Integrated Service Solution for Digital Discount Coupons Processing.” *New Contributions in Information Systems and Technologies Advances in Intelligent Systems and Computing*, 395–401. Springer International Publishing, Switzerland.
- [3] Banerjee, Sy, and Scott Yancey. (2010). “Enhancing Mobile Coupon Redemption in Fast Food Campaigns.” *Journal of Research in Interactive Marketing*, 4(2): 97–110.
- [4] Holliman, Geraint, and Jennifer Rowley. (2014). “Business To Business Digital Content Marketing: Marketers’ Perceptions of Best Practice.” *Journal of Research in Interactive Marketing*, 8(4): 269–93.
- [5] Babakus, Emin, Peter Tat, and William Cunningham. (1988). “Coupon Redemption: A Motivational Perspective.” *Journal of Consumer Marketing*, 5(2): 37–43.
- [6] Mobile Marketing Association. (2011, March). “Mobile Couponing Guidelines (v1.2).” (<https://mmaglobal.com/files/mobilecouponinguidelines.pdf>)
- [7] First Data. (2012). “Solving Fraudulent Redemption: A Simple Solution for Coupons, Daily Deals and Other Promotions.” (<https://www.firstdata.com/downloads/thought-leadership/3526-Consumer-Authentic-Market-Insight.pdf>).
- [8] Salesforce. (2015, June). “Live Offers.” (https://help.exacttarget.com/en-US/documentation/exacttarget/content/live_offers/).

[9]WooThemes.(n.d.). “Coupon Management.”

(<https://docs.woothemes.com/document/coupon-management>).

[10] Recurly. (2015). “Coupons and Discounts.” (<https://docs.recurly.com/coupons>).

[11] qr2coupon. (n.d.). “How to Redeem Coupons.”

(http://www.qr2coupon.net/en/how_to/mobile-coupon-redemption)

VITA

Ram Shasank Pathath was born on July 06, 1992, in Andhra Pradesh, India. He completed his schooling in Andhra Pradesh and graduated from high school in 2007. He then completed his Bachelor's degree in Computer Science Engineering from GANDHI Institute of Technology and Management.

In fall 2014, he came to US for his masters and took several courses towards Software Engineering.