

# **Control Strategy and Payment System for Department of Textile**

**Tanvir Mamunul Hasan**  
**Student Id: 012152017**

A Project  
in  
The Department  
of  
Computer Science and Engineering



Presented in Partial Fulfillment of the Requirements  
For the Degree of Master of Science in Computer Science and Engineering  
United International University  
Dhaka, Bangladesh  
August, 2018

© Tanvir Mamunul Hasan, 2018

# Approval Certificate

This project titled “**Control Strategy and Payment System for Department of Textile**” submitted by Tanvir Mamunul Hasan, Student ID: **012152017**, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on 11-08-2018.

## Board of Examiners

1.

---

Dr. Mohammad Nurul Huda  
Professor & MSCSE Coordinator  
United International University, Dhaka

Supervisor

2.

---

Suman Ahmmed  
Assistant Professor  
United International University, Dhaka

Examiner

3.

---

Dr. Mohammad Nurul Huda  
Professor & MSCSE Coordinator  
United International University, Dhaka

Ex-Officio

## Declaration

This is to certify that the work entitled “**Control Strategy and Payment System for Department of Textile**” is the project carried out by me under the supervision of Dr. Mohammad Nurul Huda, Professor & MSCSE Coordinator

---

Tanvir Mamunul Hasan  
Student ID:012152017  
MSCSE Program

In my capacity as supervisor of the candidate’s project, I certify that the above statements are true to the best of my knowledge.

---

Dr. Mohammad Nurul Huda  
Professor & MSCSE Coordinator

## **Abstract**

To develop online one click registration & other services delivery by Department of Textiles by developing an e-center network using ICT for Department of Textiles to diminish their service seekers harassment and difficulties to get desired services. There are two types of service seekers like investors & employees of this department. In this project, all the manual services by Department of Textiles are planned to transform into e-Services. A web portal will be designed and developed both in English and Bengali language as a delivery platform of the online services. Investors will be able to apply online for new textile factories as well as make payment through online. This project will simplify and faster the work process and save time, visit & cost of the service seekers.

## **Acknowledgement**

I would like to express my special thanks of gratitude to my teacher Dr. Mohammad Nurul Huda, who gave me the golden opportunity to do this wonderful project of “Control Strategy and Payment System for Department of Textile”, who also helped me in completing my project. I came to know about so many new things. I am really thankful to them. Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

# Table of Contents

LIST OF TABLES .....	vi
LIST OF FIGURES.....	vii
1. Document Overview .....	1
1.1 Purpose.....	1
1.2 Objectives .....	1
1.3 Scope.....	1
1.4 Reference Documents .....	1
2. Design Considerations .....	2
2.1 Assumptions and Dependencies .....	2
2.2 System Environment .....	2
2.3 Design Methodology .....	3
2.4 Risks and Volatile Areas .....	3
2.5 Users .....	3
2.6 Performance Requirements .....	4
2.7 Workload: .....	5
2.8 Environmental Constraints.....	5
3. System Overview .....	6
3.1 Design Approach .....	6
3.2 Architecture.....	7
3.3 Functional Flow .....	8
3.3.3 Payment Module.....	9
3.3.6 Workflow Module .....	11
4. System Interface .....	13
4.1 Data Transfer.....	13
4.2 API exposed .....	13
4.2.1 SMS Module.....	13
4.2.2 Payment Gateway Module.....	13
4.3 Integration of Payment Gateway.....	14
5. Sub Component Details .....	22
5.1 Screen Item Description.....	22
5.2 Error and Exception Handling .....	25
5.3 Report Layout and Description .....	28
5.3.1 Report: Payment Module .....	28
5.3.2 Report Name: Master Workflow List .....	28
5.3.3 Report: Workflow Module .....	28
6. Integration Environment and Sequence.....	29
6.1 Integration Environment .....	29
6.2 Integration Sequence:.....	30
6.3 Entry-Exit Criteria .....	32
7. Conclusion .....	33
Appendix .....	34

## **LIST OF TABLES**

Table 1: Proposed Specification of Operating Server .....	2
Table 2: Proposed Specification of Client Workstations.....	2
Table 3: Customer interactions .....	5
Table 4: Proposed Specification of Operating Server(IE).....	29
Table 5: Proposed Specification of Client Workstations(IE) .....	29

## LIST OF FIGURES

Figure 1: On-line Application Management System .....	4
Figure 2: Three Tier Web Architecture .....	7
Figure 3: Online bank card payment .....	10
Figure 4: Workflow Module .....	12
Figure 6: Payment Gateway .....	30
Figure 7: SMS Gateway.....	31



# Chapter 1

## Document Overview

This document contains the basic design information regarding both the backend and frontend. In the backend side: required number of tables, tables and its attributes information (e.g. rows and columns of the tables), data types and other constraints required to meet the user requirements according to the SRS document. In the frontend side: design of the templates of user and staff panel, form design, report design and other related design view are mentioned in this document. This document also intended to help detect contradictions prior to coding, and used as a reference manual for how the modules interact at a high level.

### 1.1 Purpose

Administrative control will be managed by developing an e-center network using ICT for Department of Textiles to diminish their service seekers harassment and difficulties to get desired services. Investors will be able to apply online for new textile factories as well as make payment through online.

### 1.2 Objectives

- a. All the manual services by DOT are planned to transform into e-Services;
- b. Administrative Control over services will be customized and controllable
- c. Application along with the payment will be received through online;

### 1.3 Scope

The system has to be design and maintain to meet the demand of the DOT officials for customizing the workflow including Online Payment Mechanism.

### 1.4 Reference Documents

RFP for DoT\_SIF R2 Project.pdf

## Chapter 2

### Design Considerations

#### 2.1 Assumptions and Dependencies

- As per our previous conversation with DOT, decision has made that client side will provide SMS API for notifications. Also for e-mail notification, it depends on client to provide admin email id for email notification.
- For Online payment vendor will provide payment gateway API. Payment gateway module will start after getting proper authorization from DOT authority.
- SSL certificate will provide by the client for internet security.

#### 2.2 System Environment

:

Table 1: Proposed Specification of Operating Server

<b>Processor</b>	Intel Xeon Quad Core Processor
<b>Memory</b>	32 GB ECC
<b>Storage</b>	3 SAS Disks
<b>Mirroring</b>	Raid Level 3
<b>Network</b>	Gigabit Redundant NIC
<b>Backup</b>	USB Based Tape Device (80GB)
<b>Power</b>	Redundant Power Supply (online)

Table 2: Proposed Specification of Client Workstations

<b>Processor</b>	Intel Dual Core
<b>Memory</b>	1 GB
<b>Storage</b>	250GB HDD
<b>Network</b>	10/100 Mbps Ethernet + Wireless LAN + CDMA/Edge/GPRS/3G Modem

## 2.3 Design Methodology

The Unified Modeling Language (UML) is a graphical notation for software system modeling that is widely used in the software industry. We are using UML to describe design methodology of ORMS system.

Here we shortly describe two types of diagrams: class diagrams and UseCase diagrams. Class diagrams used to describe the static structure (data model) of a software system. ORMS software model can more formally be described by class diagram. UseCase diagrams are used to describe task interactions. They can be used to more formally describe scenarios (workflow).

Please note that the UML is a notation, not a method. The UML does not define when, why, and how to use which diagrams. It only describes the meaning of legally composed diagram elements.

## 2.4 Risks and Volatile Areas

Internet connection is must for all users as it is an online system.

## 2.5 Users

The user classes of this database system will be –

- i. **System Admin:** System Admin will be responsible for maintenance, update and other required services of the DOT system.
- ii. **DOT Admin:** DOT admin will have the full access to the system and responsible for maintaining the workflow and DOT staff activities.
- iii. **DOT Staff:** DOT staffs are the officials of the DOT who will be responsible for monitoring the online applications and related documents

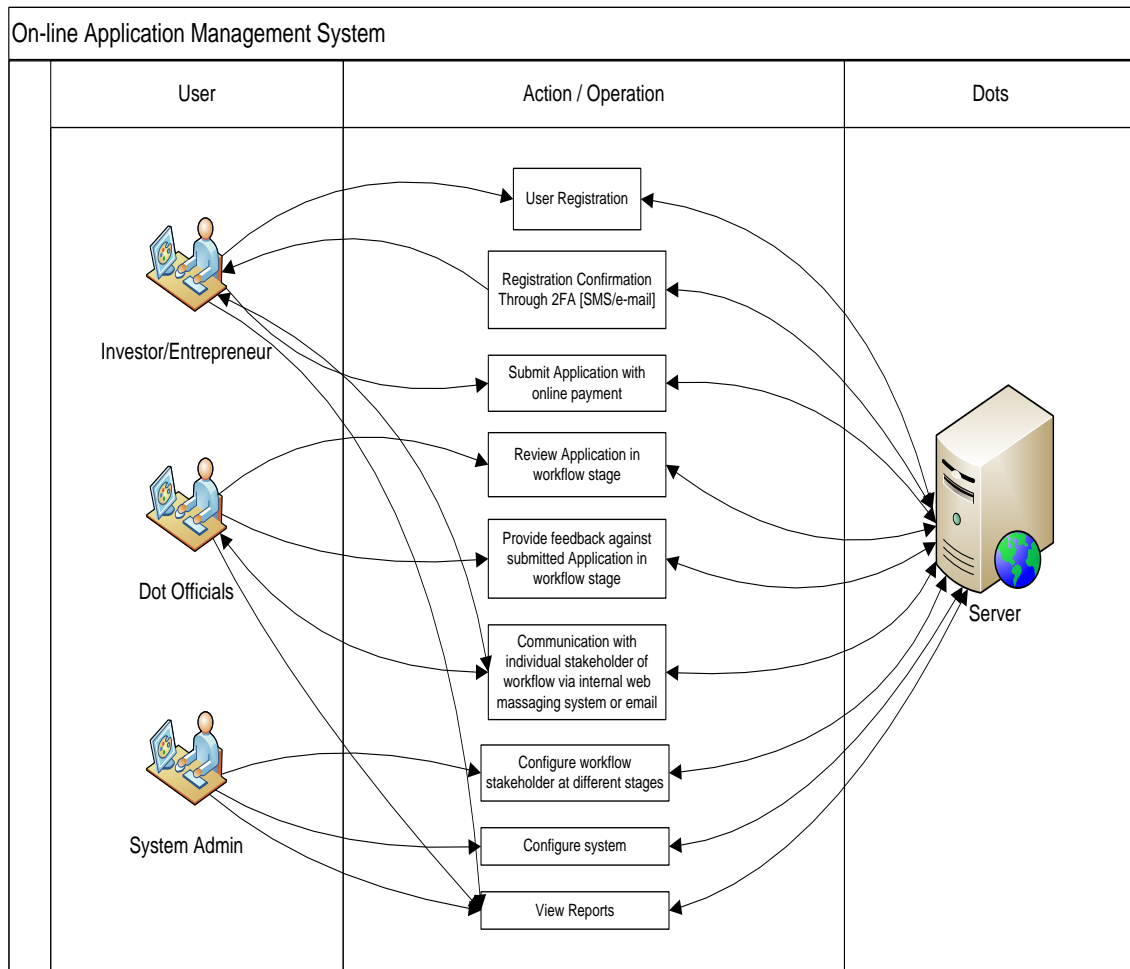


Figure 1: On-line Application Management System

## 2.6 Performance Requirements

In supporting 4,000 customers, it shall ensure that performance shall not fall below the following level:

- 95% of ALL visible pages for “normal” customers respond in 8 seconds or less, including infrastructure, excluding back ends and payment gateway.
- The response times will be measured using Apache web servers default monitoring feature. For the purpose of measuring the response time the performance tests should not exceed 80% CPU utilization during the busy hour.

## 2.7 Workload:

The software must support 10,00,000 customers which will on a busy day generate 4000 customer interactions as outlined in the table below:

Table 3: Customer interactions

Ref No	Description	Pages	Daily Total
1	Portal	Login, Portal Exit	1400
2	Portal Registration (Only Client)	Sign-Up, User registration, Dashboard, Exit	900
3	Dashboard	Login, Portal, Dashboard, D Stat, Exit	500
4	Profile edit	Login, Portal, Profile View, Portal Edit, Exit	250
5	Change Password	Login, Portal, Profile View, Password change	200
6	New Applications (Only Client)	Login, Portal, Dashboard, attached vehicle, Exit	200
8	Payment (Only Client)	Login, Portal, Dashboard, New Application, Finalize Application, Upload Attachments, Confirm Payee, 50%, Exit	100
9	Print Receipt (Only Client)	Login, Portal, Dashboard, New Application, Finalize Application, Upload Attachments, Confirm Payee, 50%, Exit	100
<b>Total :</b>			<b>4000</b>

## 2.8 Environmental Constraints

- Hardware/software environment: This application can run in all web browsers
- Memory and other capacity limitations: Web Session information is creating for each user needs to be clear monthly basis, as it consumes more storage capacity.
- Network communications: This application is fully internet dependent.
- Interaction with other software components: Payment feature of this application is dependent on payment gateway.

If any of the above-mentioned system is unavailable, then feature related to those systems will stop functioning.

## Chapter 3

### System Overview

#### 3.1 Design Approach

The Software will be built entirely from the scratch using web based technology of PHP framework (Zend Framework) and Oracle Database (11g R2). The database Oracle has distinct features providing sleek and small footprint for such large databases. PHP, on the other hand provides handy built-ins and ready to build functions for accessing, manipulating and updating of the database.

This Software will store all the parameters embedded in the program (database). In this regard, the master fields will be stored in logical tables and views and queries will be created to access it and calculations will be made each time a query runs through forms or reports. This will eliminate the time-consuming backward tracking and calculation hazards. Another feature of the software will be that it will store options fields. It means that the global parameters will be kept in case of change in the business rules, the system can automatically be set to rule and it can be administered by the DBA without loss of time.

The proposed system can be install in the client-server architecture (LAN) as well as Wide Area Network (WAN) to ensure implementation in standalone or below par IT infrastructure. This means, the views and queries created and run in the workstations will fetch the data into its memory and can be access into various cursors (temporary data storage within memory) and the workstation can act as server itself.

The software can run from any workstation or server without requiring any single software to be installed in the client side. In this regard, the server edition of the software will use the maintenance and tuning options. If all the workstations fail, it can still be accessed from the server and the front-end system will run from server.

The software will use the user and group level security and create a repository of the access levels into the database for additional security measures. The access to the system will be determined through the password-based authentication, which inherited from the Operating System (Linux Kerberos / Radius or a Windows OS Native) or can

implement the database user access policies defined by grant on schema level objects like Tables, Views, and Data Types etc.

### 3.2 Architecture

The proposed system will be in 3-tier web architecture. we intend to develop the system web architecture extensible and capable of running on n-tier architecture with multitasking, multi-user features providing maximum facilities for the users of all level. The system will use OLTP for transaction processing and will have higher end connectivity with other server(s) or resources shared over the network/Internet. This will create a repository for the present peripheral users.

In web application development, three-tier architecture refers to separating the application process into three specific layers. What the user sees via a web browser is called the presentation tier and is content served from a web server. The middle tier performs the business logic processing that occurs, for example, when a user submits a form. The back end consists of the data tier which handles the database processing and access to the data. We'll take a simplistic look at each of these.

This application should be mobile responsive, RTI compliant and interoperable with Bangladesh national portal framework. SEO are also need to be consider for this application architecture.

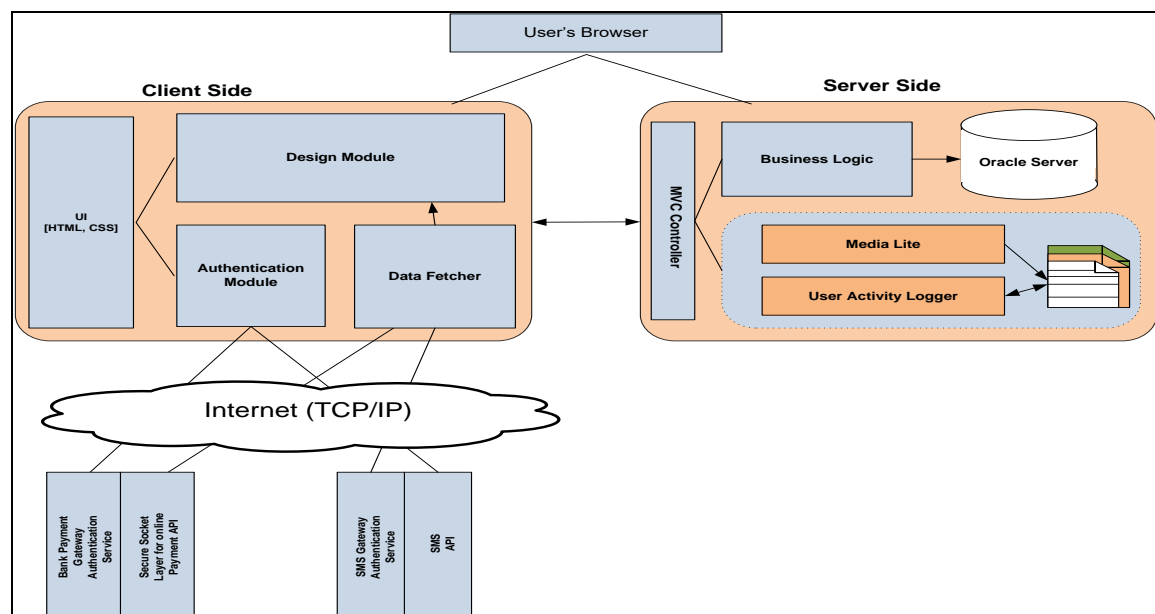


Figure 2: Three Tier Web Architecture

### 3.3 Functional Flow

In this proposed solution, we have defined this system in five modules. These are:

1. Authentication Module
2. Application Module

List of services required to be included in the system.

- I. Registration
    - a. New Registration
    - b. Existing Registration
    - c. Record open [registration from board of investment]
  - II. Regular I.R.C
  - III. 1st ad-hoc I.R.C
  - IV. 2nd ad-hoc I.R.C
  - V. Machine release letter
  - VI. Ownership transfer
  - VII. Permanent change of address
  - VIII. Temporary change of address
  - IX. Work permit for foreign national
  - X. Deferred payment recommendation letter
  - XI. Obstacles of indemnity bond recommendation letter
  - XII. Composite certificate
  - XIII. 3rd ad-hoc IRC
  - XIV. IP imposition recommendation letter
  - XV. Up recommendation letter
  - XVI. Investment related correction
  - XVII. PI visa Recommendation Letter
  - XVIII. Foreign debt recommendation letter
3. Payment Module
  4. Notification Module
  5. Administrative Module
  6. Workflow Module
  7. Report Module

**Each of these modules is integrate internally with each other. Our concern is to update workflow module for better administrative control and implement online payment mechanism in Payment Module. These revised modules are describing in below section.**



### **3.3.3 Payment Module**

#### **Online bank card payment**

##### **➤ Objects**

- Investor / Entrepreneur
- Dots officials
- Dots system

##### **➤ Cases**

- Select Online Bank payment option in payment stage.
- Get auto calculate payment amount
- Input Bank card information/ payment related data.
- Submit payment information
- System Check the payment validations.
- Check payment status from application view page
- Check payment status from application view page by dots official

➤ **Use Case Diagram**

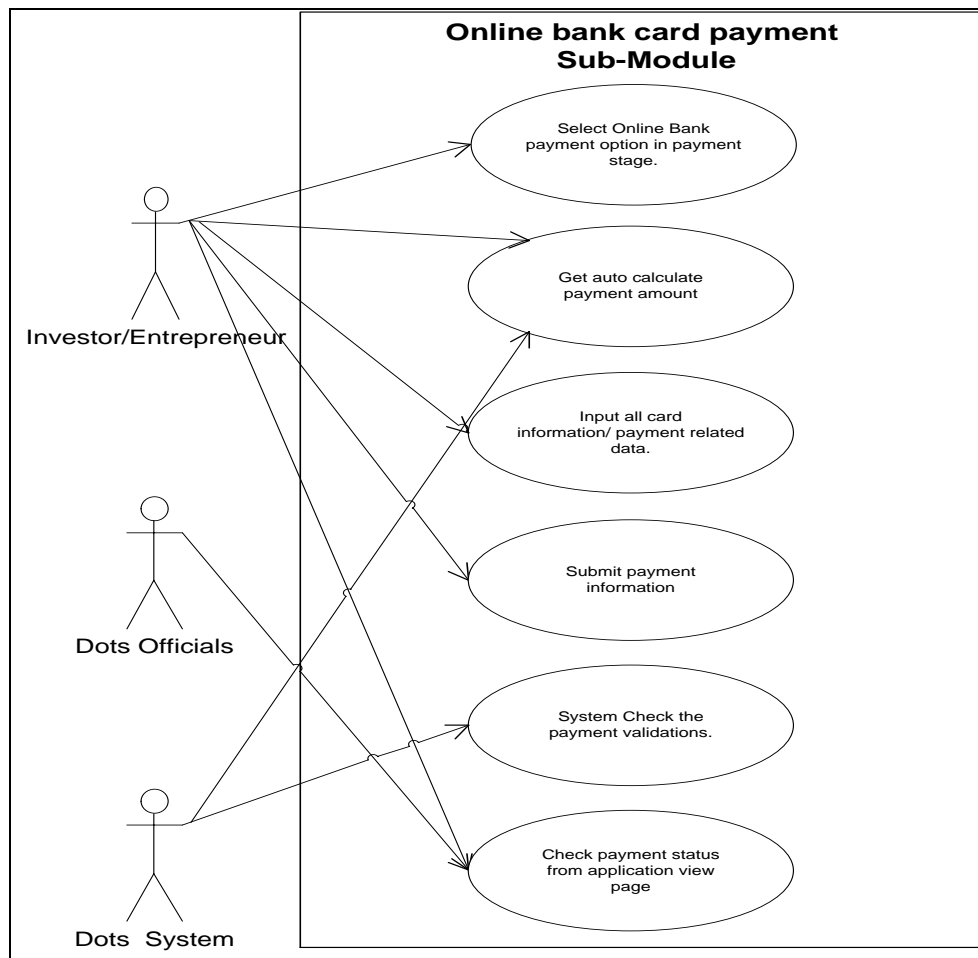


Figure 3: Online bank card payment

➤ **Description**

1. Users select Online Bank payment option in payment stage. Finalizes application will appear after payment is done.
2. System generate auto calculated payment amount, it is not changeable.
3. Insert bankcard information in payment gateway.
4. Submit payment information and then finalize application.
5. System Check the payment validations.
6. User can check payment status from application view page
7. Dots official can check payment status from application view page

### 3.3.6 Workflow Module

#### ➤ Objects

- Dots Officials
- Dots system

#### ➤ Cases

- One-Stop get a new application from Investor/Entrepreneur
- One-Stop check & forward the application with Inspection status to AD/Director.
- Director checks the application, assign multiple inspector for inspection, and forward it to the AD.
- AD check application and input an inspection date and forward to next.
- Inspectors are complete the inspection and upload inspection report to system and submit to AD.
- AD check inspection report with application and forward to DD.
- DD check application and forward it to director for final approval.
- Director approves/ rejects the application.
- In the final stage AD, prepare output letter for the Investor/Entrepreneur.

➤ Use Case Diagram

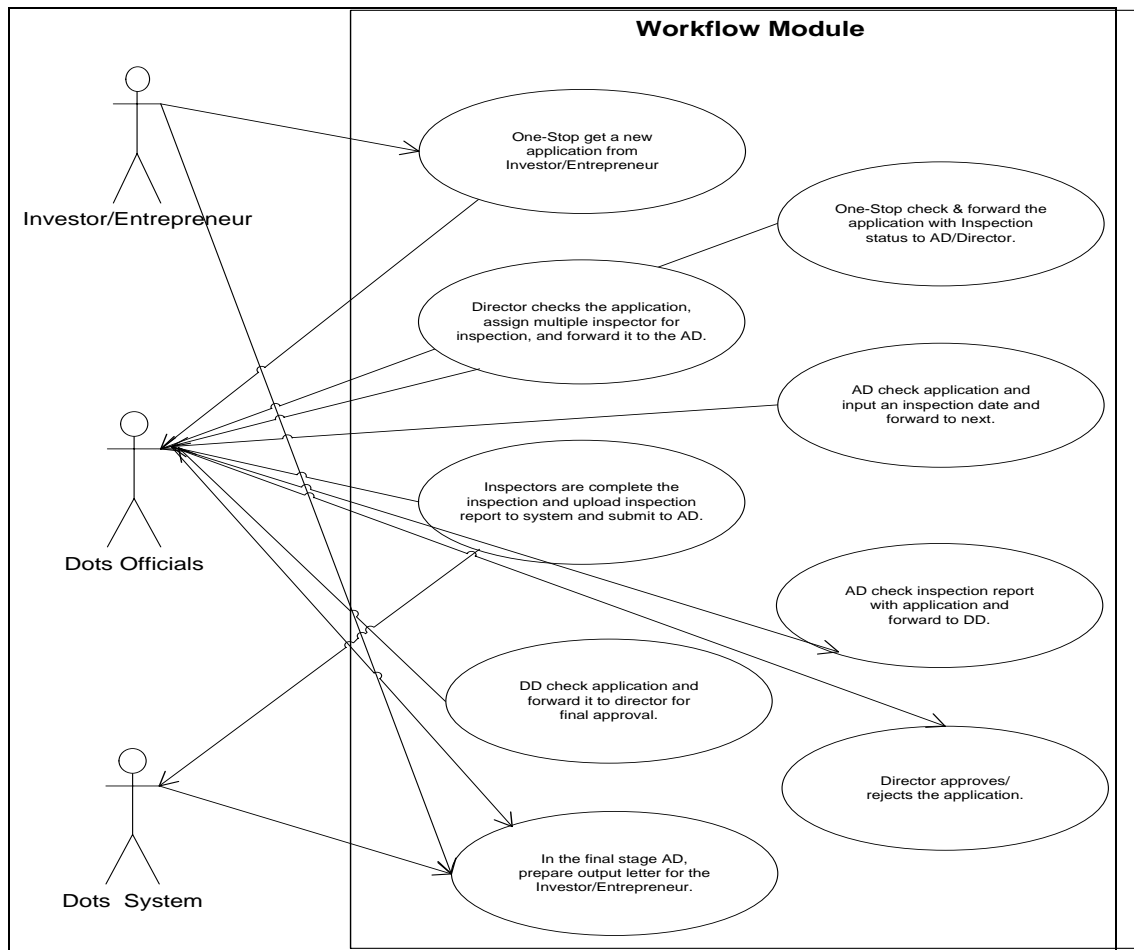


Figure 4:Workflow Module

➤ Description

1. One-Stop get a new application from Investor/Entrepreneur
2. One-Stop check & forward the application with Inspection status to AD/Director.
3. Director checks the application, assign multiple inspectors for inspection, and forward it to the AD.
4. AD check application and input an inspection date and forward to next.
5. Inspectors are complete the inspection and upload inspection report to system and submit to AD.
6. AD check inspection report with application and forward it to DD.
7. DD check application and forward it to director for final approval.
8. Director approves/ rejects the application.
9. In the final stage AD, prepare output letter for the Investor/Entrepreneur.

# Chapter 4

## System Interface

### 4.1 Data Transfer

Data transfer between any action to action or controller to action via URL is mandatory to use encryption/decryption method. For encryption use `base64Encode(string)` method and for decryption use `base64Decode(string)` method.

### 4.2 API exposed

#### 4.2.1 SMS Module

1. The web panel IP: <http://123.49.3.58:8081/>
2. For automated system you can use [http://123.49.3.58:8081/web\\_send\\_sms.php?ms=8801XXXXXXXXXX&txt=smscontent](http://123.49.3.58:8081/web_send_sms.php?ms=8801XXXXXXXXXX&txt=smscontent) & username=User ID&password=Password

The above mentioning link contains parameter.

Note: Parameter "ms= 88Mobile no(13 digits)." and "txt= Your 160 character message"

#### 4.2.2 Payment Gateway Module

1. BRAC Bank Gateway => Payment Success => Save in Banking System as valid Transaction => Return to Report (PDF Generated Receipts) opened in Acrobat Reader
2. Payment Failure => Returns to **Payment Form** with all data Populated.

### 4.3 Integration of Payment Gateway

Submit an HTML form with required information to encrypt.php. Following fields should be submitted using POST method. **All the fields are required except Optional fields specified.**

Field Name	Intended Value
action	Transaction type should be set as the value of the field. E.g. : SaleTxn - for sale transactions SaleMerchUpdated - for sale merchant updated SaleTxnVerify - for sale transaction verify
Cur	Currency code E.g. : LKR
mer_id	Merchant ID
mer_txn_id	Merchant Transaction reference ID
txn_amt	Transaction amount
ret_url	Return URL which decrypt.php is located. Only the domain name or the IP address should be specified as the value. E.g. : http://www.example.com
mer_var1	Merchant variable 1 (Optional)
mer_var2	Merchant variable 2 (Optional)
mer_var3	Merchant variable 3 (Optional)
mer_var4	Merchant variable 4 (Optional)
ipg_server_url	IPG server URL. Only the domain name or the IP address should be specified as the value. E.g. : https://www.IPGServer.com

## Plain Text Invoice (Request Message)

Plain text invoice should STRICTLY follow the below XML format. All the xml tags should be in lower case. This plain text invoice should not contain embedded new line characters. Each xml tag is displayed on a new line below for clarity.

The value of 'action' tag will define how your request is processed in IPG server.

E.g.

If value of 'action' tag is 'SaleTxn' then IPG server will produce html pages to capture card details and make the payment.

If it is 'SaleTxnVerify' then IPG server will assume this is a status request of an old transaction.

Hence depending on the value of 'action' tag other parameters on the request message may be mandatory or optional.

Please see section 9.0 Possible Values of 'action' tag for more details.

```
<req>
<action>Action</action>
<card_number>CardNumber</card_number>
<cur>CurrencyCode</cur>
<cvc>CVCNumber</cvc>
<exp_month>ExpiryMonth</exp_month>
<exp_year>ExpiryYear</exp_year>
<holder_name>CardHolderName</holder_name>
<ipg_txn_id>IPGTransactionID</ipg_txn_id>
<lang>LanguageCode</lang>
<mer_id>MerchantID</mer_id>
<mer_txn_id>MerRefID</mer_txn_id>
<mer_var1>MerVar1</mer_var1>
<mer_var2>MerVar2</mer_var2>
<mer_var3>MerVar3</mer_var3>
<mer_var4>MerVar4</mer_var4>
<prod_type>ProductType</prod_type>
<ret_url>ReturnURL</ret_url>
<txn_amt>TxnAmount</txn_amt>
```

Value of Tag	Description
Action	<ul style="list-style-type: none"> <li>• Unique constant defined to identify type of transaction request / response.</li> <li>• Depending on this value IPG server will process the request.</li> <li>• Action is case sensitive.</li> <li>• <b>This is a mandatory parameter for all request messages.</b></li> </ul>
CardNumber	<ul style="list-style-type: none"> <li>• Credit card number.</li> <li>• CardNumber should not contain embedded white space(s).</li> <li>• Also the CardNumber cannot have any alphabetic characters or special characters such as commas or hyphens.</li> </ul>
CurrencyCode	<ul style="list-style-type: none"> <li>• Alphabetic Currency Code from which the transaction is performed.</li> <li>• The length of the CurrencyCode should be exactly 3 alphabetic characters.</li> <li>• CurrencyCode is case sensitive.</li> <li>• Supported Currency Codes will be provided by bank for each merchant at their sign up with IPG.</li> </ul>
CVCNumber	<ul style="list-style-type: none"> <li>• CVC number of the credit card.</li> <li>• CVCNumber should not contain embedded white space(s).</li> <li>• Also the CVCNumber cannot have any alphabetic characters or special characters such as commas or hyphens.</li> </ul>
ExpieryMonth	<ul style="list-style-type: none"> <li>• Expiry month of credit card in 2 digits.</li> <li>• ExpiryMonth should not contain embedded white space(s).</li> <li>• Also the ExpiryMonth cannot have any alphabetic characters or special characters such as commas or hyphens.</li> </ul>
ExpieryYear	<ul style="list-style-type: none"> <li>• Expiry year of the credit card in 4 digits.</li> <li>• ExpiryYear should not contain embedded white space(s).</li> <li>• Also the ExpiryYear cannot have any alphabetic characters or special characters such as commas or hyphens.</li> </ul>
CardHolderName	<ul style="list-style-type: none"> <li>• Card holder's name as appeared on the credit card.</li> <li>• The length of CardHolderName should not exceed 50 characters.</li> </ul>



	<ul style="list-style-type: none"> <li>• CardHolderName must contain alphabetic characters and spaces only.</li> </ul>
IPGTransactionID	<ul style="list-style-type: none"> <li>• Unique ID generated by the IPG Server for each transaction.</li> <li>• Should be sent with the subsequent queries to IPG Server.</li> <li>• IPGTransactionID is case sensitive.</li> </ul>
LanguageCode	<ul style="list-style-type: none"> <li>• Alphabetic Language Code from which the IPG Server payment pages is displayed.</li> <li>• The length of the LanguageCode should be exactly 2 alphabetic characters.</li> <li>• LanguageCode is case sensitive.</li> <li>• Supported Language Codes will be provided by the bank for each merchant at their sign up with IPG.</li> <li>• Default value is 'en' for English.</li> </ul>
MerchantID	<ul style="list-style-type: none"> <li>• This is a unique ID given by the bank for each merchant at their sign up with IPG.</li> <li>• The length of the MerchantID should not exceed 20 alphanumeric characters.</li> <li>• MerchantID is <b>NOT</b> case sensitive.</li> <li>• MerchantID should not contain embedded white space(s).</li> <li>• <b>This is a mandatory parameter for all request messages.</b></li> </ul>
MerRefID	<ul style="list-style-type: none"> <li>• This is a unique ID generated by the merchant for each transaction.</li> <li>• The length of the MerRefID should not exceed 20 alphanumeric characters.</li> <li>• MerRefID is case sensitive.</li> <li>• MerRefID should not contain embedded white space(s).</li> </ul>
MerVarX	<ul style="list-style-type: none"> <li>• This is a remark generated by the merchant for each transaction.</li> <li>• The length of the MerVarX should not exceed 50 characters.</li> <li>• MerVarX is case sensitive.</li> <li>• MerVarX can contain embedded white space(s).</li> </ul>
ProductType	<ul style="list-style-type: none"> <li>• Type of the credit card.</li> <li>• Must be AMEX.</li> <li>• ProductType should not contain embedded white space(s).</li> </ul>

ReturnURL	<ul style="list-style-type: none"> <li>• Globally accessible URL of your web site to which the Encrypted-Receipt is handed over.</li> <li>• ReturnURL should be a valid URL.</li> <li>• ReturnURL is case sensitive.</li> </ul>																						
TxnAmount	<ul style="list-style-type: none"> <li>• This is the amount to debit customer and credit merchant in specified currency.</li> <li>• The length of the TxnAmount variable should not exceed 12 numbers.</li> <li>• TxnAmount should not contain embedded white space(s).</li> <li>• Also the TxnAmount cannot have any alphabetic characters or special characters such as commas or hyphens.</li> <li>• TxnAmount MUST have the accurate number of digits right to the decimal point according to the specified currency.</li> </ul> <p>E.g.</p> <table data-bbox="478 974 1133 1590"> <thead> <tr> <th>CurrencyCode</th> <th>TxnAmount</th> </tr> </thead> <tbody> <tr> <td>LKR 100</td> <td>incorrect (must be 100.00)</td> </tr> <tr> <td>LKR 100.5</td> <td>incorrect (must be 100.50)</td> </tr> <tr> <td>LKR .5</td> <td>incorrect (must be 0.50)</td> </tr> <tr> <td>LKR .25</td> <td>incorrect (must be 0.25)</td> </tr> <tr> <td>KWD 100</td> <td>incorrect (must be 100.000)</td> </tr> <tr> <td>KWD 100.5</td> <td>incorrect (must be 100.500)</td> </tr> <tr> <td>KWD .5</td> <td>incorrect (must be 0.500)</td> </tr> <tr> <td>KWD .25</td> <td>incorrect (must be 0.250)</td> </tr> <tr> <td>JPY 100</td> <td>correct</td> </tr> <tr> <td>JPY 100.</td> <td>Incorrect (must be 100)</td> </tr> </tbody> </table>	CurrencyCode	TxnAmount	LKR 100	incorrect (must be 100.00)	LKR 100.5	incorrect (must be 100.50)	LKR .5	incorrect (must be 0.50)	LKR .25	incorrect (must be 0.25)	KWD 100	incorrect (must be 100.000)	KWD 100.5	incorrect (must be 100.500)	KWD .5	incorrect (must be 0.500)	KWD .25	incorrect (must be 0.250)	JPY 100	correct	JPY 100.	Incorrect (must be 100)
CurrencyCode	TxnAmount																						
LKR 100	incorrect (must be 100.00)																						
LKR 100.5	incorrect (must be 100.50)																						
LKR .5	incorrect (must be 0.50)																						
LKR .25	incorrect (must be 0.25)																						
KWD 100	incorrect (must be 100.000)																						
KWD 100.5	incorrect (must be 100.500)																						
KWD .5	incorrect (must be 0.500)																						
KWD .25	incorrect (must be 0.250)																						
JPY 100	correct																						
JPY 100.	Incorrect (must be 100)																						

E.g.

```
<req><action>SaleTxn</action><mer_id>newmerchant</mer_id><mer_txn_id>1234</mer_txn_id><txn_amt>1.00</txn_amt><cur>LKR</cur><ret_url>http://localhost:8080/ipgclient/Response.jsp</ret_url><lang>en</lang><mer_var1>MerVar1</mer_var1><mer_var2>MerVar2</mer_var2><mer_var3>MerVar3</mer_var3><mer_var4>MerVar4</mer_var4></req>
```

## Plain Text Receipt (Response Message)

Plain text receipt STRICTLY follows the below XML format. All the xml tags are in lower case. This plain text receipt does not contain embedded new line characters. Each xml tag is displayed on a new line below for clarity.

Depending on the value of 'action' tag other parameters on the response message may be mandatory or optional.

Please see section 9.0 Possible Values of 'action' tag for more details.

```
<res>
    <acc_no>MaskedAccNo</acc_no>
<action>Action</action>
<bank_ref_id>BankRefID</bank_ref_id>
<cur>CurrencyCode</cur>
<ipg_txn_id>IPGTransactionID</ipg_txn_id>
<lang>LanguageCode</lang>
<mer_txn_id>MerRefID</mer_txn_id>
<mer_var1>MerVar1</mer_var1>
<mer_var2>MerVar2</mer_var2>
<mer_var3>MerVar3</mer_var3>
<mer_var4>MerVar4</mer_var4>
<name>CustomerName</name>
<reason>FailReason</reason>
<txn_amt>TxnAmount</txn_amt>
<txn_status>TxnStatus</txn_status>
</res>
```

Value of Tag	Description
MaskedAccNo	<ul style="list-style-type: none"> <li>• Masked customer card number or account number from which the transaction amount debited.</li> <li>• The length of the MaskedAccNo does not exceed 20 characters.</li> </ul>
Action	<ul style="list-style-type: none"> <li>• Unique constant defined to identify type of transaction request / response.</li> <li>• Action is case sensitive.</li> <li>• <b>This is a mandatory parameter.</b></li> </ul>
BankRefID	<ul style="list-style-type: none"> <li>• This is a bank reference ID for transaction identification.</li> <li>• BankRefID may repeat after 999999.</li> </ul>
CurrencyCode	<ul style="list-style-type: none"> <li>• This is the same value set in the Plain Text Request Message.</li> </ul>
IPGTransactionID	<ul style="list-style-type: none"> <li>• This is a unique ID generated by the IPG Server for each transaction.</li> <li>• The length of the IPGTransactionID does not exceed 30 alphanumeric characters.</li> <li>• IPGTransactionID is case sensitive.</li> <li>• IPGTransactionID does not contain embedded white space(s).</li> </ul>
LanguageCode	<ul style="list-style-type: none"> <li>• This is the same value set in the Plain Text Request message.</li> </ul>
MerRefID	<ul style="list-style-type: none"> <li>• This is the same value set in the Plain Text Request Message.</li> </ul>
MerVarX	<ul style="list-style-type: none"> <li>• This is the same value set in the Plain Text Request Message.</li> </ul>
CustomerName	<ul style="list-style-type: none"> <li>• Name of the customer as typed on the IPG Server payment pages.</li> <li>• The length of the CustomerName does not exceed 50 characters.</li> </ul>
FailReason	<ul style="list-style-type: none"> <li>• Short description for transaction failed reason.</li> </ul>

	<ul style="list-style-type: none"> <li>• If transaction status is ACCEPTED FailReason is blank.</li> </ul>
TxnAmount	<ul style="list-style-type: none"> <li>• This is the same value set in the Plain Text Request Message.</li> </ul>
TxnStatus	<ul style="list-style-type: none"> <li>• Transaction status to denote whether this transaction was successful or fail.</li> <li>• Returns ACCEPTED for successful transaction.</li> <li>• Returns REJECTED for failed transaction.</li> <li>• TxnStatus is case sensitive.</li> </ul>

E.g.

### Payment Authorization Successful Response Message

```
<res><ipg_txn_id>NEWMERCHANT1176147506172</ipg_txn_id><reason></reason><mer_var3>MerVar3</mer_var3><mer_var1>MerVar1</mer_var1><txn_amt>1.00</txn_amt><action>SaleTxn</action><bank_ref_id>107849</bank_ref_id><lang>en</lang><cur>LKR</cur><mer_txn_id>12345</mer_txn_id><mer_var2>MerVar2</mer_var2><mer_var4>MerVar4</mer_var4><txn_status>ACCEPTED</txn_status><name>A B C Perera</name><acc_no>371234XXXXXXXX56</acc_no></res>
```

### Payment Authorization Fail Response Message

```
<res><ipg_txn_id>NEWMERCHANT1176147223545</ipg_txn_id><reason>>ERR_04_11 – Internal error. Contact bank administrator with error code.</reason><mer_var3>MerVar3</mer_var3><mer_var1>MerVar1</mer_var1><txn_amt>1.00</txn_amt><action>SaleTxn</action><bank_ref_id>107848</bank_ref_id><lang>en</lang><cur>LKR</cur><mer_txn_id>1234</mer_txn_id><mer_var2>MerVar2</mer_var2><mer_var4>MerVar4</mer_var4><txn_status>REJECTED</txn_status><name>A B C Perera</name><acc_no>371234XXXXXXXX56</acc_no></res>
```

### Static return URL

This is a return URL which is fed to the IPG server system while merchant registration. Merchant should produce this static return URL to the bank with merchant's public key (mypubkey.txt). IPG server will hand over session of each completed transaction to this static return URL if Static return URL for failure is NOT provided and Dynamic return URL is NOT provided. If merchant web application is designed to accept the status of transaction by a single return URL, then it is easy to set this URL once rather setting the same URL for each transaction. Therefore, merchant can use static return URL given to the bank for receiving transaction status. He need not set the value for <ret\_url></ret\_url> tag for each transaction. IPG server will hand over session to Static return URL for both successful and failed transactions.

## Chapter 5

### Sub Component Details

#### 5.1 Screen Item Description

Table Name	Field Name	Screen Item Name	Length	Data type	CRUD (Create/ Read/ Update/ Delete)
BANKS	ENABLED	Status	1	VARCHAR2	C, R, U
	BANK_NAME	Bank	255	VARCHAR2	C, R, U
	BANK_ID	Bank Code	22	NUMBER	C,R
BRANCHES	ENABLED	Status	1	VARCHAR2	C, R, U
	BRANCH_ADD	Address	65	VARCHAR2	C, R, U
	BRANCH_NAME	Branch	255	VARCHAR2	C, R, U
	BRANCH_ID	Nil	22	NUMBER	C,R
	BANK_ID	Nil	22	NUMBER	C,R
	ENABLED	Status	1	VARCHAR2	C, R, U
CURRENCY_LIST	NUMERIC_CODE	Nil	100	VARCHAR2	C, R, U
	ALPHABETIC_CODE	Nil	100	VARCHAR2	C, R, U
	CURRENCY	Currency	255	VARCHAR2	C,R, U, D
	COUNTRY	Country	255	VARCHAR2	C, R, U
	ID	Nil	22	NUMBER	C, R
FEE_STRUCTURE	OTHERS_FEE	Others	22	NUMBER	C,R, U, D
	RATE_OF_FEE	Fee	22	NUMBER	C, R, U, D
L_PAYMENTS_GATEWAYS	STATUS	Status	1	VARCHAR2	C, R, U
	GATEWAY_NAME	Nil	30	VARCHAR2	C, R, U
	GATEWAY_ID	Nil	22	NUMBER	C, R, U, D
L_USER_TYPES	ENABLED	Status	1	VARCHAR2	C, R, U, D
	USER_TYPES_NAME	User Type	30	VARCHAR2	C, R, U, D
	USER_TYPE_ID	Nil	22	NUMBER	C, R, U
L_WORKFLOW	UPDATE_DATE	Date	7	DATE	C, R, U
	UPDATE_BY	Update By	22	NUMBER	C, R, U
	CREATE_DATE	Date	7	DATE	C, R, U
	CREATE_BY	Create By	22	NUMBER	C, R, U
	SEND_EMAIL	y/n	1	VARCHAR2	C, R, U
	SEND_SMS	y/n	1	VARCHAR2	C, R, U
	INSPECTION_STAGE	y/n	1	VARCHAR2	C, R, U
	ENABLED	Status	1	VARCHAR2	C, R, U
	OPTIONAL_YN	y/n	1	VARCHAR2	C, R, U
	VIEW_DOC_YN	y/n	30	VARCHAR2	C, R, U
	STAGE	Stage	22	NUMBER	C, R, U
	DESIGNATION_ID	Nil	22	NUMBER	C, R, U, D
	STAFF_ID	Nil	22	NUMBER	C, R, U, D
APP_ID	Nil	22	NUMBER	C, R, U	

	WF_ID	Nil	22	NUMBER	C, R, U
MESSAGE	STATUS	Status	1	VARCHAR2	C, R, U
	VIEW_YN	y/n	1	VARCHAR2	C, R, U
	CREATE_BY	Create By	22	NUMBER	C, R, U
	CREATE_DATE	Date	7	DATE	C, R, U
	TO_USER	Forward to	22	NUMBER	C, R, U
	SMS_MESSAGE	y/n	140	VARCHAR2	C, R, U
	EMAIL_MESSAGE	y/n	2,000	VARCHAR2	C, R, U
	WEB_MESSAGE	y/n	2,000	VARCHAR2	C, R, U
	TRACKING_NO	Tracking No	22	NUMBER	C, R, U
	MSG_ID	Nil	22	NUMBER	C, R, U
	STAGE	Stage	22	NUMBER	C, R, U
REPLAY_OF	Nil	22	NUMBER	C, R, U, D	
M_APPLICATIONS_TYPES	ENABLED	Status	1	VARCHAR2	C, R, U, D
	APP_TYPE_NAME	Application Type	50	VARCHAR2	C, R, U
	APP_TYPE_ID	Nil	22	NUMBER	C, R, U
	WORKING_DAYS	Duration	22	NUMBER	C, R, U
	APP_ID	Nil	22	NUMBER	C, R, U
M_DESIGNATION	STATUS	Nil	1	VARCHAR2	C, R, U
	TITLE	Title	50	VARCHAR2	C, R, U
	DESIGNATION_ID	Nil	22	NUMBER	C, R, U
M_ROLES	CREATE_DATE	Date	7	DATE	C, R, U
	CREATE_BY	Create By	22	NUMBER	C, R, U
	ROLE_NAME	Role	60	VARCHAR2	C, R, U
	ROLE_ID	Nil	22	NUMBER	C, R, U
PAYMENT_INFO	MOBILE_NO	Nil	22	NUMBER	C, R, U
	STATUS	Nil	22	NUMBER	C, R, U, D
	REMARKS	Nil	100	VARCHAR2	C, R, U, D
	TRAN_DATE	Nil	7	DATE	C, R, U
	TXN_AMT	Nil	22	NUMBER	C, R, U
	AUTH_CODE	Nil	30	VARCHAR2	C, R, U
	MER_TXN_ID	Nil	30	VARCHAR2	C, R, U
	IPG_TXN_ID	Nil	30	VARCHAR2	C, R, U
	BANK_REF_ID	Nil	22	NUMBER	C, R, U
	ACC_NO	Nil	30	VARCHAR2	C, R, U
	USER_ID	Nil	22	NUMBER	C, R, U
	TRACKING_NO	Nil	22	NUMBER	C, R, U
	GATEWAY_ID	Nil	22	NUMBER	C, R, U
	PAYMENT_ID	Nil	22	NUMBER	C, R, U
	BRANCH_NAME	Nil	512	VARCHAR2	C, R, U
	DRAFT_DATE	Nil	7	DATE	C, R, U, D
	DRAFT_NO	Nil	50	VARCHAR2	C, R, U, D
BRANCH_ID	Nil	22	NUMBER	C, R, U	
BANK_ID	Nil	22	NUMBER	C, R, U	
STAFF	ENABLED	Status	1	VARCHAR2	C, R, U
	OFFICE_ID	Nil	22	NUMBER	C, R, U
	ADDRESS	Address	150	VARCHAR2	C, R, U
	CONTACT_NO	Contact No	22	NUMBER	C, R, U
	DESIGNATION_ID	Nil	22	NUMBER	C, R, U
	STAFF_NAME	Employee Name	100	VARCHAR2	C, R, U
	USER_ID	Nil	22	NUMBER	C, R, U

	STAFF_ID	Nil	22	NUMBER	C, R, U
USERS	STAFF_YN	y/n	1	VARCHAR2	C, R, U
	ENABLED	Status	1	VARCHAR2	C, R, U
	ACTIVE_HASH	Activation Code	1,000	VARCHAR2	C, R, U, D
	LAST_LOGIN	Last Login	7	DATE	C, R, U, D
	CREATE_DATE	Date	7	DATE	C, R, U
	CREATE_BY	Nil	22	NUMBER	C, R, U
	USER_TYPE_ID	Nil	22	NUMBER	C, R, U
	DEFAULT_PASSWORD	Nil	1,000	VARCHAR2	C, R, U
	PASSWORD	Password	1,000	VARCHAR2	C, R, U
	PHONE_NO	Phone No	22	NUMBER	C, R, U
	EMAIL_ID	Nil	40	VARCHAR2	C, R, U
	USER_NAME	Name	50	VARCHAR2	C, R, U
USER_ID	Nil	22	NUMBER	C, R, U	
USER_ROLE S	ENABLED	Status	1	VARCHAR2	C, R, U
	UPDATE_DATE	Date	7	DATE	C, R, U
	UPDATE_BY	Update BY	22	NUMBER	C, R, U
	CREATE_DATE	Create Date	7	DATE	C, R, U, D
	CREATE_BY	Cerate By	22	NUMBER	C, R, U, D
	ROLE_NAME	Role	60	VARCHAR2	C, R, U
	USER_ID	Nil	22	NUMBER	C, R, U
	ROLE_ID	Nil	22	NUMBER	C, R, U
WORKFLO W	USER_ROLES_ID	Nil	22	NUMBER	C, R, U
	WF_UPDATE_DATE	Nil	7	DATE	C, R, U
	WF_UPDATE_BY	Nil	22	NUMBER	C, R, U
	SEND_EMAIL	y/n	1	VARCHAR2	C, R, U
	SEND_SMS	y/n	1	VARCHAR2	C, R, U
	UPDATE_DATE	Date	7	DATE	C, R, U
	CREATE_DATE	Date	7	DATE	C, R, U
	NOTE	Remarks	200	VARCHAR2	C, R, U
	APPROVAL_YN	y/n	1	VARCHAR2	C, R, U
	OPTIONAL_YN	y/n	1	VARCHAR2	C, R, U, D
	DESIGNATION_ID	Nil	22	NUMBER	C, R, U, D
	STAFF_ID	Nil	22	NUMBER	C, R, U
	VIEW_DOC_YN	y/n	30	VARCHAR2	C, R, U
	STAGE	Stage	22	NUMBER	C, R, U
	WF_ID	Nil	22	NUMBER	C, R, U
	APP_ID	Nil	22	NUMBER	C, R, U
	TRACKING_NO	Tracking No	22	NUMBER	C, R, U
ID	Nil	22	NUMBER	C, R, U	
INSPECTION_STAGE	y/n	1	VARCHAR2	C, R, U	
FORWARDED_BY_ID	Nil	22	NUMBER	C, R, U	
SMS_HISTO RY	SMS_ID	ID		NUMBER	C, R, U
	SMS_TEXT	Text		VARCHAR2	C, R, U
	SMS_DATE	Date		DATE	C, R, U
	USER_ID	Nil		NUMBER	C, R, U
MSG_HISTO RY	MSG_ID	ID		NUMBER	C, R, U
	MSG_TEXT	Text		VARCHAR2	C, R, U
	MSG_DATE	Date		DATE	C, R, U
	MSG_ID	Nil		NUMBER	C, R, U



## 5.2 Error and Exception Handling

### System Error List:

SL	Error Code	Error Message
1	901	USER NOT FOUND
2	902	REQUIRED VALUE CANNOT BE NULL
3	903	USER NOT PERMITTED TO CREATE ANOTHER USER
4	904	ERROR IN GENERATING USER ID
5	905	DUPLICATE REGISTRATON NO
6	906	YOU CAN NOT DELETE THIS REGISTRATION NO
7	907	INVALID INVETSMENT TYPE ID
8	908	YOU CANNOT DELETE THIS ADDRESS, AS IT IS USED BY INDUSTRY PROFILE
9	909	INVALID INVSETMENT SOURCE TYPE
10	910	YOU CANNOT DELETE THIS USER GROUP.
11	911	INVALID SUB-SECTOR ID
12	912	YOU CANNOT DELETE THIS MODE.
13	913	INVALID ACCOUNT TYPE ID
14	914	YOU CANNOT DELETE THIS TYPE.
15	915	INVALID DEPARTMENT ID
16	916	YOU CAN NOT DELETE THIS DEPARTMENT, AS IT IS LINKED WITH ANOTHER ONE
17	917	INVALID DESIGNATION ID
18	918	YOU CAN NOT DELETE THIS DESIGNATION, AS IT IS LINKED WITH ANOTHER ONE
19	919	INVALID BANK TYPE ID
20	920	YOU CANNOT DELETE THIS BANK, AS IT IS ALREADY IN USE!
21	921	INVALID ROLE ID
22	922	YOU CANNOT DELETE THIS ROLE, AS IT IS ASSIGNED TO USER!
23	923	INVALID SUB-ACCOUNT ID
24	924	YOU CANNOT DELETE THIS SUB-ACCOUNT, AS IT IS LINKED WITH ACCOUNT GROUP!
25	925	INVALID USERTYPE ID
26	926	YOU CANNOT DELETE THIS USERTYPE, AS IT IS ASSIGNED TO USER!
27	928	INVALID USER ID/PASSWORD
28	929	YOU CAN NOT EDIT THIS ACCOUNT, AS IT IS LINKED WITH ANOTHER ACCOUNT
29	930	NO_DATA_FOUND

## API Error List:

Error Code	Error Message	Error Description
0	Successful.	No errors.
- 1	Failed to load JCE Provider.	Check your class path is accurately set to provided JAR libraries.
- 2	Internal error, Contact bank administrator with log file and stack trace.	Check your class path is accurately set to provided JAR libraries. If the problem exists contact bank administrator with log file and stack trace.
- 3	myprivkey.txt file not found.	Please check your security key path is accurately set to folder which contains myprivkey.txt file OR myprivkey.txt file exists in default security key path folder.
- 4	My Private Key not loaded.	Contents of myprivkey.txt file may have been corrupted. Re-generate merchant keys and re-exchange.
- 5	ipgpubkey.txt file not found.	Please check your security key path is accurately set to folder which contains ipgpubkey.txt file OR ipgpubkey.txt file exists in default security key path folder.
- 6	IPG Public Key not loaded.	Contents of ipgpubkey.txt file may have been corrupted. Request latest ipgpubkey.txt file from the bank.
- 7	Internal error, Contact bank administrator with log file and stack trace.	Please re-generate merchant keys and get the latest ipgpubkey.txt from the bank. If the problem exists contact bank administrator with log file and stack trace.
- 8	Internal error, Contact bank administrator with log file and stack trace.	Please re-generate merchant keys and get the latest ipgpubkey.txt from the bank. If the problem exists contact bank administrator with log file and stack trace.
- 9	Internal error, Contact bank administrator with log file and stack trace.	Please re-generate merchant keys and get the latest ipgpubkey.txt from the bank. If the problem exists contact bank administrator with log file and stack trace.
- 10	mypubkey.txt file not found.	Please check your security key path is accurately set to folder which contains mypubkey.txt file OR mypubkey.txt file exists in default security key path folder.
- 11	My Public Key not loaded.	Contents of mypubkey.txt file may have been corrupted. Re-generate merchant keys and re-exchange.
- 12	Internal error, Contact bank administrator with log file and stack trace.	Check your class path is accurately set to provided JAR libraries. If the problem exists contact bank administrator with log file and stack trace.
- 13	Internal error, Contact bank administrator with log file and stack trace.	Check your class path is accurately set to provided JAR libraries. If the problem exists contact bank administrator with log file and stack trace.
- 14	Internal error, Contact bank administrator with log file and stack trace.	Check your class path is accurately set to provided JAR libraries. If the problem exists contact bank

- 15	Encrypted Receipt is null or blank.	Please check Encrypted receipt is null or blank before invoking setEncryptedReceipt() method.
- 16	Encrypted Receipt has been tampered, Encrypted Receipt MUST contain alphanumeric characters only.	Encrypted Receipt has been corrupted or tampered. Cannot decrypt.
- 17	Signature verification failed. Request latest ipgpubkey.txt file from the bank.	May be using an older version of ipgpubkey.txt file. Request latest ipgpubkey.txt file from the bank.
- 18	Plain Text Invoice is null or blank.	Please check Plain text invoice is null or blank before invoking setPlainTextInvoice() method.
- 19	mypubkey.txt and myprivkey.txt do not tally. Re-generate merchant keys and re-exchange.	mypubkey.txt and myprivkey.txt do not tally. Re-generate merchant keys and re-exchange.
- 20	Internal error, Contact bank administrator with log file and stack trace.	Check your class path is accurately set to provided JAR libraries. If the problem exists contact bank administrator with log file and stack trace.
- 21	ipgpubkey.txt file corrupted, Request latest ipgpubkey.txt file from the bank.	ipgpubkey.txt file corrupted, Request latest ipgpubkey.txt file from the bank.
- 22	myprivkey.txt file is corrupted, Re-generate merchant keys and re-exchange.	myprivkey.txt file is corrupted, Re-generate merchant keys and re-exchange.
- 23	mypubkey.txt file is corrupted, Re-generate merchant keys and re-exchange.	mypubkey.txt file is corrupted, Re-generate merchant keys and re-exchange.
- 24	Encrypted receipt is corrupted or tampered.	Encrypted Receipt has been corrupted or tampered. Cannot decrypt.
- 25	Encrypted receipt is corrupted or tampered.	Encrypted Receipt has been corrupted or tampered. Cannot decrypt.
-101	Registry entry read error.	Please uninstall IPG2 client software and re-install.(Only in IIS Version)
-101	Registry entry read error.	Please uninstall IPG2 client software and re-install.(Only in IIS Version)
-102	Java Virtual Machine load error.	Please re-check all IPG2 client setup configurations. If the problem exists contact bank administrator.(Only in IIS Version)
-103	Java class find error.	Please re-check all IPG2 client setup configurations. If the problem exists contact bank administrator.(Only in IIS Version)
-104	Java method find error.	Please re-check all IPG2 client setup configurations. If the problem exists contact bank administrator.(Only in IIS Version)
-105	Java error code invoke error.	Please re-check all IPG2 client setup configurations. If the problem exists contact bank administrator.(Only in IIS Version)
-106	Essential parameters are missing, blank or null.	Re-check parameters passed to all API methods.(Only in IIS Version)
-199	Unknown error.	Please re-check all IPG2 client setup configurations. If the problem exists contact bank administrator.(Only in IIS Version)

### 5.3 Report Layout and Description

#### 5.3.1 Report: Payment Module

Monthly Payment Report					
SL#	Application	Application Type	Payment Date	Application Status	Amount
1	REGISTRATION	PROPOSED REGISTRATION	15-Dec-15	APPROVED	5000000.00
2	REGISTRATION	PROPOSED REGISTRATION	23-Dec-15	APPROVED	42361.00
3	REGISTRATION	PROPOSED REGISTRATION	29-Dec-15	APPROVED	42368.00
4	REGISTRATION	PROPOSED REGISTRATION	14-Dec-15	APPROVED	12242.00
5	INVESTMENT RELATED CORRECTION	INVESTMENT RELATED CORRECTION	14-Dec-15	APPROVED	42777.00
6	COMPOSITE CERTIFICATE	COMPOSITE CERTIFICATE	21-Dec-15	APPROVED	454545.00

#### 5.3.2 Report Name: Master Workflow List

S L #	Application	Stage	Staff (Designation)	Doc. View	Optional Stage	Send SMS	Send EMAIL	Status
1	REGISTRATION	1	ONE STOP (ONE STOP)	YES	NO	NO	YES	ACTIVE
2	REGISTRATION	2	director 01 (DIRECTOR)	YES	YES	NO	NO	ACTIVE
3	REGISTRATION	3	AD 01 (AD)	YES	YES	NO	NO	ACTIVE
4	REGISTRATION	4	Inspector 01 (INSPECTOR)	YES	YES	NO	NO	ACTIVE
5	REGISTRATION	5	AD 01 (AD)	YES	NO	NO	NO	ACTIVE

#### 5.3.3 Report: Workflow Module

Stage	Staff Name	Designation	Update Date	Note	Status
1	ONE STOP	ONE STOP	12/23/2015 15:03	to verify	<b>APPROVED</b>
2	DIRECTOR 01	DIRECTOR	12/23/2015 15:07	ins	<b>APPROVED</b>
3	AD 01	AD	12/23/2015 15:09	ok	<b>APPROVED</b>
4	INSPECTOR 01	INSPECTOR	12/23/2015 15:11	ok	<b>APPROVED</b>
5	AD 01	AD	12/23/2015 15:20	ok	<b>APPROVED</b>
6	DD 01	DD			<b>ON PROCESS</b>
7	DIRECTOR 01	DIRECTOR			
8	AD 01	AD			

## Chapter 6

### Integration Environment and Sequence

#### 6.1 Integration Environment

Table 4: Proposed Specification of Operating Server(IE)

<b>Proposed Specification of Operating Server</b>	
Processor	Intel Xeon Quad Core Processor
Memory	32 GB ECC
Storage	3 SAS Disks
Mirroring	Raid Level 3
Network	Gigabit Redundant NIC
Backup	USB Based Tape Device (80GB)
Power	Redundant Power Supply (online)

Table 5: Proposed Specification of Client Workstations(IE)

<b>Proposed Specification of Client Workstations</b>	
<b>Processor</b>	Intel Dual Core
<b>Memory</b>	1 GB
<b>Storage</b>	250GB HDD
<b>Network</b>	10/100 Mbps Ethernet + Wireless LAN + CDMA/Edge/GPRS/3G Modem

## 6.2 Integration Sequence:

Payment gateway and SMS gateway integration are describing in below:

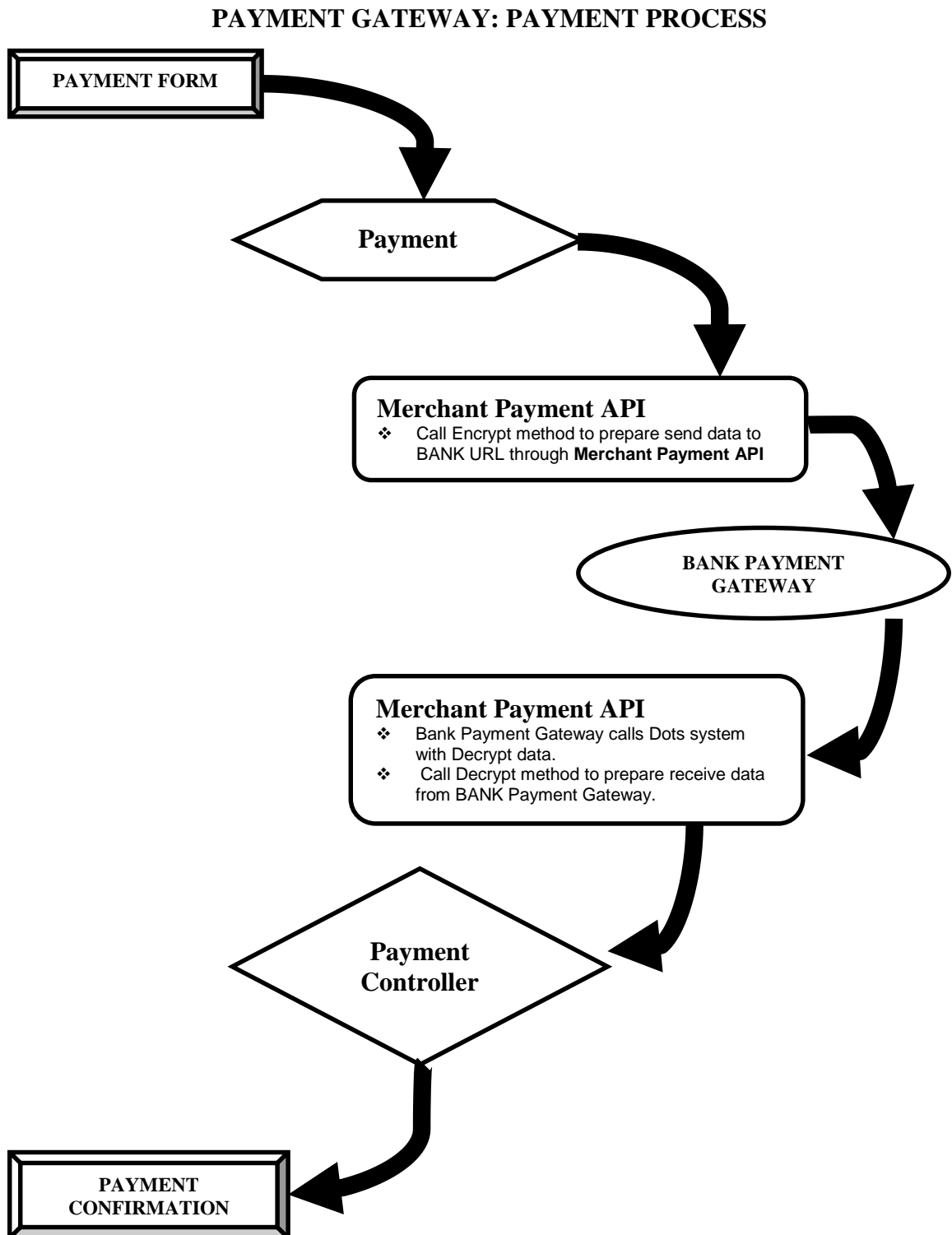


Figure 5: Payment Gateway

## SMS GATEWAY: MESSAGE SENDING PROCESS

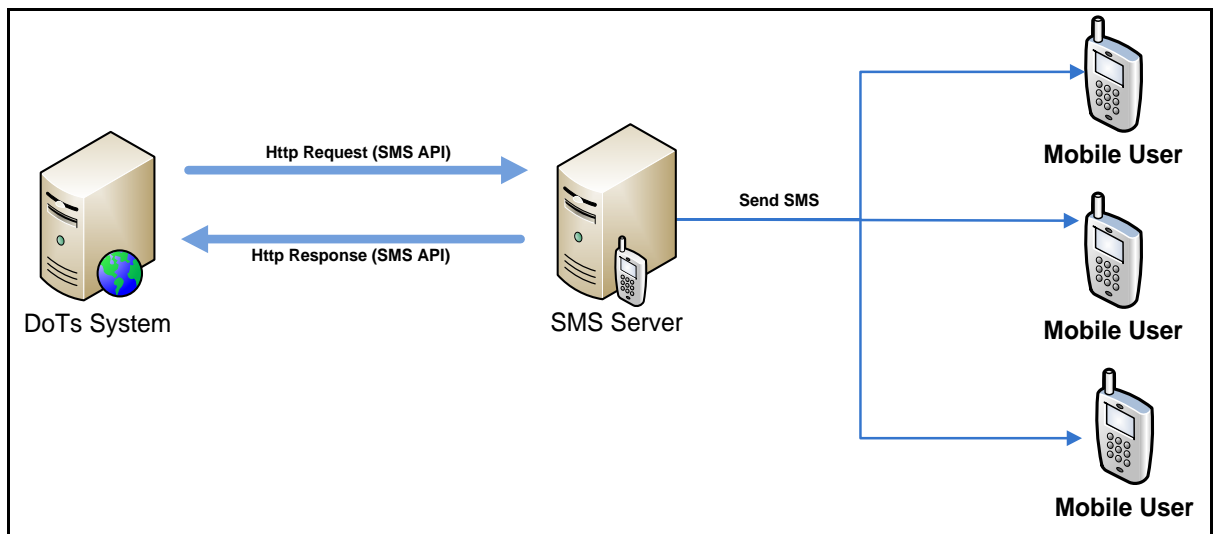


Figure 6: SMS Gateway

- The web panel IP: <http://123.49.3.58:8081/>
- For automated system you can use [http://123.49.3.58:8081/web\\_send\\_sms.php?ms=8801XXXXXXXXXX&txt=sms content&username=User ID&password=Password](http://123.49.3.58:8081/web_send_sms.php?ms=8801XXXXXXXXXX&txt=sms content&username=User ID&password=Password)

The above mentioning link contains parameter.

Note: Parameter "ms= 88Mobile no(13 digits)." and "txt= Your 160 character message"

### **6.3 Entry-Exit Criteria**

**The inputs from the integration phase include:**

- Defined and Approved Requirements
- Integration Plan
- Appropriate Integration Environment

**The following exit criteria should be considered for completion of a Integration phase:**

- Achieving complete Functional Coverage
- Executing the major functional/business flows successfully by leveraging various inputs and ensuring that they are working fine
- Fixing all the ‘Show Stopper defects’ or ‘Blockers’ and ensuring that none of the identified Critical/Severity 1 defects are in Open Status
- Re-testing and closing all the high-priority defects to execute corresponding Regression scenarios successfully

**The following output is achieved through the exit criteria:**

- Integration Test Logs
- Integration Test Summary Report/Findings Report



## **Chapter 7**

### **Conclusion**

Database server and application server will install in separate server. Source code of DOTS will have placed in htdocs folder in apache web server of application server. Database server will be offline. Only application server can access by internet.

User training was conducted based on different level of user. Theoretical and Hands-on both type of training was arranged. User manual was prepared for better uses.

Project has been completed within the allocated budget in terms of time and cost. Lessons Learned ,Best Practices and Improvements are documented, and sent to the SEPG for the knowledge repository. Metrics was produced after the project was delivered, and the minutes were documented in the closure meeting

# **Appendix**

## **Definitions and Acronyms**

**DOT** – Department of Textiles

**QMS**-Quality Management System

**OS**- Operating system

**SRS**- System Requirement Specification

**UML** -Unified Modeling Language