Software Design Specification

UNION BANK ONLINE SHOPPING PLATFORM

Software Design Specification
Version 1.0, March 2017

Document Control

This document is maintained under change control. All changes must be formally documented and agreed by the project team before they are put into effect.

Document Information

	Information
Document Id	Union bank online shopping platform Software design specification
Document Owner	DEVOPS
Issue Date	*03/03/2017*
Last Saved Date	*03/03/2017*
File Name	Union bank online shopping platform SDS.docx

Author

Version	Issue Date	Changed by	Changes
[1.0]	*03/03/2017*	*Samuel Ezeala*	*Baseline*

Reviewers/Change Control

Version	Date	Author	Comments
[1.0]	*03/03/2017*	*Adebowale Owoseni*	

Document Approvals

Designation	Name	Signature	Date
Team Lead, Core Programmes	Ayodeji Ajibade		
Head, Devops	Wale Akanni		

Table of Contents

System Overview

I. Introduction

The process through which Corporate Communication department sells branded material to staff at the moment is very manual and is froth with rigors. it involves sending emails back and forth, tracking via excel sheets, register and so on.

The solution manages this process by providing a centrally accessible platform to all staff that automates the process and also affords better tracking and archiving for better record keeping as necessary Process Objective

The purpose of this solution is make the buying experience of staff and management of the office store a more effective and efficient one.

Benefit to Customer

- Convenient way of purchasing branded material from the bank
- Proper and effective monitoring of requested branded material
- Record keeping is encouraged in the process
- Tracking of requested materials is made easy Benefit to the Bank

II. Functional Requirement

The user requirements and functional requirement are outlined in the table below, as indicated on the BRS. Some requirements will be added base on systems exigencies, but will not alter or conflict with the core requirements and functionalities.

ID	Requirements Reviewed	Comment
R01	Sign On with AD credentials	OK
R02	Display all branded materials available	OK
R03	Display the total amount of the selected branded material and an OK button	OK
R04	Pop up a no refund policy with a continue button	OK
R05	Allow customer select preferred delivery location	OK
R06	Allow customer enter account number	OK
R07	Validate Token field for SMS token	OK
R08	Send email to customer after successful payment	OK
R09	Display list of all pending order to admin	OK
R10	Display a Proceed and Refund button	OK
R11	Send Email to customer on Proceed	OK
R12	Reverse customers fund to the debited account on refund	OK
R13	Send an alert to staff that request have been rejected with reasons.	OK

III. System Process Flow

The figure below show a process workflow of the application. The flowchart captures the two stakeholders namely the customer and store Admin. The application is not limited to the activities captured in the process flow, but due to space constrain, the process flow only focuses and the core activities require of the application. The swimlane diagram in Figure one below shows the interaction between the user and the system, different touch points and the behavior at each point.

IV. System User

The key actors are the customers and the admin.

The admin has two validation levels, first is the Active Directory validation to confirm the user is a staff bank, why the second validation is done against a database to validate the user role if he/she is profiled on the system as an administrator. If the user is not profiled on the application as an administrator and Authorized error is displayed back to the user, but for a successful validation the user is redirected into the admin page the list of all pending orders is displayed.

On the home page of the admin the user can either process the order/refund the order or view the details of the order as specified by the BRS. Other functions that can be carried out by the admin includes:

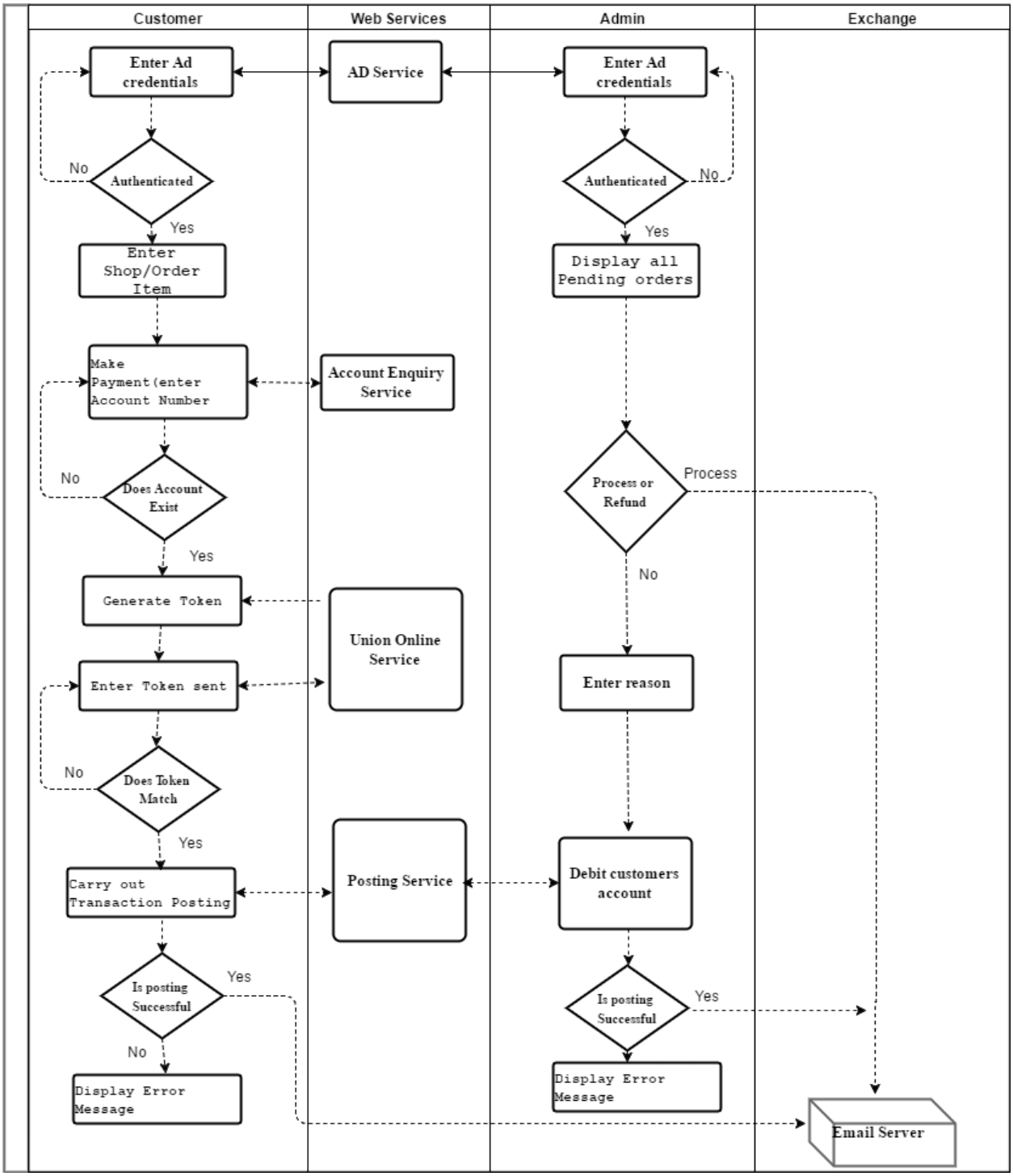

Figure 1. System Process Flow.

- Adding Category (CRUD)
- Adding Product (CRUD)
- Adding Users (Admin user)
- View Audit Logs
- View all Orders place on the platform etc.

The customer has just one level of validation, which is Active directory validation. The customer will need to sign on to the application in order to view the list of items on sales. The application provides variety of options for the user to query the items, such as searching by categories on the cart.

An auto decrement is put in place on the database for each product purchase, when the product quantity count gets to zero (0), the product gets disabled automatically and shows out of stock on the customer's page.

The customer can add directly to cart from the product list page or click on the item to get a full description of the item being purchase. For each item added to cart, an AJAX request is made to the server and reloads the page on the event that the item get successfully added to cart with an item count displayed on the top right corner of the application which is also a link in to the cart view.

On the cart page a list of items added to cart is displayed and the sum total, with an option to either remove an item from the cart or update the item quantity and also a checkout.

On checkout a no refund policy pops up and also a list of branches for the customer to select from, as delivery location, the delivery location gets stored in a session variable. The next page shows the account page where the user is expected to enter a valid UBN account. On account validation a token is generated and sent to the uses phone as an SMS with another input field displayed below for the user to enter the token sent to their phone. After a successful validation of the token sent the users account gets debited with amount of total item being purchase and the union store account is credited with the same amount. This whole transaction is handled by the Union Online web service for token generation and validation, posting service for debiting and crediting account.

The application also keeps an order history log for the user to view the list of previous orders made on the platform.

V. Database Design

The diagram below describes the database tables of the application.

Admin User Table

Columns | Indexes | Constraints | Triggers | Data | Script | Grants | Synonyms | Partitions | Subp

Column Name	ID	PK	Index Pos	Null?	Data Type
USERID	1			N	INTEGER
USERNAME	2			Y	VARCHAR2 (50 Byte)
FULLNAME	3			Y	VARCHAR2 (100 Byte)
EMPNUM	4			Y	VARCHAR2 (15 Byte)
BRANCHCODE	5			Y	VARCHAR2 (5 Byte)
BRANCHNAME	6			Y	VARCHAR2 (150 Byte)
ISACTIVE	7			Y	VARCHAR2 (1 Byte)
TITLE	8			Y	VARCHAR2 (200 Byte)
DEPARTMENT	9			Y	VARCHAR2 (255 Byte)
EMAIL	10			Y	VARCHAR2 (200 Byte)

Category Table

Column Name	ID	PK	Index Pos	Null?	Data Type
CATEGORYID	1			N	INTEGER
CATEGORYNAME	2			Y	VARCHAR2 (100 Byte)
ISACTIVE	3			Y	VARCHAR2 (1 Byte)
ISDELETE	4			Y	VARCHAR2 (1 Byte)

Login/Logout Log Table

Column Name	ID	PK	Index Pos	Null?	Data Type
LOGID	1			N	INTEGER
USERNAME	2			N	VARCHAR2 (30 Byte)
EMPNUM	3			N	VARCHAR2 (15 Byte)
LOGINDATE	4			Y	DATE
LOGINTIME	5			Y	VARCHAR2 (15 Byte)
LOGOUTDATE	6			Y	DATE
LOGOUTTIME	7			Y	VARCHAR2 (15 Byte)
FULLNAME	8			Y	VARCHAR2 (100 Byte)

Posting Log Table

LOGGID	1			N	INTEGER
REQUESTID	2			N	INTEGER
PAYMENTREF	3			N	VARCHAR2 (50 Byte)
BATCHID	4			N	VARCHAR2 (50 Byte)
INITIATINGBRANCH	5			Y	VARCHAR2 (5 Byte)
ACCOUNTNUMBER	6			Y	VARCHAR2 (15 Byte)
CREDIT	7			N	INTEGER
CURRENCY	8			Y	VARCHAR2 (3 Byte)
NARRATION	9			Y	VARCHAR2 (255 Byte)
TRANSACTIONSTARTDATE	10			Y	DATE
TRANSACTIONENDDATE	11			Y	DATE
STATUS	12			N	VARCHAR2 (20 Byte)
GLCASAINDICATOR	13			Y	VARCHAR2 (10 Byte)
RESPONSECODE	14			Y	VARCHAR2 (5 Byte)
RESPONSEMSG	15			Y	VARCHAR2 (255 Byte)

Events Log Table

Column Name	ID	PK	Index Pos	Null?	Data Type
LOGID	1			Y	INTEGER
EVENT	2			Y	VARCHAR2 (255 Byte)
ORDERID	3			Y	VARCHAR2 (255 Byte)

Order Details Table

Column Name	ID	PK	Index Pos	Null?	Data Type
ORDERID	1			N	VARCHAR2 (255 Byte)
FULLNAME	2			N	VARCHAR2 (255 Byte)
EMAIL	3			N	VARCHAR2 (100 Byte)
EMPNUM	4			Y	VARCHAR2 (20 Byte)
BRANCH	5			Y	VARCHAR2 (150 Byte)
PICKUPBRANCH	6			Y	VARCHAR2 (150 Byte)
TOTALAMOUNT	7			Y	INTEGER
STATUS	8			Y	VARCHAR2 (10 Byte)
ORDERDATE	9			Y	DATE
ORDERTIME	10			Y	VARCHAR2 (10 Byte)
ADMINSTATUS	11			Y	VARCHAR2 (10 Byte)
COMMENTS	12			Y	VARCHAR2 (255 Byte)

Ordered Items Table

Column Name	ID	PK	Index Pos	Null?	Data Type
ORDERID	1			N	VARCHAR2 (255 Byte)
PRODUCTID	2			N	INTEGER
QUANTITY	3			N	INTEGER
ITEMTOTAL	4			N	INTEGER

Product Table

Column Name	ID	PK	Index Pos	Null?	Data Type
PRODUCTID	1			N	INTEGER
PRODUCTNAME	2			Y	VARCHAR2 (100 Byte)
CATEGORYID	3			Y	INTEGER
ISACTIVE	4			Y	VARCHAR2 (8 Byte)
ISDELETE	5			Y	VARCHAR2 (8 Byte)
CREATEDDATE	6			Y	DATE
MODIFIEDDATE	7			Y	DATE
DESCRIPTION	8			Y	VARCHAR2 (4000 Byte)
ADDITIONALINFO	9			Y	VARCHAR2 (4000 Byte)
PRICE	10			Y	INTEGER
ISFEATURED	11			Y	VARCHAR2 (8 Byte)
QUANTITY	12			Y	VARCHAR2 (255 Byte)

Product Image Table

Column Name	ID	PK	Index Pos	Null?	Data Type
MAINIMG	1			Y	VARCHAR2 (200 Byte)
SUBIMG_I	2			Y	VARCHAR2 (200 Byte)
SUBIMG_II	3			Y	VARCHAR2 (200 Byte)
SUBIMG_III	4			Y	VARCHAR2 (200 Byte)
PRODUCTID	5			Y	INTEGER

VI. System Implementation

S/No	Task	Duration (Days)	Start Date	End Date	Responsible Team/officer
Design (3 days)					
1	Review requirement	2			Samuel/Debo
2	System Design and Documentation	1			
Development (16 days)					
1	**Automation of Withdrawal Booklet Charges system**	12			Samuel
Integration/Review (4 days)					
1	Test application	3			Samuel
2	Application Review	1			

VII. Software

1. **Development Tool**: .NET Framework 4.5, C# programming language, Oracle Database
2. **Web Application:** Yes

VIII. Hardware

Existing server for hosting .Net Applications Interactions