FUNCTIONAL SPECIFICATION

Project Code:	POS-01
Project Name:	Pizza Ordering System

Revision History

Version (x.yy)	Date of Revision	Description of Change	Reason for Change	Affected Sections	Approved By
1.00	19-Sep-2011	Initial Draft			
2.10	Sept-2013	Revision	Mapping with CPC Tool		
2.20	Nov-2013	Revision	Aligning with UCF		

Affected Groups

Development Engineering
Quality Assurance
XYZ Automation Ltd.

List of Reference Documents

Name	Version No.
1. Request For Proposal	1.1
2.	
3.	
4.	

Prepared by/Date **Reviewed by/Date** **Approved by/Date**

Table of Contents

1 Introduction

XYZ Automation Ltd focuses on automating various systems that have been working manually since years.

XYZ Automation Ltd recently planned to automate “Pizza Ordering System” - a standalone/Web application [Core Java Batches - Swing Application; J2EE Batches - Web Application] intended to be used by user for ordering of pizza items.

Scope and Overview:
The scope of the Pizza Ordering System (POS) will be to provide the functionality as described below. The system will be developed on a Windows operating system using Java/J2EE.

2 System Overview

The Pizza Ordering System should support basic functionalities (explained in section 2.1) for all below listed users.

- Administrator (A)
- Customer (C)

2.1 Authentication & Authorization

2.1.1 Authentication:

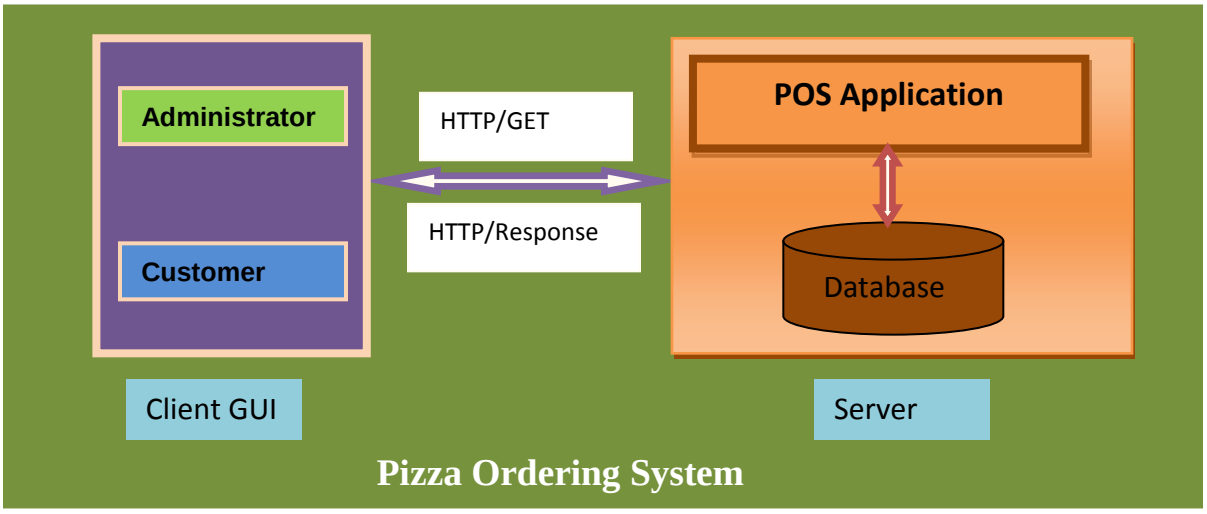
Any end-user should be authenticated using a unique login ID and password.

2.1.2 Authorization

The operations supported and allowed would be based on the user type. For example, Administrator has the rights to add/modify/delete pizza store and food item details.

2.2 Functional Flow

The functional flow of the messages across different application components is shown below.
Ex. - Web Application.



2.3 Environment

The system will be developed on a Windows XP machine using J2EE, JSP/HTML, and JDBC.

- Intel hardware machine (PC P4-2.26 GHz, 512 MB RAM, 40 GB HDD)
- Server – Apache Tomcat 6 or higher
- Database – Oracle 9i or higher
- JRE
- Eclipse IDE

3 Sub-system Details

The Pizza Ordering System (POS) is defined with two types of users (Administrator & Customer), wherein all users need to login successfully before performing any of their respective operations.

Find below (section 3.1 & 3.2) tables that provides functionality descriptions for each type of user / sub-system. Against each requirement, indicative data is listed in column 'Data to include'. Further, suggested to add/modify more details wherever required with an approval from customer/faculty.

3.1 Administrator

The administrator as a user is defined to perform below listed operations after successful login.

ID	Objects	Operations	Data to include	Remarks
AD-001 to AD-004	PizzaStore	Add Delete View Modify	StoreName, City, State, etc	Store ID should be auto generated
AD-005 to AD-008	FoodItem	Add Delete View Modify	Name, Price, etc	Food ID should be auto generated.
AD-009	Order	Modify	Order ID, Status etc	Order status can be Pending/Dispatched /Cancelled

3.2 Customer

The customer as a user is defined to perform below listed operations after successful login.

ID	Objects	Operations	Data to include	Remarks
US-002 to US-004	ShoppingCart	Add View Modify/ Delete	Food ID, Food Name, Unit Price, Quantity, Total Price etc	Use collection of FoodItem objects for the shopping cart
US-005	Order	Confirmation	Shipping details and credit card details	Use multiple steps to take shipment and credit card details from the user. Order ID should be auto generated.
US-006	Order	Cancellation	Order ID	Payment should be reverted.

[Swing Application - Core Java]

* US-003 : Allow user to view food items from multiple stores simultaneously.
  - Hint: Use multithreading.
* US-005 : Allow user to generate order details in HTML format.

[Web Application - J2EE]

* US-005 : Use | Create Web services for Payment process.
* US-005 : Allow user to generate order details in PDF format.

NOTE:

* Total Price should be calculated and display while taking credit card details from the user.
* Admin should be able to see all orders and change the status accordingly.
* On deleting Food Item, all affecting orders should be cancelled and the user should be intimated by phone/email.
* The payment should be reverted on order cancellation.

3.3 Login | Logout

[Swing Application - Core Java]

- Use System properties to enable the application to Startup with default/last user details for login.
- Enable the application to run from command prompt with user credentials.

[Web Application - J2EE]

- Implement Session tracking for all logged in users before allowing access to application features. Anonymous users should be checked, unless explicitly mentioned.

4 Data Organization

This section explains the data storage requirements of the Pizza Ordering System and **indicative data description** along with suggested table (database) structure. The following section explains few of the tables (fields) with description. However in similar approach need to be considered for all other tables.

4.1 Table: ShipmentDetails

The table contains information related to shipment details.

Field Name	Description
OrderID	Order ID of the order, auto-generated after successful payment
Name	Customer's Name
Address	Address of the Customer
City	City Name
State	State Name
PIN	Pin Code of the address
MobileNumber	10 digit contact Number
Landline	6-10 digit Landline
Emailid	Email ID of the customer
OrderStatus	Status of the order

4.2 Table: UserCredentials

The table contains Authentication Information for Administrator and Customer

Field Name	Description
UserType	Administrator and Customer
UserID	User Identification, corresponding to UserProfile table
Password	Password
LoginStatus	Login status of the user

4.3 Table: CreditCardDetails

This table contains information related to the credit cards of the users used for making payments.

Field Name	Description
CreditCardNumber	Credit Card Number of the user
ValidFrom	Validity of Credit Card from
ValidTo	Validity of Credit Card to
Balance	Existing Credit Limit of the user

5 Assumptions

- User Interface: The type of client interface (front-end) to be supported– GUI based/Web based
- The scope of the application is limited to only one country.
- Information about State, Place and Locality are already stored in database.
- All food items are available across stores with sufficient quantity.

- The user can purchase food items from store of his choice and can ask for shipment to any address

6 General Expectations

- The server should be a concurrent server servicing multiple clients
- Database can be implemented using Oracle 9i or above
- To begin with, the application should support at least 1 admin and 2 customers.
- Compilation and Build should be done using Eclipse IDE
- Source-code and all documents must be maintained (checked-in) in configuration management system (subversion)
- Wipro's coding standards (for Java) should be followed,
- Deliverables should include compiled and tested source code, Unit Test Code (Using WiproUT), WiproStyle report and System test-plan / report documents.

NOTE:

1. **Validation of user Data**[1]
   - ✓ Struts 2 validation via XML or annotations or Spring MVC using JSR-303 annotations
   - ✓ AJAX validation without forcing the page to reload (Wherever applicable)
   - ✓ JavaScript validation (if necessary)
   - ✓ In case of Swing applications, use 'ClassInputVerifier' for validation
2. **UI Design –(for Web Application) Use DIV/CSS to control the style and layout**
3. **Create at least one SQL DML-statement inside PL/SQL blocks**

7 Acceptance Criteria

All P1 requirements have to be mandatorily implemented

8 Traceability to Requirements

Appropriate requirements from RS and FS are mapped here.

Document Reference ID & Description: (Doc ID from which this document is derived)		
Sl. No.	**Reference document: RS Requirement/Feature (Section ID/Name)**	**Current document: FS Location (Section ID/Name)**
1.		
2.		

9 Acronyms and Glossary

Acronym and glossary for this document mentioned in the below table.

Abbreviation	**Remark**
POS	Pizza Ordering System
RS	Requirement Specification
FS	Functional Specification

[1] Validations should be performed at all levels of application appropriately.