La Vida Cafe - Casual Dining

SRS Document

La Vida Cafe   **3**

Disclaimer

This Software Requirements Specification document is a guideline. The document details all the high level requirements. The document should be used as a guideline by the students to design the Solution Architecture for the project. The document also describes the broad scope of the project and high level logical object model. But while developing the solution if the developer has a valid point to add more details being within the scope specified then it can be accommodated after consultation.

La Vida Cafe

Introduction

The purpose of this document is to define scope and requirements of a Casual dining outlets of a leading Mexican food chain in entertainment center - "La Vida Cafe". The customer experience should be seamless is the key objective of this automation.

This document should be used by the development team to architect the solution the project.

Management Summary

It is vital for casual dining restaurants to deliver prompt and delightful service to their customers. There are various models of operations in casual dining that a food chain can adopt like cafes, pubs, bars, fast food restaurants and drive-thru. Challenges that are constantly required to be handled include Deals, Promotions Mix and Match, Item discounts etc. The complexity of menu with variants, Fast Cash handling and service.

The proposed solution will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

Key Assumptions

1.   The system will does not include payment processing.

2.   The menu creation is not part of this project. They will entered or uploaded as a master

3.   For the purpose of project the events in the system are simplified. An actual implementation for casual dining will be fairly complex and will have various industry practices incorporated like multiple delivery models, special day flyers printing etc.

4.   The prices of items are assumed as inclusive of taxes to avoid computations.

High Level Architecture

La Vida Cafe high level architecture is illustrated through the context diagram shown below. It will have following categories of users:

1.   Administrator

2.   Point of Sales (POS) Agent

3.   Kitchen Staff

4.   Manager
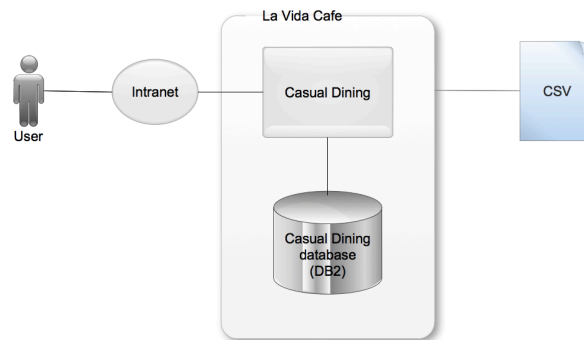
La Vida Cafe Context Diagram

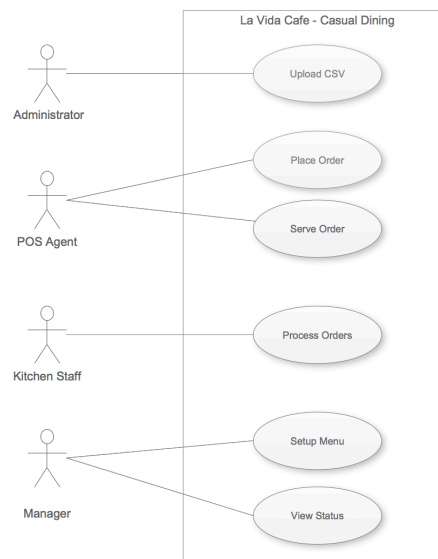|	
La Vida Cafe	Web application on internal servers for the quick order selection and processing, status update and serving and closing the order. Setup Menu for various catering options available in the Cafe.
La Vida Cafe Database	Designed to store menu with options, customer orders
CSV	Masters such as Menu items, Options, Menu Category

Functional Requirements

The high level functional requirements for the La Vida Cafe - Casual Dining are outlined in the Use Case diagram described in this section.

La Vida Cafe will provide a secure user-id/password based secured login mechanism to access its services. The details of this are not outlined here. The development team is expected to create these keeping in mind the general practices followed by the web applications. Login will be a prerequisite to use Casual Dining application. Internal users will be provided user id/ password pair separately.

Once user logs in, they get to see menu options coming from the use case of the role player.

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system.

Use Case Diagram

Use Cases

Upload Masters

Use Case Element	Description
Number	UC.01
Application	Masters in Casual Dining are uploaded using CSV Format
	Menu Type will have Menu Type id, Menu Type (E.g. Regular, Breakfast Catering, Lunch Catering, Event Catering, Kids birthday Catering)
	Menu Sections will have Menu Section id, Menu Section (Eg. Appetizers, Enchiladas, Tacos, Burritos, Soups and Salads, Desserts)
	Kitchen Sections master will have columns such as Kitchen Sections id, Kitchen Section Name
	Item master will have Item Code, Item Name, Description, Classification (Red/ Green), Price, Kitchen Section id, Menu Section id
Use Case Name	Upload Masters
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **Upload Masters** menu item on the landing page
Basic Flow	System prompts for the file name to be uploaded. Standard file upload dialog is presented to select a file from the local system.
	• The selected file data is uploaded in the related tables; if an existing record is encountered, the old details are replaced with the new details.

Use Case Element	Description
Alternate Flow	• In event of incorrect CSV format, system gives an error and NO data is uploaded. • Operation is cancelled
Output	System displays the number of records uploaded. It also highlights the number of records updated (i.e. already existing ones being replaced)

Place Order

Use Case Element	Description
Number	UC.02
Application	Order is taken from the customer and entered in the system
Use Case Name	Place Order
Primary Actor	POS Agent
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **Pace Order** link on the landing page
Basic Flow	The interface design is touch based. For this project purpose the user needs to click on the buttons or icons to navigate through the functionality. • The system displays a form with following fields for the user to select • Menu Type List - user selects one option • Section Names are displayed for the Menu type selected in a grid like structure. • The user clicks on e.g. menu section as Appetizer, the list of appetizers are displayed in the rows and columns below. • The user selects the items and clicks on the qty selector displayed on the left side of the screen. The qty selector appears like a bold digital calculator. • The top left corner of the screen displays the items and qty selected with the total amount row wise and a total of order value. On 'Transfer to Kitchen' the system saves order in the system with automatic order number and menu items as line items having order number as their reference. The ordered items are internally assigned kitchen section ids from the master and the same is visible as a queue to be prepared for the kitchen sections.
Alternate Flow	On Cancel, there is no impact on database
Output	None

Process Order

Use Case Element	Description
Number	UC.03
Application	The kitchen sections process orders in the queue
Use Case Name	Process Order
Primary Actor	Kitchen staff
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the **Section Queue** link on the landing page.
Basic Flow	• The system displays the list of items with order numbers in the priority queue based on time of order logging. • The ordered items displayed have three buttons Red, Amber and Green • The default button Red is highlighted for items that are yet to be processed. • The user clicks on the item that is picked up for processing, user can click on multiple items to begin prepare stage. The Amber button lights up. • On completion of packing and pushing in the serving aisle, the user clicks on the serviced status that highlights the Green button. • Service queue keeps getting refreshed for any new items falling in the queue and any existing queue items being serviced as Green.
Alternate Flow	None
Output	

Service Order

Use Case Element	Description
Number	UC.04
Application	The POS agents service order to the customer based on the completion status of each items that is ordered.
Use Case Name	Service Order
Primary Actor	POS Agent
Secondary Actor	None
Pre-condition	User is an authorized role in the system
Trigger	The user clicks on the **Service Order** link on the landing page.
Basic Flow	A list of pending orders to be serviced are displayed with items ordered and their kitchen status. This does not include status on beverages. The default list displays the order number, number of items, kitchen items count, ready count. On click on any one of them will display the item wise status, Red, Amber, Green. Once user services the order for all the items, the order is marked as completed. it moves out of the service queue.

Use Case Element	Description
Alternate Flow	Pressing Cancel abandons operation, no database gets affected
Output	None

Setup Menu

Use Case Element	Description
Number	UC.05
Application	Menu is setup for various types of offering that the cafe offers.
Use Case Name	Setup Menu
Primary Actor	Manager
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the **Setup Menu** link on the landing page.
Basic Flow	• The system displays Menu type list for selecting one of the type. • The system displays the menu section choices for selecting one of the section. • The system displays the list of items that can be selected to fall under the menu section. • The user enters the number for display of menu section such as '1 or 2 or 3' • Each item will have a serving size eg. 1, 2, 5, 10 • The menu type is setup for various menu sections. • The user clicks on preview, the system displays the menu card in a print format in the order of various menu sections defined. • Click on Submit will save the menu for menu type selected. note: a separate option to print menu as per the menu type can be given by the developer of this project.
Alternate Flow	Pressing Cancel abandons operation, no database gets affected.
Output	None

An update of status or order in WIP by kitchen sections is not included here, the developer of this project may like to add the use case with additional role player kitchen staff. On completion of WIP, the items are marked as 'Prepared', the Server can view the status and can serve the items in order of priority assigned.

View Status

Use Case Element	Description
Number	UC.06
Application	The Manager views the status of order being processed
Use Case Name	View Status
Primary Actor	Manager
Secondary Actor	None

Use Case Element	Description
Pre-condition	None
Trigger	The user clicks on the **View Status** link on the landing page.
Basic Flow	• The data for the following is counted/summed and displayed. • Total Orders = Count of orders received in the current shift • Total Served = Count of Orders in Served State • Total WIP = Count of orders in WIP • Total Order Value = Total of orders booked • Top Menu Items = Count of items Served in the descending order.
Alternate Flow	None
Output	None

Logical Object Model

A high level logical object model of the system is shown below. During technical design it will be transformed into a physical model covering all system entities. Such a diagram will include their relationship and its cardinality.
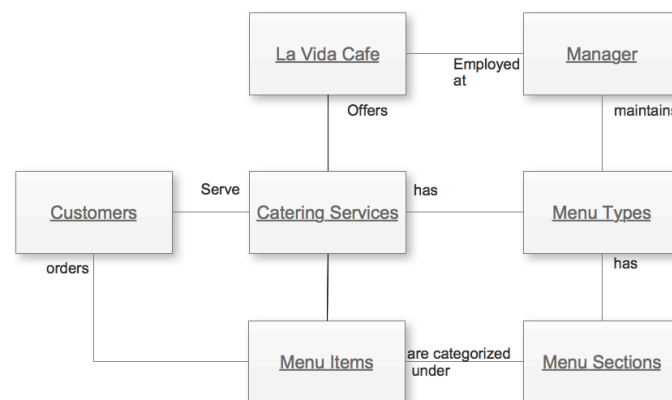
Logical Object Model

1. La Vida Cafe Casual Dining is a web based system to service customers for various catering services offered by the cafe.

2.   Catering services are defined in the Menu type when Regular menu is replaced by the event based menu is activated. Such menu types are Breakfast Catering, Lunch Catering, Dinner Catering, Birthday Catering etc.

3. Each Menu type has various sections such as Appetizers, Soups and Salads, Entrees, Burritos, Enchiladas, Desserts etc.

4. Menu items are defined as item codes and item names with price. Each item also has the menu section id stored along with it as it will be categorized for display and print purposes under this very menu section id.

5. Menu is setup by the manager to store the selected menu sections and items under it.

Sample Menu with Sections [Print option for menu could use this format]

Database Design Guidelines

This involves the transformation of the use cases, state diagrams, and logical object model into detailed and optimized physical database table designs.

Typically persistent classes will map to table(s) with their attributes as columns of the table. In some cases a high level object may map in to a master-child table. Invoice is one such example where it maps in to "invoice_header" and "invoice_line_item" table.

Associations between two persistent objects are realized as foreign keys to the associated objects. A foreign key is a column in one table that contains the primary key value of the associated object.

Similarly, a standard technique in relational modeling is to use an intersection entity to represent many-to-many associations. Following is a broad checklist for physical database design:

1. Database must be properly normalized except those instances where de-normalization help improves performance. This option must be used with special care.

2. All persistent classes that use the database for persistency must map to database structures.

3. Many-to-many relationships must have an intersecting table.

4. Primary keys should be defined for each table, unless there is a performance reason not to define a primary key.

5. Indexes should be defined to optimize access.

6.   Data and referential integrity constraints should be defined.

Testing Approach

Quality of the software can be achieved with basic hygiene and consistency followed during design and development of User Interface(UI), Navigation, Validations as per the business process requirement.

To ensure the project delivers acceptable quality to the customer, its important to create a checklist of the conventions to be followed across. Common checks as below are for your reference during design and development:

Common Checks	Validation Type
Page Title is valid for the feature being provided on the page	UI
Order of the Data Entry Fields is logical as per the functionality being provided by the feature	UI
Order of the Display only Fields makes viewing and understanding easy for the user	UI
Spellings and Correctness of Label for the Data Entry and Display fields	UI
The labels are not wrapping onto another row thereby adding a blank row on the page	UI
The fields with drop down are displayed in single row instead of drop down coming on the next row	UI
Data Entry field basic validations are working i.e Text field /Numbers / Dates allow data for their type only	Functional
The dates are following a standard format dd/mmm/yyyy on all forms	UI
The color scheme of all forms i.e headers labels , alerts, entry fields are uniform throughout the application	UI
The action buttons for a New Data Entry Form are uniform for all forms that is allowing data entry	UI
The action buttons are performing the desired action e.g. "submit" is creating a new record if there are no errors and recording all the input fields, whereas 'cancel' is not creating a new record in the database	Functional
The links provided on the forms are opening correctly.	Functional
The data feed mechanism for Read and Write files is generating a log with count of entries.	Navigation

Suggested Technical Reading

The project is aimed at making the student understand concepts of Design and Development using IBM Rational tools, Web Sphere Application Server and DB2 Database. The following reading reference is easy to understand and should be read to get a clear understanding of capabilities of the tools and how you would leverage them to execute a project.

Technical Reference	URL to access
RAD - Tackling challenges of software development with Rational Application Developer for WebSphere Software	http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html
IBM Education Assistant - Rational Application Developer 7.5	http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html

Technical Reference	URL to access
Rational Technical Library	http://www.ibm.com/developerworks/rational/library/