



## **Shop & Influence**

SRS Document

<b>Shop &amp; Influence</b>	<b>3</b>
<b>Introduction</b>	<b>3</b>
<b>Management Summary</b>	<b>3</b>
<b>Assumptions</b>	<b>3</b>
<b>High Level Architecture</b>	<b>4</b>
<b>Functional Requirements</b>	<b>5</b>
Use Case Diagrams	<b>5</b>
Use Cases	<b>5</b>
<i>Upload Masters</i>	<b>5</b>
<i>Register Shopper</i>	<b>6</b>
<i>Billing</i>	<b>7</b>
<i>View Brand Performance</i>	<b>7</b>
<i>View Customer Buying Patterns</i>	<b>8</b>
<b>Logical Object Model</b>	<b>9</b>
<b>Database Design Guidelines</b>	<b>10</b>
<b>Testing Approach</b>	<b>10</b>
<b>Suggested Technical Reading</b>	<b>11</b>

## 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.

# Shop & Influence

## Introduction

The purpose of this document is to define scope and requirements of Customer program of a retail chain **Best-Buy**. The Shop & Influence system is web based, functional at their Retail outlets. The customers are registered on the system and their buying patterns are recorded for further analysis.

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

## Management Summary

A retail business house works on increasing its sales and engaging customers on long term basis. All the transactions done by the user at the time of billing are recorded in the Shop & Influence system. The proposed system Shop & Influence will deliver:

1. Customers are Registered and a card is issued to the customer.
2. All customer transactions are recorded along with the card number.
3. The System generates analysis of Customer buying patterns.
4. Brand performance is monitored with slow and fast moving parameters.

Shop & Influence will be a web-based system and will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

## Assumptions

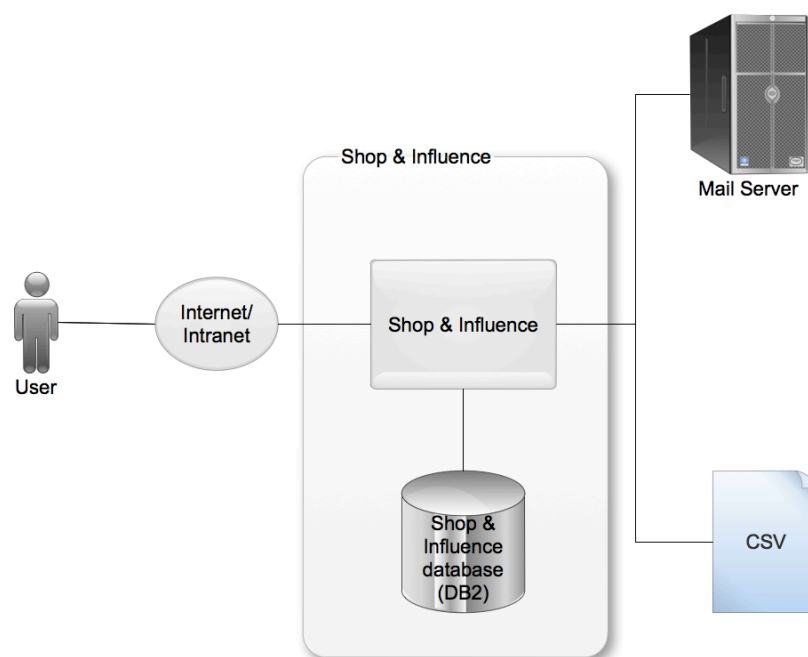
1. A membership card with a unique id is issued to the customer for showing at the time of billing. For project's simplicity, the shopper card number is being used to access shopper's information and post the purchase related details to be accessed later for analysis.
2. The billing system is not taken as a separate entity. The Shop & Influence system will have an input screen to record the Shopper id (Shopper Card Number ), Bill date ( today's date), Purchase Qty, Purchase value at the time of billing.

3. The system assumes stocks are in place for this products being purchased, there is no validation on stocks at the time of billing for the scope of this project.

## High Level Architecture

Shop & Influence's high level architecture is illustrated through the context diagram shown in figure below. The users of the system are

1. Retail Billing
2. Retail Management
3. Administrator



*Shop & Influence Context Diagram*

Shop & Influence

The proposed web-based application lets the Billing activity access the shopper record through the shopper card number. Application tracks the buying pattern, brands, season etc for creating offers

Shop & Influence Database

The shopper record can be accessed at any store location of Best-Buy. This will hold all the Shop & Influence data including billing purchase, Purchase history, shoppers profile.

Email Server

Mail server that will be used for notifications to the shoppers

## Functional Requirements

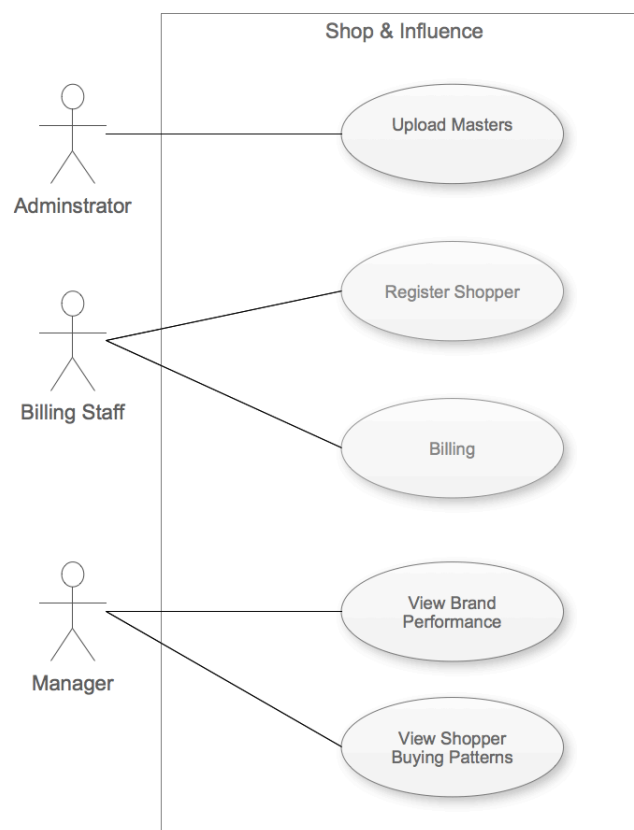
The high level functional requirement for the Shop & Influence system is represented in the Use Case diagram shown below. The remaining sections in the document describe the major use cases.

Shop & Influence 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.

The application opens with a home page and has a different set of options for different profile of users. Role based access to be implemented.

## 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

Use Case Element	Description
Application	<p>Masters are uploaded via CSV files for the following</p> <p>Brand master will have Brand id, Brand label</p> <p>Category master will have Category id, Category Name [e.g. Lighting, Home Decor, Kitchen ware, Soft Furnishings, Groceries etc.)</p> <p>Product Master will have Product id, Product name, Brand id, Category id</p> <p>Users master will have user id and username</p>
Use Case Name	Upload Masters
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the <b>Upload Masters</b> menu item on the landing page
Basic Flow	<p>System prompts for the file name to be uploaded. Standard file upload dialog is presented to select a file from the local system.</p> <ul style="list-style-type: none"> <li>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.</li> </ul>
Alternate Flow	<ul style="list-style-type: none"> <li>In event of incorrect CSV format, system gives an error and NO data is uploaded.</li> <li>Operation is cancelled</li> </ul>
Output	System displays the number of records uploaded. It also highlights the number of records updated (i.e. already existing ones being replaced)

### Register Shopper

Use Case Element	Description
Number	UC.02
Application	The customers of the Best-Buy keep frequenting the store often. They register themselves to get invited for brand wise sales. A simple process of registering at the time of checkout at billing counter
Use Case Name	Register Shopper
Primary Actor	Billing Staff
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the <b>Register Shopper</b> menu item on the application home page

Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> <li>System opens a form to enter the following information <ul style="list-style-type: none"> <li>First Name</li> <li>Last Name</li> <li>e-mail id</li> </ul> </li> <li>All the inputs are mandatory.</li> <li>Click on Enroll to add the shopper to the database. The entry is created along with the creation date.</li> <li>A unique id is created at the time of enrolling and a card is issued.</li> </ul>
Alternate Flow	Cancel will abandon the operation, no data is saved.
Output	E-mail is sent to the customer confirming enrollment.

### Billing

Use Case Element	Description
Number	UC.03
Application	A normal checkout is to pay by cash or credit card and get the billing done. Here shopper card number is used to access the shopper record and add shopping details for the current purchase.
Use Case Name	Billing
Primary Actor	Billing Staff
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the <b>Billing</b> menu item on the home page.
Basic Flow	<ul style="list-style-type: none"> <li>The Shop &amp; Influence system captures the Date, Products, Qty, Value, purchase value, shopper card number and the current date for maintaining history.</li> <li>Click Save to finish the transaction.</li> <li>System adds brand, products, qty, value in the shopper's record.</li> </ul> <p>There are no cash or credit card payment to be tracked in the system]</p>
Alternate Flow	Operation is cancelled, no database operation
Output	None

### View Brand Performance

Use Case Element	Description
Number	UC.04
Application	The retail management uses billing data effectively to come up with brand performance on a continuous basis. This keeps the customers in flow to the retail outlet and feel satisfied with the brands being marketed by the store.



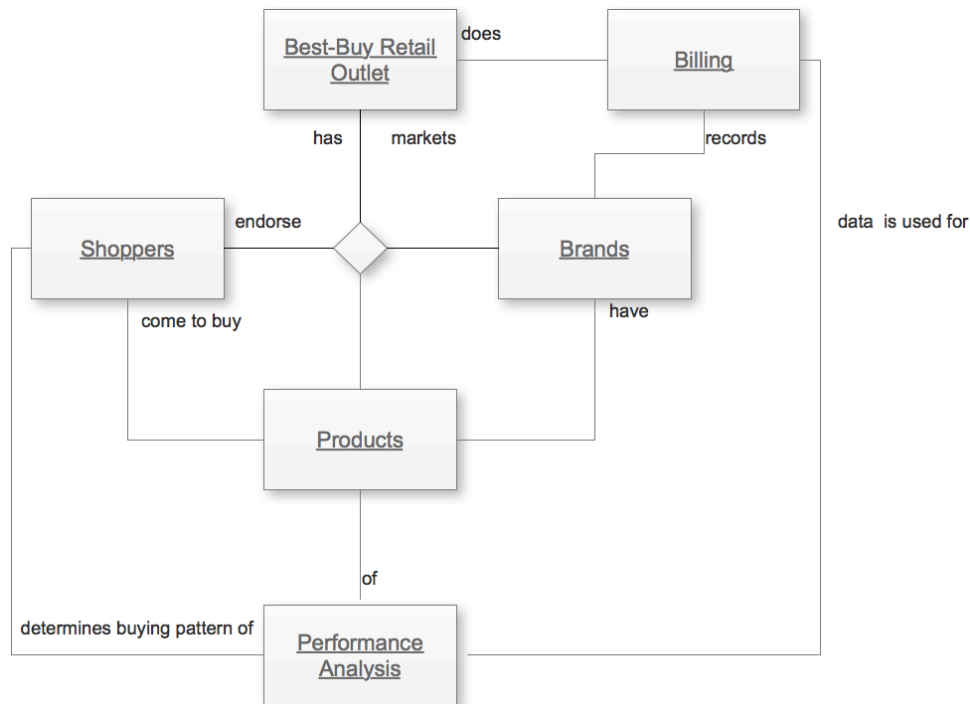
Use Case Element	Description
Use Case Name	View brand performance
Primary Actor	Management
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the <b>Brand Performance</b> menu item on the Home page
Basic Flow	<ul style="list-style-type: none"> <li>The dashboard for Brand wise sales analysis is displayed with following data.</li> <li>Highest Selling brands Value and No. of transactions</li> <li>Lowest Selling brands - Value and No. of transactions</li> <li>Average sales per brand - Value and No. of transactions</li> <li>Share of wallet - Amount spent on brand / total amount spent on category</li> </ul>
Alternate Flow	<ul style="list-style-type: none"> <li>None</li> </ul>
Output	None

### View Customer Buying Patterns

Use Case Element	Description
Number	UC.05
Application	Management keeps close eye on what are the buying patterns of their key customers and low customers. How do they improve it is always a focus of the Buying pattern report.
Use Case Name	View Customer Buying Patterns
Primary Actor	Management
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the <b>Buying Pattern of a Shopper</b> menu item on the home page.
Basic Flow	<ul style="list-style-type: none"> <li>A dashboard of buying patterns is displayed with following information in various sections</li> <li>Sales figures distributed month wise in an order of low to high</li> <li>Sales figures, total number of transactions, new shoppers added in the last month</li> <li>Average number of purchases per shopper</li> <li>Purchase frequency segments [High value shoppers break up ]</li> <li>Purchase recency [ High value shoppers - When was the last time shopper bought from the store]</li> <li>Purchase pattern change from High spender to Looking somewhere else [last three months high to low sales shopper wise]</li> </ul>
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. Best-Buy Retail markets various brands in household supplies (including groceries, fresh produce, frozen)
2. Shoppers visit Best-Buy for various household needs getting filled with popular and niche brands of products marketed by Best-Buy.
3. The products belong to branded labels as well as home brand Best-Buy.
4. The shoppers are issued a Best-Buy card that creates a shopper profile in the system.
5. All the purchases are linked to the Best-Buy card during billing process.
6. The billing data is used by Performance Analysis tool of the system
7. The Performance analysis of Shoppers buying patterns is determined as wells as the brand wise performance of fast and slow moving brands.

## 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 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/mm/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

Common Checks	Validation Type
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, WebSphere 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	<a href="http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html">http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html</a>
IBM Education Assistant - Rational Application Developer 7.5	<a href="http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html">http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html</a>
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	<a href="http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html">http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html</a>
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	<a href="http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html">http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html</a>
Rational Technical Library	<a href="http://www.ibm.com/developerworks/rational/library/">http://www.ibm.com/developerworks/rational/library/</a>