



m-SFA

SRS Document

m-SFA - Sales Force Automation	3
Introduction	3
Management Summary	3
Benefits of SFA	3
Key Assumptions	4
High Level Architecture	4
Functional Requirements	5
State Diagrams	5
Use Case Diagrams	5
Use Cases	6
<i>Maintain Masters</i>	6
<i>View Dealer Snapshot</i>	7
<i>View Products</i>	8
<i>Manage My Schedule</i>	9
Logical Object Model	10
Database Design Guidelines	11
Testing Approach	12
Suggested Technical Reading	12

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.

m-SFA - Sales Force Automation

Introduction

The purpose of this document is to define scope and requirements of automation for a mobile Sales Force who are servicing the customer from field itself. Using their tablet devices with internet connectivity they will access this system, pull or push information about the customer needs into the system.

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

Management Summary

The group's Decorative Business Unit – India (DBU) is contributing to around 75% of the business. DBU has 75 sales offices across the country and a sales force of around 500. The sales force services some 19000 dealers. The challenges of communicating the current pricing, schemes with such vast geographical coverage are phenomenal. The company decided to empower its sales force by giving them devices with internet connectivity to access the m-SFA application.

The SFA objective is to improve effectiveness and efficiency of the sales force leading to improved customer servicing. Access to information while being on the field brings independence in operations and also fosters higher quality of interaction with dealers. The key drivers for the SFA are

1. The Field officer can access dealer related information from the core application.
2. Information on products, latest buzz, activities, status updates are readily available, Its like an online office.
3. No pulling manual files to look at shade card or prices etc.. Data in hand is better than data in bag.
4. Now the Field Officer can focus on product level data along with branch/dealer level data
5. Field officer is independent, eliminates the need to call office time and again for asking something or other. Relying on too many people reduces
6. No back logs to update regarding field visits, dealer complaints are immediately entered in the application. a notepad kind of device with connectivity saves me a lot of time and quality of interaction improves.

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

Benefits of SFA

1. Higher responsiveness and empowerment Sales information at fingertips
2. Capture and track status of Dealer issues/complaints Credit Note/L&B CN initiations

3. Improved Efficiency Non Value adding activities to reduce Dependency on unit for reports, filing visit reports, etc to reduce
4. Quality Dealer Interaction Better tracking of complaints issues TSO to close all issues Information available online

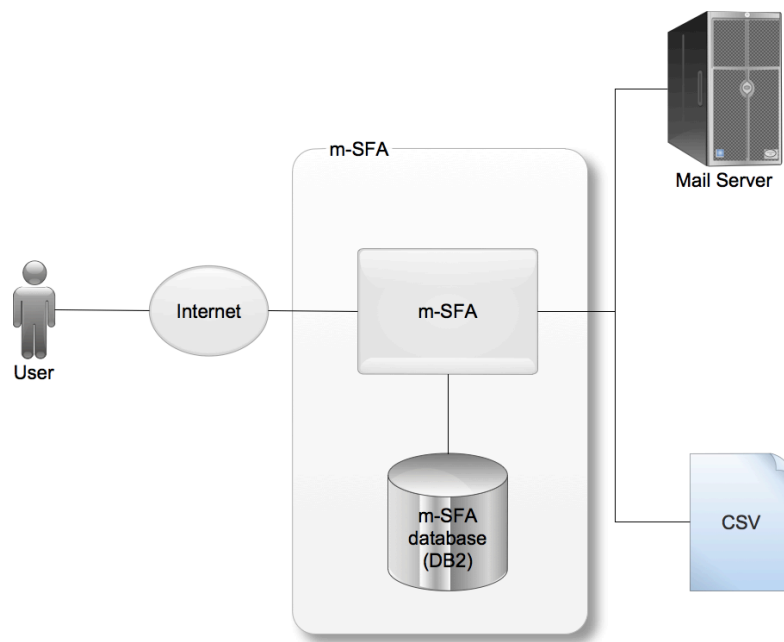
Key Assumptions

1. The project owner understands the challenges of servicing customers spread out in various territories. The concepts of dealer network remains same for any industry. The challenges are also similar.
2. Building this kind of application shall make the project owner aware of pain areas of having a large workforce operating out on their own. The need for information is critical to service a widespread customer network.

High Level Architecture

m-SFA high level architecture is illustrated through the context diagram shown below. It will have following categories of users:

1. Field Officer/Executive
2. Administrator



m-SFA Context Diagram

m-SFA

The application services access to content like products, campaigns, status of complaints. The transactions such as orders processing status, meeting schedules are managed in the application.

m-SFA Database

This will hold data for m-SFA on a central server.

CSV

Product Category, Products, Dealers, Field officer data are uploaded using CSV.

Functional Requirements

The high level functional requirements for the m-SFA are outlined in the Use Case diagram described in this section.

m-SFA 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 m-SFA. Internal users will be provided user id/password pair separately.

Once Field Executive logs in, it displays three options, 1) My Schedule, 2) Dealer Snapshot and 3) Products.

Click on My Schedule displays data in various sections, "Today's Schedule", "Pending Visits"

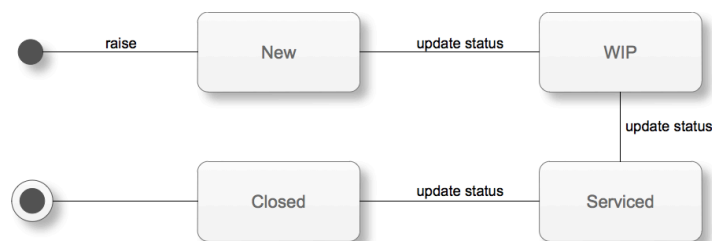
Click on Dealer Snapshot displays a Search box with a pick list of Dealers. Selecting one of the dealers will display information for only the dealer selected.

Click on Product info displays the Product Categories as links. Selecting a Category displays the product details and a product data sheet as a clickable option.

Please refer to use cases for the detail on each of these pages.

State Diagrams

The following diagram shows workflow of enquiry or issues raised by the dealers via Field Executive /Officer.

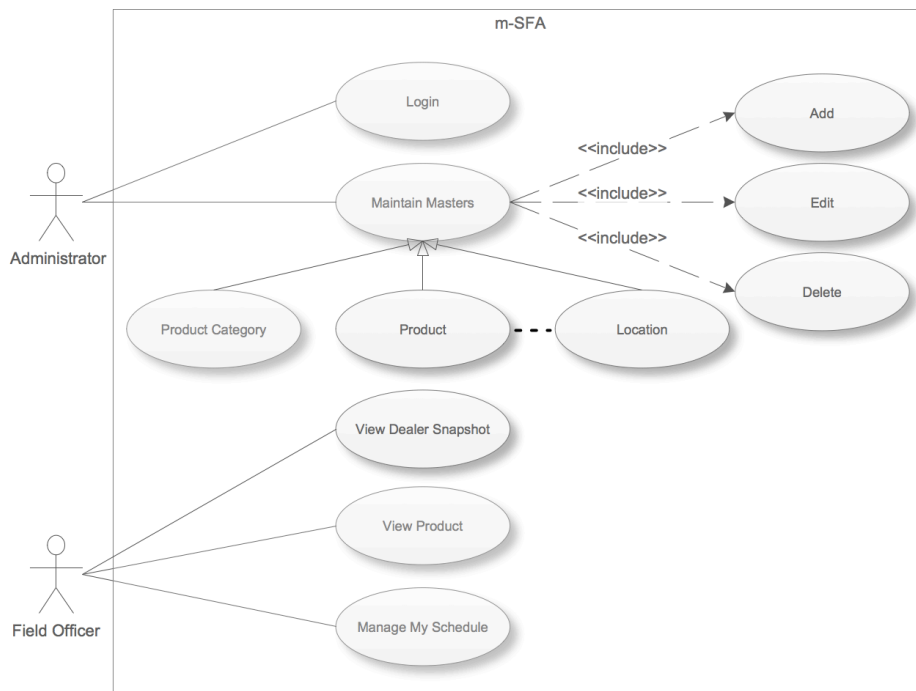


Dealer Request(s) State Diagram

1. The field executive adds an Enquiry, Order or a Complaint in the system as raised by the Dealer. On entry the state of the request is 'New'.
2. Once the request is communicated to the backend team (offline process) the status is changed to 'Work In Progress'.
3. All requests that have been serviced by the backend team are moved to Serviced state.
4. Once the dealer confirms their request has been serviced, the request is moved into closed state.
5. Please note that there is no document flow with this workflow as its being used only for status updates for the type of requests raised by Dealer.

Use Case Diagrams

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



Use Case Diagram

Use Cases

Maintain Masters

Use Case Element	Description
Number	UC.01
Application	<p>Master maintenance in terms of basic operations viz. add, edit, delete and view. All master maintenance i.e. Resource Master, Accessories Master, Services Master are child use cases of this Use Case.</p> <ul style="list-style-type: none"> Product Categories master contains two columns 1) Product Category and 2) Product Name. Some of the product categories are Ancillaries, Decorative Paints, Automotive, Industrial Product Master contains columns such as 'Product Category, Product Name, Product brief. Product File location Location Master contains list of locations where the dealers outlets are operating from . The master will comprise of location id and location name. For the purpose of simplicity, City Names shall be added in this master. Columns to be maintained are Location id, Location name Dealer Master contains Dealer Code, Dealer Name Outlets Master contains Outlet id, Outlet Name, Add1, Add2, Location id, Dealer Code Field Executives Master contains Employee Code, Employee Name, Location
Use Case Name	Maintain Masters

Use Case Element	Description
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the Maintain Masters menu item on the admin interface page
Basic Flow	<ul style="list-style-type: none"> System presents a list of masters that can be maintained. Administrator selects the desired master. System displays a list view and links for <i>add</i>, <i>edit</i> and <i>delete</i>. <ol style="list-style-type: none"> In case add, a new master record data entry form is presented. The master record is saved on clicking the save button provided form clears all the data validations (if any). The list view is updated accordingly. In case of edit, from the list view user is prompted to select the desired record to edit, Selected record is opened for editing. The edited master record is updated on clicking the update button, provided form clears all the data validations (if any). In case of delete, from the list view user selects the check box(s) against each record. Selected records are deleted up on clicking the delete button. However, user is presented a confirmation dialog before deleting the records.
Alternate Flow	<ul style="list-style-type: none"> In event of any error, it is clearly displayed and user is asked to reenter data or perform operation again.
Output	System displays the details of the successful operation.

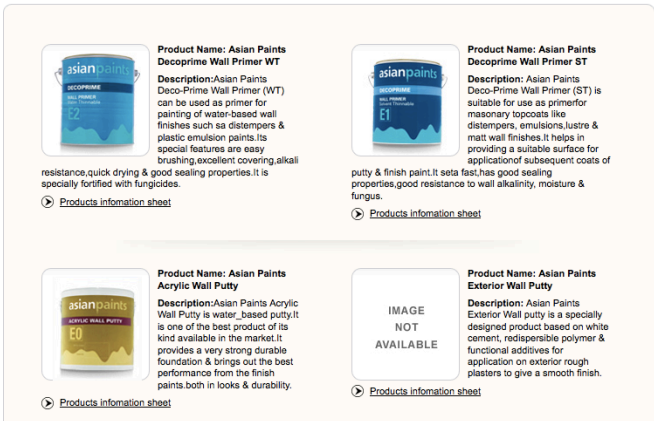
View Dealer Snapshot

Use Case Element	Description
Number	UC.02
Application	The Field Executive needs to access the dealer info on orders, enquiries, complaints by searching on dealer code
Use Case Name	View Dealer Information
Primary Actor	Field Executive
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the Dealer Snapshot link on the Landing page

Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> System displays the list of Dealers falling under Field Executive's location. User selects one dealer name from the list. The system pulls up the data for dealers as per the following sections. The [New] link in front of each section displays a row in the data grid to for user to enter the section data. A 'Save' button appears for every new entry that is being made in the system. The status of new entries in each of the section will be updated as 'New' Orders Status [New] <ul style="list-style-type: none"> Order Date, Product Category Name, Product Name, Qty, Status Complaints [New] <ul style="list-style-type: none"> Complaint Date, Product Category, Product Name, Description, Status Enquiries [New] <ul style="list-style-type: none"> Enquiry Date, Product Category, Description, Status Quarter Forecast & Actuals [New] <ul style="list-style-type: none"> Quarter 1/2/3/4, Year, Product Category, Product Name, Forecast Qty, Actual Qty (is arrived at as Sum total of the orders closed in the Quarters) Q1 = Jan-Feb-Mar Q2 = Apr-May-June Q3 = July-Aug-Sept Q4 = Oct-Nov-Dec Click on any of the existing entries displays the entry in an editable mode with a button for update. Please note the Status can be changed as per the workflow model defined in the workflow diagram. New -> Work In Progress(WIP) -> Completed -> Closed The entries in closed status will move out from this view.
Alternate Flow	None
Output	None

View Products

Use Case Element	Description
Number	UC.03
Application	The Field Executive accesses Product Information online as and when required by the dealer for some clarifications. This reduces the hassles of carrying and also expense of creating brochures and product sheets for the dealer.

Use Case Element	Description
Use Case Name	Product Information
Primary Actor	Field Executive
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the Products option on the Landing page
Basic Flow	<ul style="list-style-type: none"> System displays the list Product categories. The user selects the category and system displays a page of products similar to a layout given below:  <ul style="list-style-type: none"> User clicks on the Product sheet link and the system pulls up the PDF containing product details of the product selected. Click on any of the existing entries displays the entry in an editable mode with a button for update.
Alternate Flow	None
Output	None

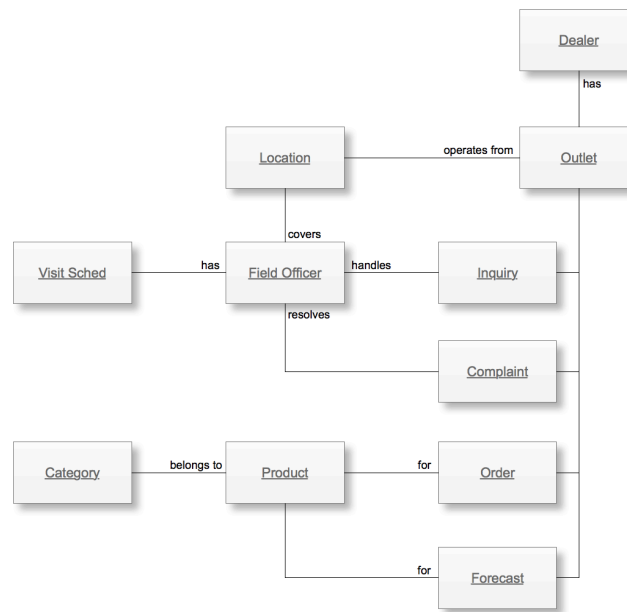
Manage My Schedule

Use Case Element	Description
Number	UC.04
Application	<p>The Field Executive gets their appointment scheduled for the day as per the visit frequency of 15 days to every dealer that the FE is managing. The entries in the schedule can be postponed by selecting another date or entries from the pending list can be moved to the today's schedule.</p> <p>The status of the meeting is updated basis which the system computes the next appointment date for the visit to a dealer.</p>
Use Case Name	Manage My Schedule
Primary Actor	Field Executive
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the My Schedule link on the Landing page

Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> System displays the list of appointments for the current date and the list of pending appointments side by side in two sections. Meetings Today <ul style="list-style-type: none"> Dealer Name, Count of Enquiries in 'New' Pending Appointments <ul style="list-style-type: none"> Dealer Name, Count of Enquiries in 'New' State The above data will be lists displayed side by side. The user should be able to move entry from one list to the other. In case the user clicks on any of the entry in the Meetings Today section, the system displays the following: <ul style="list-style-type: none"> Date picker for selecting the next date. Validation for selection is a previous date cannot be selected. Status : Done/Pending. The Done option will display time In and out input fields with a basic time format : HH:MM. The out time cannot be less than or equal to the In time data. Done updates the status of the entry as 'Completed' and it adds the 15 days to current date and adds a new entry for a meeting capturing, Employee id, Dealer Code, Date, Status as 'Open' <p>This interface is required to be user friendly with drag and drop features of pulling entries from one section to the other.</p>
Alternate Flow	None
Output	An altered view of the changes made are displayed after each change made by the user.

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. The company services the dealers as their customers. These dealers are spread across various locations.
2. The dealer operates from an Outlet in particular location.
3. The field officer/executives are assigned locations for servicing the dealers operating in those locations.
4. The dealers sell products falling under a particular category.
5. Dealer provides quarterly forecast of the sales for various products.
6. Dealer raises an order for product.
7. A field executive/officer schedules a visit every 2 weeks to the dealer location.
8. On his/her device, the field officer records the Order, Enquiry, Complaints while on visit to the dealer outlet.
9. The status of the dealer requests are updated by the Field officer as and when the information is received (offline)

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
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
Rational Technical Library	http://www.ibm.com/developerworks/rational/library/