Smile-Farm

SRS Document

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.

Smile Farm, an e-revolution for farmers

Introduction

The purpose of this document is to define scope and requirements of an agricultural initiative “Smile-Farm”. It is a vision of extending e-revolution to farmers and cooperatives through a three pronged strategy: a) development of knowledge inputs and e-services of relevance to rural India, b) evolve effective dissemination strategies to encourage active use of the facilities by farmers & cooperatives, c) the broad aim is to empower farmers and cooperatives through latest in information and communication technology

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

Management Summary

The Indian farmers are dependent on timely supply of the fertilizers for maintaining their soil productivity. Majority of the Indian farmers are very poor and illiterate which results in improper application of fertilizers. They lack in awareness and education and often depend on unscrupulous middle men for advice.

The objective is to take the latest information and knowledge inputs to farmers so that they can improve their productivity and improve their economic well being.

1. Register farmers into Smile-Farm. They can subscribe to the Query Services on various Categories of the Smile-Farm initiative.
2. Address issues on calls with farmers, document the conversations, respond to them using current knowledge artifacts available in the system.
3. The Subject Matter Experts on various Categories keep updating Tips and Advisories in the system.
4. Include Advisors Tips on the Categories to provide solutions when the farmer has a specific need.
5. The solutions provided are stored and are accessible using search on Category, Location

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

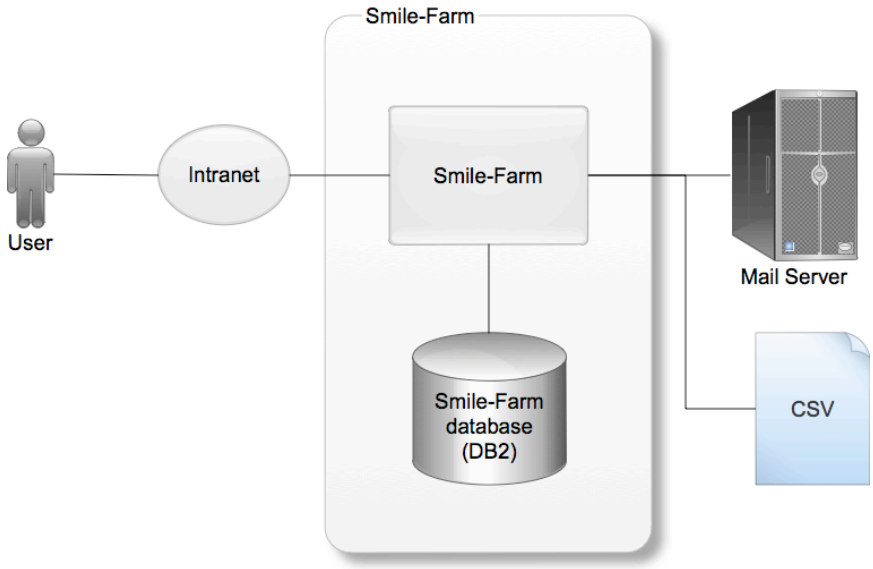
Key Assumptions

1. The system is ideally meant for farmers having mobiles for receiving SMS of the information they have subscribed to. For project scope, email id is being used for communicating.
2. The Project owner is familiar with Farming aids and initiatives taken by various organization for empowerment. They are eye openers from the technology point of view.

High Level Architecture

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

1. Content Manager
2. Subject Matter Expert
3. Administrator



Smile-Farm Context Diagram

Smile-Farm	Maintain Farmers profile by uploading their basic contact, geographical details along with crop preferences and animals if they have. Content Managers at the state level cater to information needs for farmers in their state. Maintain an Expert Panel as Subject Matter Experts for handling queries received from farmers Create a subscription based model for information services offered by the company.
Smile-Farm Database	Designed to store Farmers profile, geographical areas, Mapping of Marketing Exec and content managers to the geographies. Information categories, Experts are mapped to Categories. Services farmers subscription to queries on Information categories.
CSV	Masters like locations, Farmers, categories, SMEs are uploaded via CSV
Mail Server	All notifications are routed through the Mail Server

Functional Requirements

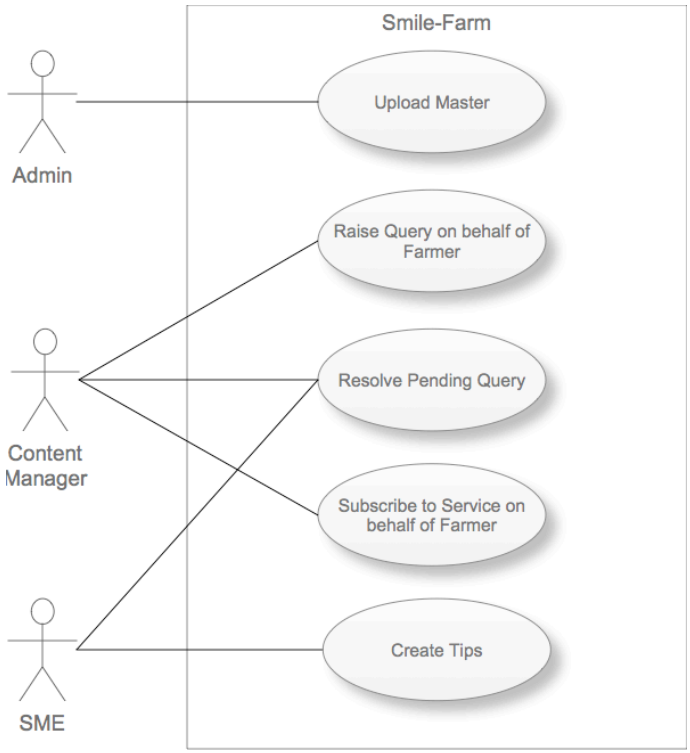
The high level functional requirements for the Smile-Farm are outlined in the Use Case diagram described in this section.

Smile-Farm 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 Smile-Farm. Internal users will be provided user id/password pair separately.

Once user logs in, s/he can view the dashboard of farmer queries with status, Latest Tips and Advisories added by SMEs are displayed. This menu options shall primarily come from the use case of the role player.

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system. The MiS use cases are not detailed here.



Use Case Diagram

Use Cases

Upload Masters

Use Case Element	Description
Number	UC.01

Use Case Element	Description
Application	Masters in Smile-Farm are uploaded using CSV Format State Master contains State id, State Name Location Master contains Location id, Location Name, State id Category Master contains Category id, Category Name (Though categories are hierarchical, for simplicity, its a flat structure for this project) SME Master contains SME Id, SME Name SME Category contains SME id, Category id (SMEs can have multiple categories expertise, thus a one to one mapping is required for each category id)
Use Case Name	Maintain Masters
Primary Actor	Content Manager
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.
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)

Raise Query

Use Case Element	Description
Number	UC.02
Application	Farmers call in the helpline and seek information on various information categories. Smile-Farm endeavor is to support farmers with contextual information so that they reap immediate benefits and are not lost in figuring out what to do. To make information being given relevant and crisp, the system uses the knowledge acquired over a period of time and has significant value in terms of being referenced in providing solutions to farmers.
Use Case Name	Raise Query on behalf of Farmer
Primary Actor	Content Manager
Secondary Actor	None
Pre-condition	Authorized user is logged in to the system
Trigger	User clicks on the **Raise Query** link on the landing page

Use Case Element	Description
Basic Flow	• The content manager types the customer mobile number from in the mobile to pull the customer data automatically. • If Data does not exist, Content managers enters the Mobile #, Name of the farmer, State (Picker), Location(Picker - filtered list for the State) • If existing in database then any open or unresolved query is displayed on the top clearly highlighting the status. • In addition last few resolved queries are also displayed. • The Query asked by the farmer is entered in the query box. • User selects the Category in which query falls. Eg. Seeds Sowing or Crop is infested. Leaves are falling etc. System automatically lists various relevant content on the basis of query categorization to help rapid resolution of query. **Queries Resolved** in the queried category are displayed as line items. The order of display is Query with maximum reference is displayed on the top and so on. **Tips on Category** Recently added tips and advisories on the category and farmer's state are displayed **Favorite in Category**, All Queries that have been bookmarked are displayed for reference. • It displays the Information on the basis of (a) recently added Tips, (b) most referenced for farmers state. • The reference count goes up by 1 for the information being included as referenced for the solution to customer. • View count goes up by 1 for the information being clicked and viewed • Most queries are resolved in a single call. In event of non-resolution in the same call, content manager can set up a callback time (based on the estimated time to resolve the query). • The system automatically sends notification to the content manager on the call back time. • Subscribe to services can also occur in the same query, a link for subscription is provided. A separate use case is mentioned for reference to the flow.
Alternate Flow	None
Output	Notification to content manager for resolving query

Resolve Pending Query

Use Case Element	Description
Number	UC.03
Application	Some Queries require Expert advise to resolve
Use Case Name	Resolve Pending Query
Primary Actor	SME
Secondary Actor	Content Manager
Pre-condition	None
Trigger	The user clicks on the **Resolve Query** link on the landing page
Basic Flow	• Any open or unresolved query is displayed on the top clearly highlighting the status for the logged in SME or Content Manager who created the query in the system. • The SME user reads the Query from customer and enters the solution to be communicated to customer. • SME Clicks on Communicate. • An email for the same is sent form the system. • The Query moves to Communicated State • Content Manager can access the query and the solution provided by SME • The customer is contacted for any assistance with solution. • The reference count goes up by 1 for the information being included as referenced for the solution to customer. • View count goes up by 1 for the information being clicked and viewed • The query is closed.
Alternate Flow	None
Output	Email to customer for solution provided by SME

Subscription to Services

Use Case Element	Description
Number	UC.04
Application	The farmers seek valuable information in the area of their interests, thus they subscribe to information on those domains to pushed to them via SMS or emails.
Use Case Name	Subscription to Services
Primary Actor	Content Manager
Secondary Actor	None
Pre-condition	User is an authorized role in the system
Trigger	The user clicks on the **Subscribe** link on the Query page.

Use Case Element	Description
Basic Flow	The system displays list of Information Categories to select from The selected entries are saved as subscription for the farmer whose query form is being worked upon.
Alternate Flow	Pressing Cancel abandons operation, no database gets affected
Output	None

Create TIPS

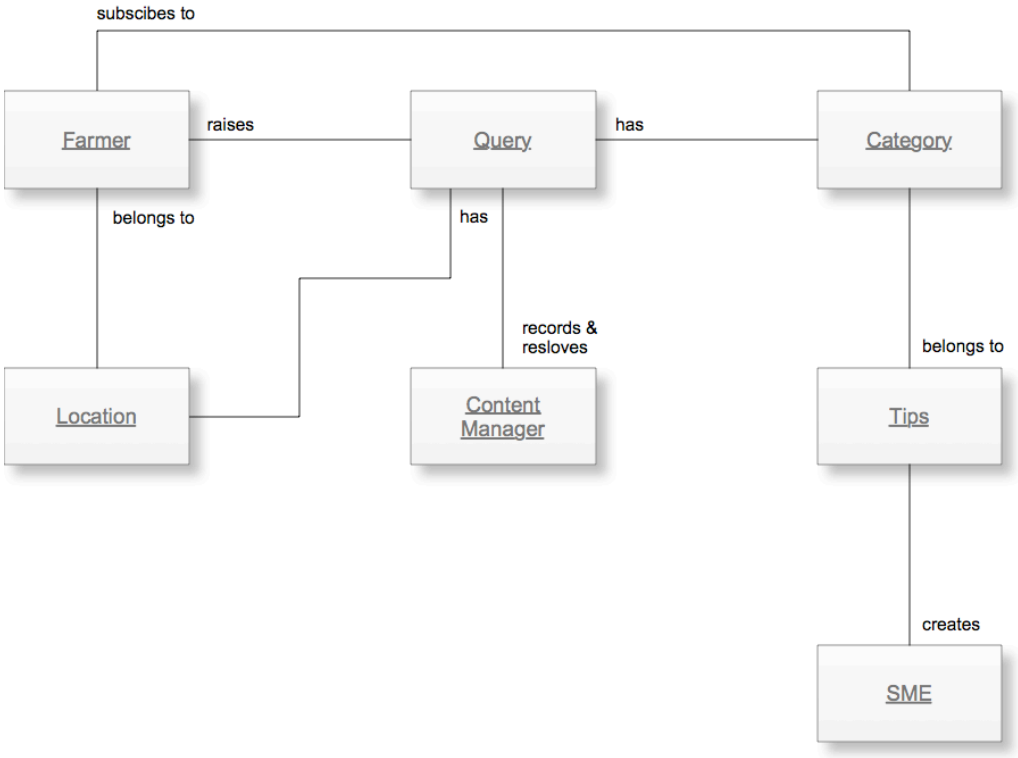
Use Case Element	Description
Number	UC.05
Application	The Subject Matter Experts keep bouncing useful tips that Content managers pass on to the Farmers who either come asking or subscribe to Information domains of Agriculture. These tips are specific to Information Domains as well as the Geographical region.
Use Case Name	Create TIPs
Primary Actor	Subject Matter Expert (SME)
Secondary Actor	None
Pre-condition	SME is an authorized user of the system
Trigger	The user clicks on the **Create Tips** link on the landing page.

Basic Flow	• System displays a list view of the Tips already created for the domain SME belongs to. The Tips also display the Count for the tip being referenced while resolving a query.
	
	• The user can select a tip to modify. The Tip details open up in a form for the user view edit.
	
	• The user creates a new Tip by clicking on Add Tip, a blank form appears with the following fields.
	
	  ◦ Tip
	
	  ◦ For State : Pick list from the State Master is displayed for user selection. this could be a multiple selection .i.e. the same tip is valid for multiple states
	
	  ◦ Effective Month for Communication: Month picker
	
	• The system saves Tip along with its Information Domain, for the States selected by user and Communication Month

Alternate Flow	Pressing Cancel abandons operation, no database gets affected
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. Farmer is the customer for the Smile-Farm Query Management System.
2. Farmers from various locations contact the Content Manager enquiring for assistance.
3. The Smile-Farm system maintains information Categories and associates all the content of query to these pre-defined category (Crops, Animals, weather etc)
4. The farmer subscribes to the information category(s) for seeking updates or information.
5. The queries are recorded by content manager on behalf of the farmer and resolved also based on the information displayed by the system for a similar query(s) in the past.
6. There are Subject matter experts who resolve queries for a category.
7. The subject matter expert also creates Tips and advisories that are useful for solving the query.

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