SPADE - Spare Parts Demand Forecasting System

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.

Spare Parts Demand Forecasting System - SPADE

Introduction

The purpose of this document is to define scope and requirements of a Spare Parts Demand Forecasting System for the services department of an auto major in India - **Concept Motors**. This will enable company's supply chain department to ensure smooth supply of spare parts across all service stations. The proposed system will provide structured forecasting mechanisms to plan timely availability of spare parts.

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

Management Summary

Concept Motors has a challenge to manage the smooth availability of spare parts across all the regions to ensure high customer satisfaction levels. Shortage of spares lead to lost sales opportunities and customer dissatisfaction whereas excess inventory meant prohibitive inventory carrying costs. To address this challenge, an effective demand forecasting system is envisaged. The proposed system - SPADE will deliver:

1. Multiple mechanisms of forecasting demand based on past consumption pattern
2. A graphical dashboard highlighting part-wise demand trends to the management

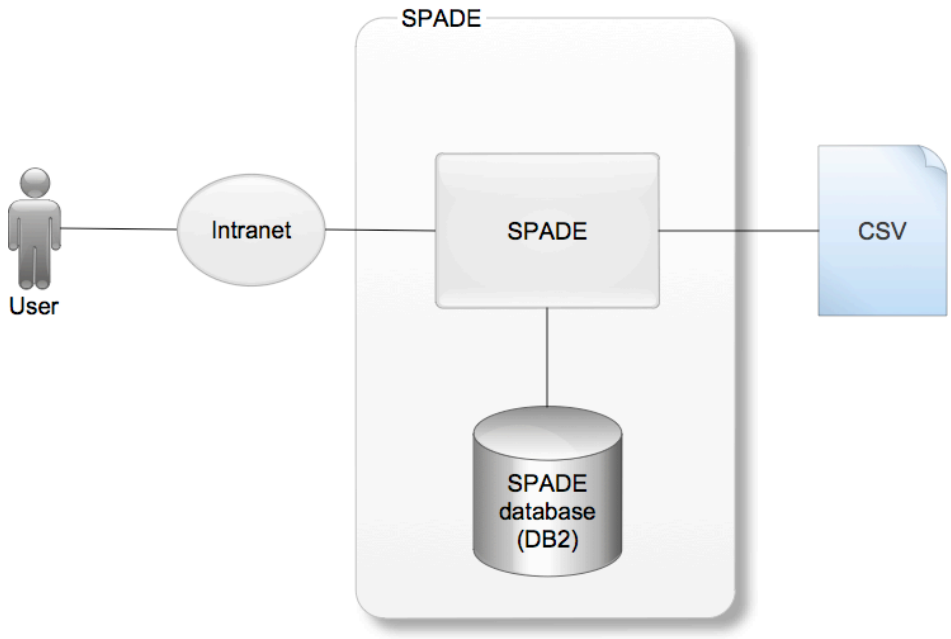
SPADE will be a web-based system and will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

Assumptions

1. The past consumption data of various parts will be available in a CSV format for upload.
2. CSV data will be uploaded manually on a periodic basis by the system administrator.
3. The system administrator will configure forecasting methods and their corresponding parameters & values for each spare parts category.

High Level Architecture

SPADE's high level architecture is illustrated through the context diagram shown in figure 1 below.



Context Diagram

SPADE	The proposed web-based application, SPADE will be accessed by the users over Concept Motor's intranet. Users will be able to access the management dashboard to plan parts supplies, view MIS reports highlighting deviation between forecast and actual demand. The admin user will be able to upload past consumption data, and configure a forecasting method for each spare part category.
CSV File(s)	The spare part master will be uploaded from the CSV file. The CSV has 3 columns viz. part code (can never be duplicate), its category (a part can belong to only one category), and part description. The past data will be uploaded in a CSV format. The CSV file has 4 columns viz. part code, year (YYYY format) , week (1 through 54) , amount consumed in numbers. *Our basic unit of period is always assumed to be in week*. Such data will be received from Concept Motor's IT Systems in a CSV format and will be periodically uploaded by the SPADE admin user.
SPADE Database	This will hold all the SPADE data including the uploaded data, spare part category configurations, and demand forecasts.

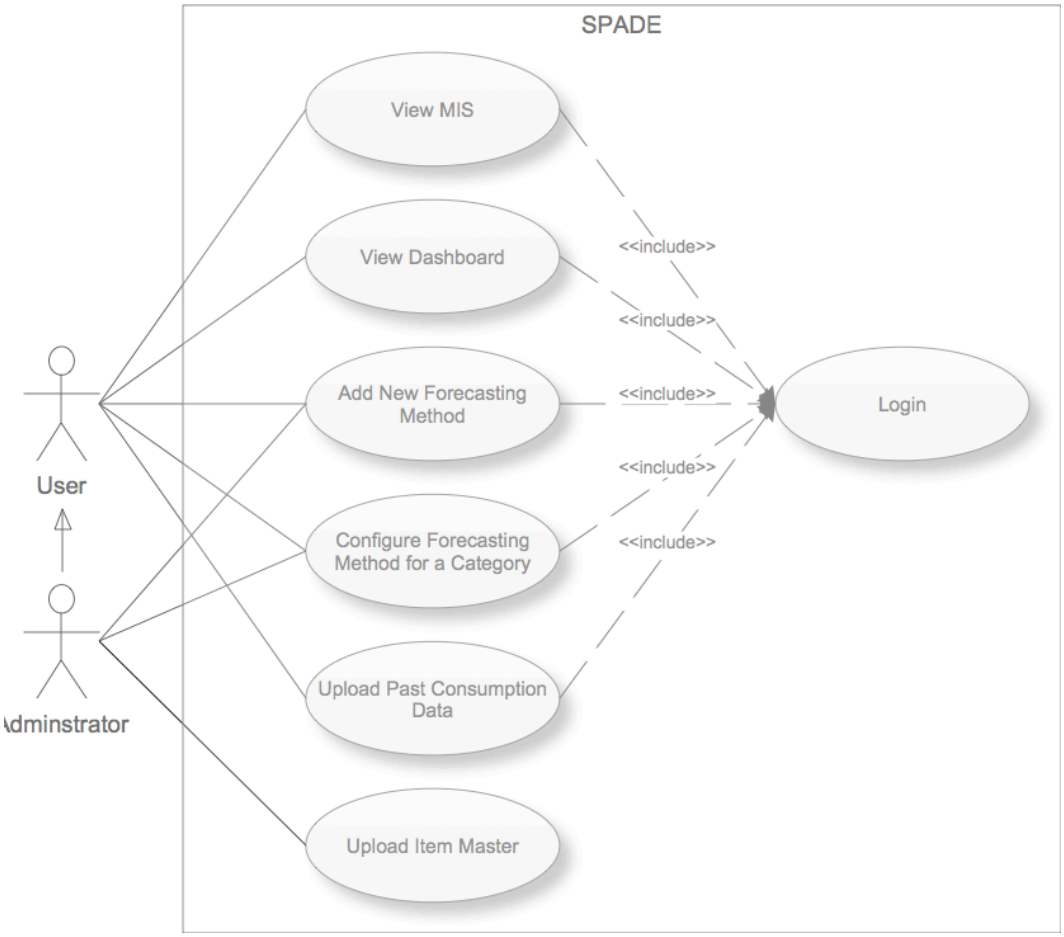
Functional Requirements

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

SPADE 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.

Use Case Diagrams

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



Use Case Diagram

Use Cases

Upload Part Master

Use Case Element	Description
Number	UC.01
Application	Uploading part master data
Use Case Name	Upload Part Master
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the **Upload Part Master** menu item on the admin interface 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 part master; if an existing part is encountered, the old part details are replaced with the new details.
Alternate Flow	• In even 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 parts updated (i.e. already existing part being uploaded)

Upload Past Consumption Data

Use Case Element	Description
Number	UC.02
Application	Uploading past consumption data; this data will be used to forecast demand and to compute deviation with actual consumption after the time system has gone live (as before that there was no forecast data!).
Use Case Name	Upload Past Consumption Data
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the **Upload Past Consumption** menu item on the admin interface 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 part master; if the consumption data for an already uploaded time period/part combination is encountered, it is ignored.

Use Case Element	Description
Alternate Flow	• In even 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 ignored (i.e. the ones for whom the data was already present).

Configure Forecasting Method for Spare Part Category

Use Case Element	Description
Number	UC.03
Application	Configuring a demand forecasting method and its parameter values for a spare part category
Use Case Name	Configure Forecasting method for spare part category
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the **Configure Category Forecasting Method** menu item on the admin interface page
Basic Flow	• System displays the configuration form • Administrator selects the desired category from a drop down menu • Administrator is now allowed to select a forecasting method; s/he selects the forecasting method; as a result, system now displays all the parameters that should be assigned values • Administrator fills the desired values and saves the details. The configuration details are accordingly updated. The dashboard will now follow the just updated category configuration to forecast demand of the spare parts belonging to the this category.
Alternate Flow	• If the selected category has been already configured, the current data is displayed and the administrator is allowed to change the same including selection of a new forecasting method. • Administrator has the option to cancel the operation any time.
Output	System displays a proper message on successful configuration of category configuration and allows to configure another category if required.

View Dashboard

Use Case Element	Description
Number	UC.04
Application	Viewing demand forecast for a part
Use Case Name	View Dashboard
Primary Actor	User (administrator is a kind of user)

Use Case Element	Description
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **View Dashboard** menu item on the home page
Basic Flow	• User selects the desired category from a drop down menu • User is now allowed to select a part for the already selected category • System displays the forecast through a simple line graph and a corresponding data is also displayed in a table format.
Alternate Flow	None
Output	Display of the demand forecast

View MIS

Use Case Element	Description
Number	UC.05
Application	Viewing part-wise MIS of deviation between forecasted and actual demand
Use Case Name	View MIS
Primary Actor	User (administrator is a kind of user)
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **View MIS** menu item on the home page
Basic Flow	• User selects the desired category from a drop down menu • User is now allowed to select a part for the already selected category • User now is allowed to select the period of MIS • System displays the deviation between forecasted and actual demand for the selected period.
Alternate Flow	None
Output	MIS View

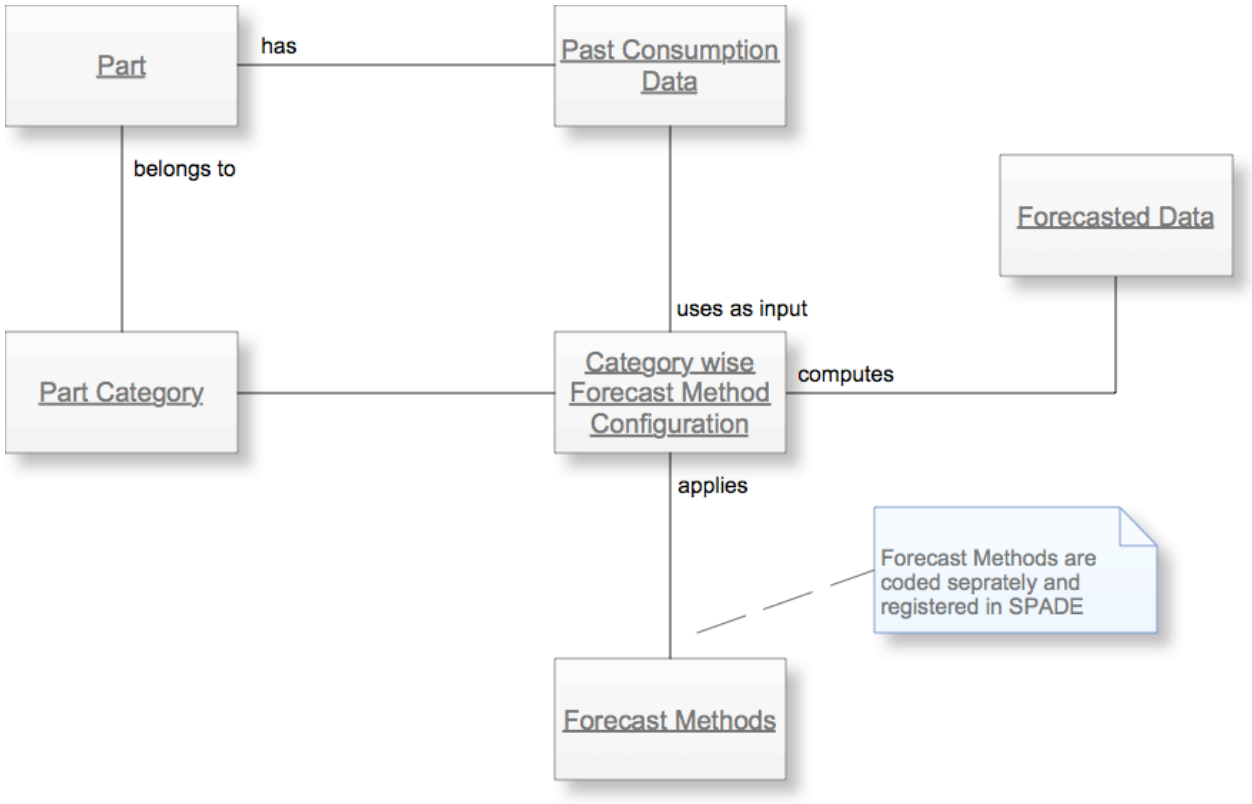
Add New Forecasting Method

Use Case Element	Description
Number	UC.06
Application	Add a new forecasting method that can be used to demand forecasting for a category (of spare part)
Use Case Name	Add New Forecasting Method
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the **Add New Forecasting Method** menu item on the admin interface page

Use Case Element	Description
Basic Flow	This involves addition of a new forecasting method in terms of its coded logic (for example, a class having forecast method, and required state variables), and a way to save the state in to a database. This logic can be typically coded as a new (enterprise) java bean and registered in to SPADE via a Add New Forecasting Method Form. Once it is registered in to SPADE, the configure forecasting method for a category form will start to show up this method in its "forecasting method" drop down list.
Alternate Flow	None
Output	System displays a proper message on successful configuration of category configuration and allows to configure another category if required.

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. Each Part belongs to a Part Category.
2. Part has its Past Consumption Data.

3. Category wise Forecast Method Configuration uses Past Consumption Data and applies the applicable Forecast Method to compute the Forecasted Data.

4. For each Part Category, there is Forecast Method Configuration. Note, the configuration details depend up on the Forecast Method chosen for that Part Category.

Forecasting Algorithms

Following algorithms will be made available for demand forecasting:

Simple Moving Average

A simple moving average method forecasts demand by computing the average consumption of a part over a specific number of periods (i.e. weeks in our case). For example, a 5-week simple moving average is the five week sum of weekly consumption divided by five. As its name implies, a moving average is an average that moves. Old data is dropped as new data comes available. This causes the average to move along the time scale. The example below illustrates the concept further.

Weekly Consumption: 11,12,13,14,15,16,17

First day of 5-week SMA: (11 + 12 + 13 + 14 + 15) / 5 = 13

Second day of 5-week SMA: (12 + 13 + 14 + 15 + 16) / 5 = 14

Third day of 5-week SMA: (13 + 14 + 15 + 16 + 17) / 5 = 15

Exponential Moving Average

Exponential moving averages reduce the lag by applying more weight to recent prices. The weighting applied to the most recent price depends on the number of periods in the moving average. There are three steps to calculating an exponential moving average. First, calculate the simple moving average. An exponential moving average (EMA) has to start somewhere so a simple moving average is used as the previous period's EMA in the first calculation. Second, calculate the weighting multiplier. Third, calculate the exponential moving average. The formula below is for a 10-week EMA.

SMA: 10 period sum / 10

Multiplier: (2 / (Time periods + 1)) = (2 / (10 + 1)) = 0.1818 (18.18%)

EMA: {today's consumption - EMA(previous day's consumption)} x multiplier + EMA(previous day's consumption).

For more details on the above methods, please refer to standard text books or resources on the web. You may select a third algorithm of your choice to demonstrate addition of a new forecasting method.

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

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, 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/