Storage Space Tracker

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.

Storage Space Tracker

Introduction

The purpose of this document is to define scope and requirements of a 'Storage Space Tracker' for a Carrying and Forwarding (C&F) agent with a warehouse, where the manufactured / shipped goods of various multinational companies are stored before they are sent to the wholesale market. Due to increase in customers, it has become difficult for the company to manually manage the warehouse space and ensure that there is no loss of business. The system required should be able to manage the warehouse space smartly and ensure that there is no loss in business by keeping spaces vacant..

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

Management Summary

A Leading Warehouse is trying to automate their Space Management system which will help manage their space, thus increasing number of customers. The management has much higher expectations from the new software. They expect the new software to give warehouse workers information that makes locating a particular shipment simple, clear, and unmistakable. It should enable to make better decisions and complete logistics cycles much faster. The software is intended to deal with receipt, storage and movement of goods. The system shall inform the customer as soon as their shipment leaves the warehouse premises, and about the availability of allocated space for them in the warehouse.

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

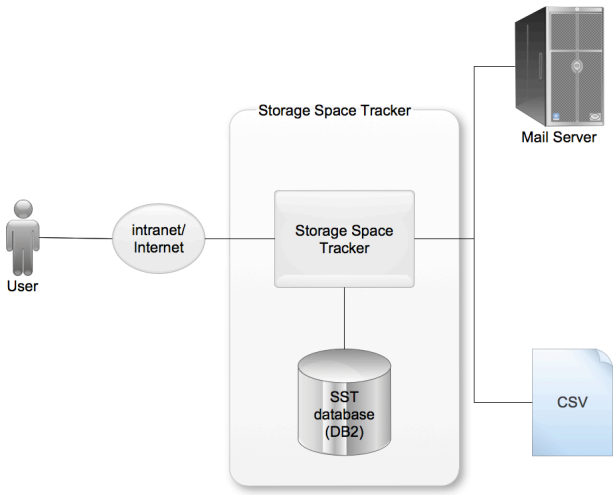
Key Assumptions

1. The developer of this project should be aware of challenges of warehouse companies, space allocation mechanisms.
2. The real life systems are complex in algorithms, for the project purposes the specifications have been kept simple.
3. The warehouse capacity is assumed as 5000 pallets. The customer will enter the requirements in pallets, the system will not compute the pallets required as per carton dimensions.
4. Customers are predefined with deposit money received and stored in the master record. Each time a billing takes place, the bill amount gets subtracted from the available balance.

High Level Architecture

Storage Space Tracker's high level architecture is illustrated through the context diagram shown below. It will have following categories of users:

1. Warehouse Admin
2. Warehouse Staff
3. Customer



Storage Space Tracker Context Diagram

Storage Space Tracker	The system will help manage availability of space while the receipts of goods and deliveries are happening simultaneously. The customers booking for storage of goods will get a clear status of space availability.
Storage Space Tracker Database	Customers, Warehouse transactions such as Receipts, Dispatches, users of the system
CSV	Customers, Pallets, Rates masters are uploaded via CSV
Mail Server	All notifications are routed through the Mail Server

Functional Requirements

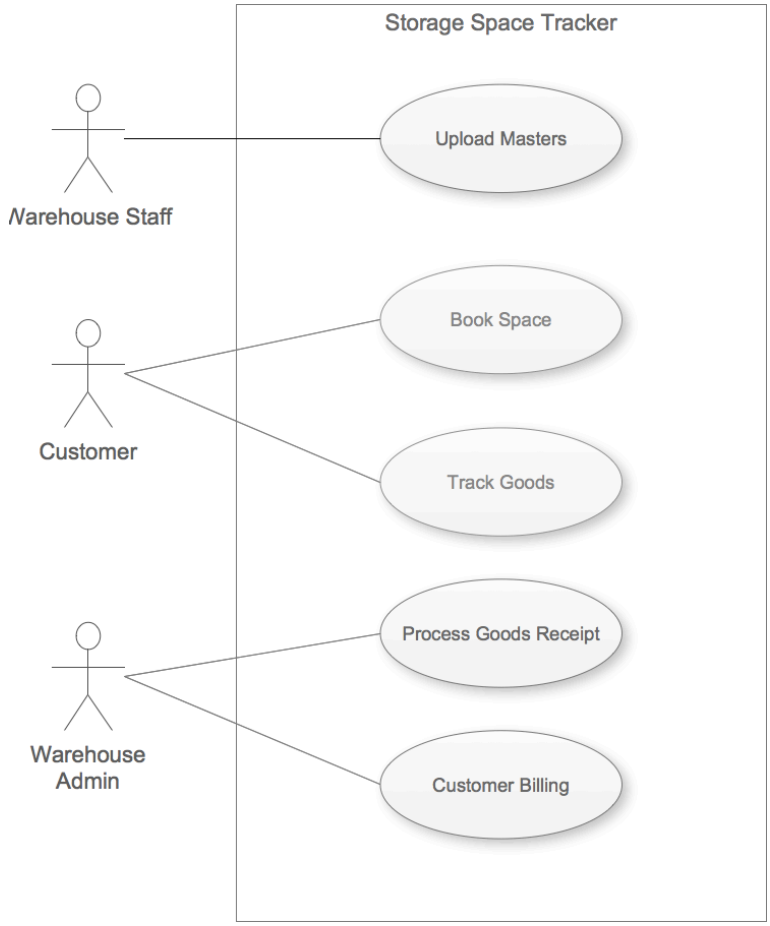
The high level functional requirements for the Storage Space Tracker are outlined in the Use Case diagram described in this section.

Storage Space Tracker 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 Storage Space Tracker. Internal users will be provided user id/password pair separately.

Once user logs in, 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
Application	Masters in Storage Space Tracker are uploaded using CSV Format Customer master will have customer id, company name, customer name, add1, add2, city, country, contact number, mobile number, email id, balance from deposit. Customer billing rate will have customer id, rate per pallet Storage pallet master will have pallet number, floor number
Use Case Name	Upload Masters
Primary Actor	Warehouse staff
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **Upload Masters** menu item on the landing page

Use Case Element	Description
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)

Book Space

Use Case Element	Description
Number	UC.02
Application	The customer books the space online
Use Case Name	Book Space
Primary Actor	Customer
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **Book Space** link on the landing page
Basic Flow	• The system displays a form for entering the booking information • The user fills data in Start Date, End Date, Count of Pallets The system validates in the backend for availability of Pallets for the Start and End date provided. The system validates for the available balance by subtracting (No.of pallets * rate) from the previous available balance. • The system displays more information gathering fields like Delivery address, Clearing required (Y/N), Description of goods, Number of cartons, Carton label information and image scanned. Use clicks on submit to complete the booking process. Booking reference is generated for tracking.
Alternate Flow	Alert for validation not cleared, let the user enter the information again
Output	None

Track Goods

Use Case Element	Description
Number	UC.03
Application	Goods Tracking is critical for the customer
Use Case Name	Track Goods
Primary Actor	Customer
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the **Track Goods** link on the landing page
Basic Flow	• The system displays list of bookings where the goods are yet to reach, are stored in the warehouse, have been billed. • The system allows the user to select any one of the entry for viewing details. • The user can also enter despatch details for yet to reach booking reference. • The system notifies the warehouse admin of yet to reach despatch details via email.
Alternate Flow	None
Output	None

Process Goods Receipts

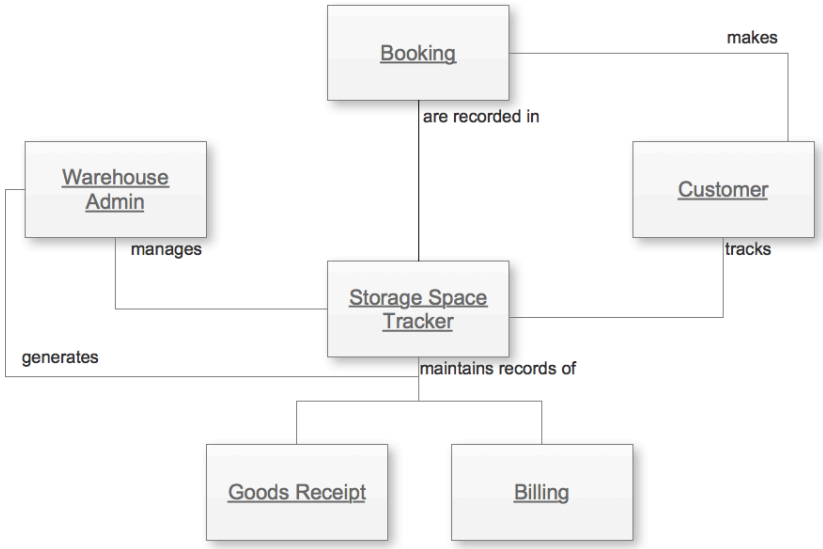
Use Case Element	Description
Number	UC.04
Application	The bookings due for goods receipt in the warehouse are processed from warehouse approval perspective.
Use Case Name	Process Goods Receipts
Primary Actor	Warehouse Admin
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the **Process Goods Receipts** link on the landing page.
Basic Flow	The system displays list of Bookings due for goods arrival on the current date. User selects one of the bookings, the system displays the details of despatch entered by the customer. User marks the booking as Ready for goods receipt. System notifies the warehouse staff for the bookings ready for goods receipt.
Alternate Flow	Pressing Cancel abandons operation, no database gets affected
Output	None

Customer Billing

Use Case Element	Description
Number	UC.05
Application	
Use Case Name	Customer Billing
Primary Actor	Warehouse Admin
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the **Billing** link on the landing page.
Basic Flow	• System displays the customer bookings that are due for despatch as per the booking end date in warehouse. • The user can select booking to modify. The details open up in a form for the user view edit. • User enters the dispatch time. • The user clicks on the bill button for automatic bill generation. • The system updates the pallet availability and bill is saved in the system for the customer. • An email notification of bill is sent to the customer.
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. The storage space tracker is the backbone of warehouse operations run by C&F agents.
2. The Customers make direct booking of space for the goods till they are shipped to their destination.
3. Storage Space tracker computes the available space and accordingly makes a booking.
4. The physical goods received into the warehouse are authorized by the warehouse manager. Until this step, goods are not placed in the space allocated.
5. Warehouse manager generates a bill once the goods are ready to be shipped out.

Database Design Guidelines

This involves the transformation of the use cases, state diagrams, and logical object model into detailed and optimize 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 Storage Space Tracker 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

Common Checks	Validation Type
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/