Policy Serve - Buying Insurance Online

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.

Buying Insurance Online - Policy Serve

Introduction

The purpose of this document is to define scope and requirements of Buying Insurance Online System for an Insurance Aggregator - **Jackson Group.** This will enable company to sell Insurance Products from various Insurance companies via a single portal. The proposed system will provide online shopping to customers, shortlist Product for further quotes. The Insurance Companies can view on what has been shortlisted from their offerings. Accordingly they can post the final rate for the customer to view and finalize the purchase.

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

Management Summary

Jackson Group is venturing into Insurance Aggregation business. They conceptualized an online solution for customers to provide smooth shopping experience while exploring General Insurance options for range of products under Automobiles. In a nutshell, their business model of partnering with Insurance company, creating offerings around their products with benefits is simplified in the proposed automation. The system will be drawing comparison between products and enable the customer to decide quickly. Customer has an option to shortlist for comparison or seeking quotations. The proposed system - Policy Serve will allow:

1. A single place to shop for insurance Products from various Insurance Companies.
2. Product offerings with standard and paid benefits are created.
3. Online shortlisting of the products based by prospective customers.
4. Price and Benefits comparison before requesting for final quote.
5. Insurance company quote with the final pricing on the customer's request.
6. The decision on request is concluded online.

Policy Serve 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.

About Insurance Regulations and Standards

IRDA: While developing a solution for Insurance Aggregators, its important to know about governing IRDA Act. *The Insurance Regulatory and Development Authority (IRDA) is a national agency of the Government of India, based in Hyderabad. Mission of IRDA as stated in the act is "to protect the interests of the Policyholders, to regulate, promote and ensure orderly growth of the Insurance industry and for matters connected therewith or incidental thereto"* Insurance aggregators have to comply with the regulations for conducting their business with Insurance companies and brokers. Thus the system created for this domain should ensure compliance factors governing the operations of the business.

ACORD: Association for Cooperative Operations Research and Development, is a global nonprofit standards development organization serving the insurance industry. *ACORD's mission is to facilitate the development of open consensus data standards and standard forms*.

In the real world application, the aggregator portal is a fairly complex application having multiple lines of business e.g. Life Insurance, General Insurance. They are further categorized into various types of insurance products. A company dealing in General Insurance would be dealing with Products for Vehicles, Travel, Health, Term etc. An aggregator partners with various insurance companies and Insurance Companies to showcase there products on these lines of business. Each insurance company would have their own set of benefits and execution processes. To create a scalable model for an aggregator portal thus is a mammoth technical challenge. Following ACORD standards will ensure data standards across various products being offered under multiple lines of business.

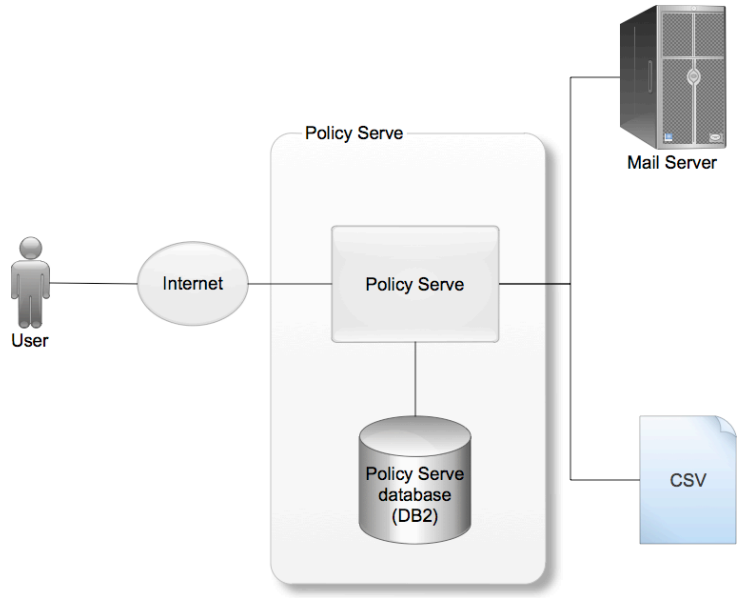
In this SRS, only Car Insurance is being considered as part of General Insurance Line of Business at Jackson Insurance.

Assumptions

1. The insurance aggregators website is governed by IRDA rules and regulations applicable currently. Thus validations and information sharing with insurance partners requires compliance.
2. As a compliance of the insurance domain, the scope of this project is defined within the guideline of IRDA.
3. ACORD standards are not implemented in this SRS, the mention of ACORD is for knowledge purposes
4. Online payments are kept out of scope of this project.
5. Role based access to be provided, Internal users shall login using their user ids and passwords.

High Level Architecture

Policy Serve's high level logical architecture is illustrated through the diagram shown in figure below.



Policy Serve	Policy Serve will be accessed by the customers, Insurance companies and company staff over the Internet. The prospective customers visit the site and buy insurance online from the products and pricing listed. They make an online request using shortlisted Product offerings from various insurance companies. The insurance company responds with a quotation based on customer's inputs. Customers, Insurance companies and Internal users have their own pages in Policy Serve.
CSV File(s)	Car Make master will have a single column as Make. Car Model will be a master with 2 columns Make, Model Name Model Year will be a master with 2 columns Year, Model Name Benefits will have 4 columns - Benefit Id, Benefit Title, Type (Optional/Standard) Insurance Company will have 5 columns - Company Code, Company Name, Contact Name, E-mail id & Mobile Number
Policy Serve Database	This will hold all the Policy Serve data including the uploaded data. Insurance Company offerings request for insurance with shortlisted products, Insurance Company quotes, Policy Concluded.
Jackson Web Server	Website application of Jackson. The Policy Serve link shall be available on this website
Mail Server	The mail server used for notification to the prospective customer and Insurance company on various events in the system.

Functional Requirements

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

Policy Serve will provide a secure user-id/password based secured login mechanism for internal users and Insurance Companies to access its features. The prospective customers shall access the application details by registering with their email-id and password. 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.

There would be separate landing pages for the 4 types of users. Customer, Insurance Company, Company Admin, Company Users.

Registered Customer - Present a form to gather basic details of the Insurance requirement and self. Display Insurance Company wise polices matching the requirement Customer should be able to shortlist the Products for price and feature comparison and request for a quote. Customer can check or uncheck the Products before submitting for the quote. The options to Request for Insurance, View Quotes should be present on this customer's landing page.

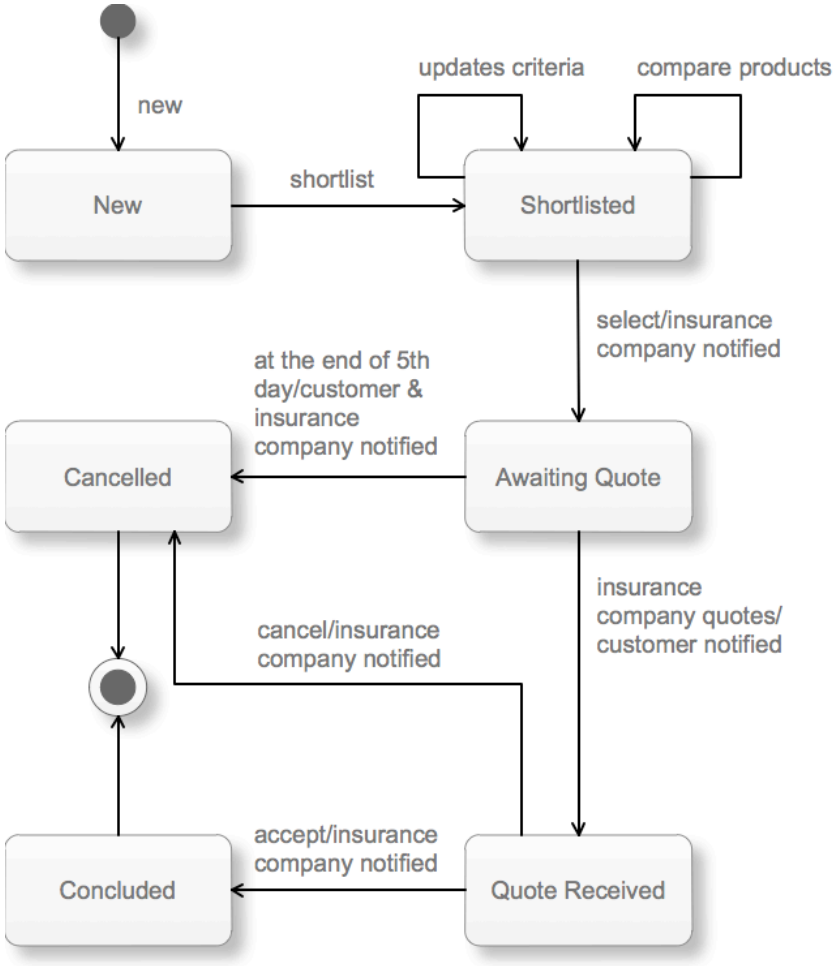
Insurance Officer - Request Queue, Quoted, Concluded

Jackson Administrator - Master Maintenance

Jackson Management - Dashboard

State Diagrams

The figure below depicts the state transition of Insurance requisition by customer to policy concluded.



1. Customer enters a **New** Request with self contact details and car related basic details
2. In the New state, customer can Shortlist offerings to evaluate from various Insurance companies.
3. In the **shortlisted** state of request, the criteria to evaluate can be modified and request to compare the selected products from shortlisted offerings.

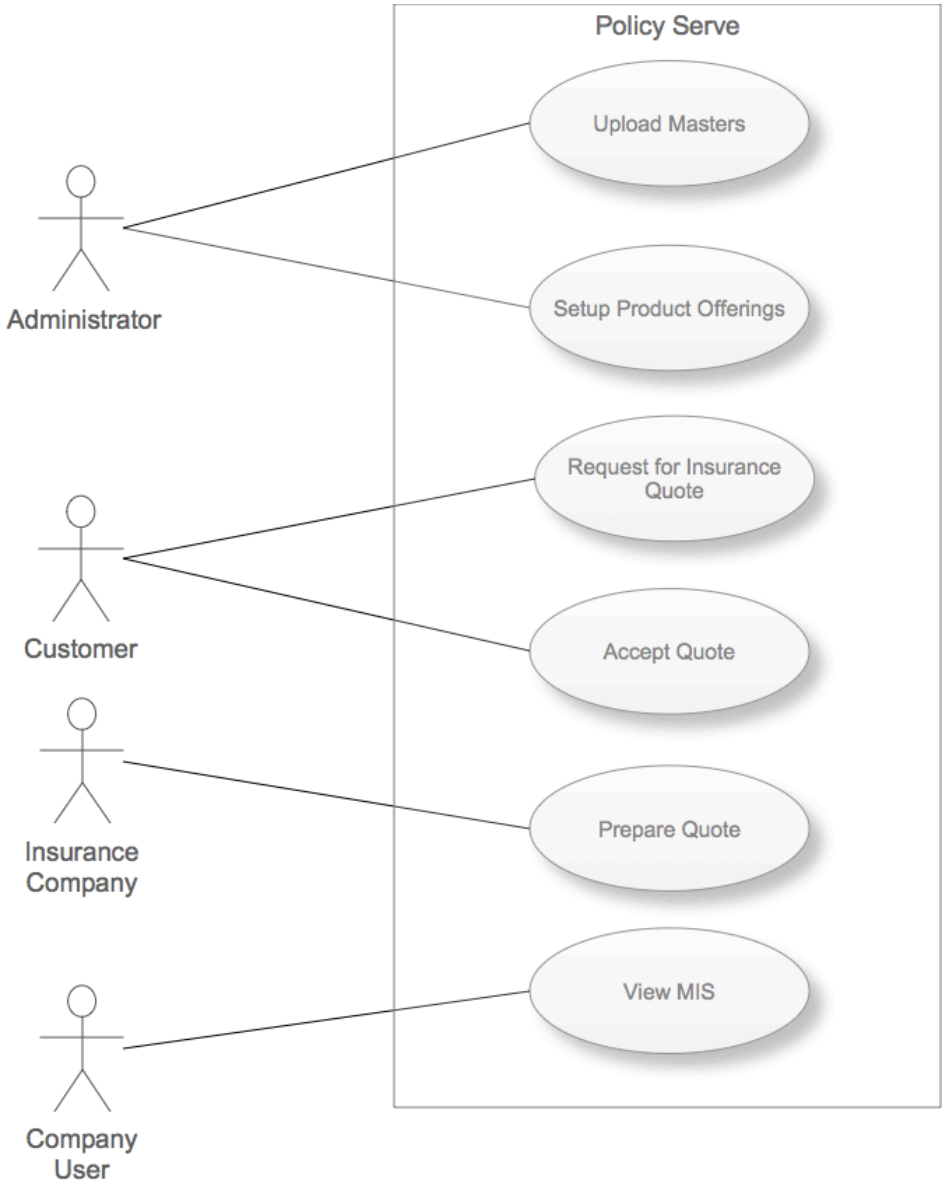
4. The customer can select 1 from the shortlisted offerings for getting a quotation. The state of request is transitioned to **Awaiting Quote**. The state of the request moves to **Cancel** state if the insurance company does not respond within 5 days of raising a request.

5. The Insurance company receives email notification for the customer request. Insurance company responds with a Quotation. The request is transitioned to **Quote Received** state.

6. The customer on receiving the quote can either Accept it or Cancel it. In the case of Accept the state of request moves to **Concluded** state, else it moves to **Cancel** state.

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
Application	Masters for the system to provide the user options to select from while doing entering request for Insurance. The list of masters for to be maintained are Car Makes (Vehicle Brand E.g. Ford, Chevrolet, Maruti etc,) Car Models (Ford Fiesta, Chevrolet Spark, Maruti SX4 etc.) Years (YYYY) Product Benefits - These are standard and optional products benefits derived from this master. The applicable charges for a benefit are provided as fixed charges in the master Standard Benefits normally provided are 24X7 Toll Free Helpline, Anytime anywhere claims reporting. Cashless claim settlement, Online approval on minor claims, Discount on Claim free experience An optional benefit examples would be Drivers insurance, Direct Settlement, Pickup and towing facility, Cleaning of vehicle, Warranty on accidental repairs. Insurance Company - Organizations that sell insurance products and are interesting in hosting and serving customers via the aggregator website. In real operations, the insurance company will have a formal approval process before their data is entered into the system. For the purpose of this project, bare minimum data is captured via CSV upload.
Use Case Name	Upload Master
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	A valid CSV file with records
Trigger	Administrator clicks on the **Upload Masters** menu item on the admin interface page
Basic Flow	• Administrator selects the master to be uploaded. • 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 master; if an existing record is encountered, the data is replaced on the record.
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.

Setup Product Offering

Use Case Element	Description
Number	UC.02
Application	The products of various Insurance companies are displayed as options for the customer to select from. The products are composed of Product title, its Unique selling proposition from the company, standard price as applicable to the Make, Model and year of the car. As a full fledged operations, the product offerings are setup by the partnering insurance company. For the purpose of project, the aggregator company Jackson is handling this task via their administrator.
Use Case Name	Setup Product Offering
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	All Masters are updated in the database.
Trigger	Administrator clicks on the **Setup Product Offering** menu item on the admin interface page.
Basic Flow	The following steps allows the user to setup a product offering. Step 1 • Select the Insurance company from a List • The details of Insurance company from master are displayed Step 2] • Enter Product Title [Text] • Enter Risk coverage description[Text] • Select Car Make+Model+Year from masters pick list • Enter Basic Price of the product Step 3 • Select the applicable Standard Benefits displayed from the master. Step 4 • Select the Optional Benefits displayed from the master. • Fixed charges for these benefits are entered. (Flat rate is being used here, in real case, it could be a % value to be computed as per some criteria) Step 5 • Save & Continue, Save & Exit and Cancel links are available for the product offering to be saved. • Click on Save & Exit, System runs validations none of the fields should be blank. • Click on Save & Continue, the system displays a new form for the same Insurance Company with the same details as filled previously. The user can modify the required fields and continue to create offerings for the insurance company.

Use Case Element	Description
Alternate Flow	Operation is cancelled, no data is captured in the database
Output	System displays the message, New Product Created.

Note: if the time permits, product edit and delete use cases can be included in the application

Insurance Request

Use Case Element	Description
Number	UC.03
Application	The prospective customers comes to aggregators website to shop for insurance. To look for the product that meets complete or maximum of requirements, the customer is asked to filled a request form. The system looks up options available for the Car make, model and year. Displays Name of Insurance company, product title and risk benefits
Use Case Name	Insurance Request
Primary Actor	Customer
Secondary Actor	None
Pre-condition	None
Trigger	Applicant clicks on the **Request for Insurance** link on the website of Jackson Insurance.

Use Case Element	Description

Basic Flow	• System displays the request form with following fields in two sections:
	• Section: Car Basic Details
	  ◦ Previous Policy Expiry Date [dd/mm/yyyy] Date Picker
	  ◦ Year of Manufacture : MMM/YYYY
	  ◦ Car Make : Pick list from the master
	  ◦ Car Model : Pick list of Models valid for selected Car Make
	• Section: Contact Details
	  ◦ Name
	  ◦ Mobile (Pre-fix 91, 10 digit number)
	  ◦ Date of Birth (picker for DD, MM, YYYY)
	  ◦ City of Residence
	  ◦ Email id
	• Click on search options, saves the application in the database as New Request.
	• The system looks up the Insurance Company offerings for Car Make+Model +Year of Manufacture.
	• The criteria matching offerings are displayed with a radio button for user to shortlist. User selects products using radio buttons.
	• Link for Compare Products or Modify Search Criteria
	• Click on Compare moves the request to Shortlisted state.
	• The system displays a form with comparison rows giving price and list of standard and optional benefits across the Insurance companies selected. The benefits are displayed as ticked or crossed as applicable in each cell (Insurance company & Benefits)
	• Navigate to shortlisted offering view for user to select the insurance company for final quote
	• Customer selects one insurance company from the previously shortlisted offerings.
	• A form section to select the optional benefits from the insurance company is displayed

Use Case Element	Description
Basic Flow Continued	• The customer enters the balance fields to complete the request like • Variants of the Car, Fuel Type, Any Claims taken in previous policy term, Electrical accessories value, non electrical value of accessories, CNG Klt(Y/N) • Discounts like Voluntary excess, if preferred then enter the amount, member of ARAI association etc. • Click on link to Request quote. The Request now moves to Awaiting Quote status. • Email notification is sent the Insurance company for responding with a quote • Response could be received instantly/ in few hours/ few days • The IRDA rules apply here. As per the regulation the insurance company partnering on Aggregator portal cannot keep the lead beyond 5 days. Thus the system will have a backend activity that will automatically mark the request as Cancel, if no response is received from the Insurance company. For the purpose of testing, you may test with few hours as a no response case.
Alternate Flow	• User can Cancel the request before submitting for quote. The record does not get saved in the database.
Output	Email to Insurance company for the quotation request.

Insurance Company Responds with Quote

Use Case Element	Description
Number	UC.04
Application	The Insurance officers at the Insurance company receives the request in their queue. Computes and enters the final price. The request gets a status of Received Quote
Use Case Name	Respond with Quote
Primary Actor	Insurance Officer
Secondary Actor	None
Pre-condition	None
Trigger	The Insurance officer clicks on **Request Queue** menu item on the Insurance company's landing page of Policy Serve

Use Case Element	Description
Basic Flow	• A view of new request received from the prospective customers is displayed along with the time elapsed since the request was received. (The system displays only those requests that are in “Awaiting Quotation” status and have the shortlisted offering from the Insurance company whose officer is processing the Request Queue. • Click on one of the request, the system opens a request form in read only mode. • Pricing and benefits section opens up for the Insurance Officer to update. • Insurance officer enters the pricing as computed using the inputs received. (For simplicity it has been taken as manual data entry) • The customer’s specific requirement benefits are considered for entering charges for each entry. • The status of the request on submit quote becomes ‘Received Quote’. In a real case example, the request could have multiple status. Such a case would take place when customer has selected more than 1 offering from different Insurance companies. The request would be considered as responded only when both responses are received. Where as in the current case, the selection for a quote is restricted to only a Single Offering.
Alternate Flow	Cancel will terminate the response activity and take the control back to the Request Queue
Output	None

Note: In real case, the insurance process might require some documents to be furnished along. The list of such documents can be linked to the product offering. In the current project, the document attachment has been kept out of scope.

Policy Concluded

Use Case Element	Description
Number	UC.05
Application	The customer checks the response from the insurance company and makes a decision
Use Case Name	Policy Concluded
Primary Actor	Customer
Secondary Actor	None
Pre-condition	None
Trigger	Customer clicks on the **View Quotation** menu item on the home page
Basic Flow	A view of Product with updated pricing and benefits included is displayed. A link to Accept Quote or Cancel is displayed on the page. Click on Accept Quote. The status of Request is marked as “Concluded” Note: Policy issue process is not included in the current scope

Use Case Element	Description
Alternate Flow	Click on Cancel marks the request status as ‘Cancel’
Output	Email to customer for the next steps of policy issue.

View MIS

Use Case Element	Description
Number	UC.06
Application	MIS
Use Case Name	View MIS /Dashboard
Primary Actor	Jackson Group’s authorized users
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **View MIS** menu item on the landing page
Basic Flow	Dashboard view of the following is displayed on complete page. (For a date selected, by default current date should be selected) • No. of Requests received, processed in a location. • Requests processed within the 5 day SLA, • Requests missed SLA • Processed for the day and Awaiting Quote • Product wise Request Concluded • Insurance company wise Request concluded • Insurance company SLA performance across 1 day through 5 days
Alternate Flow	None
Output	None

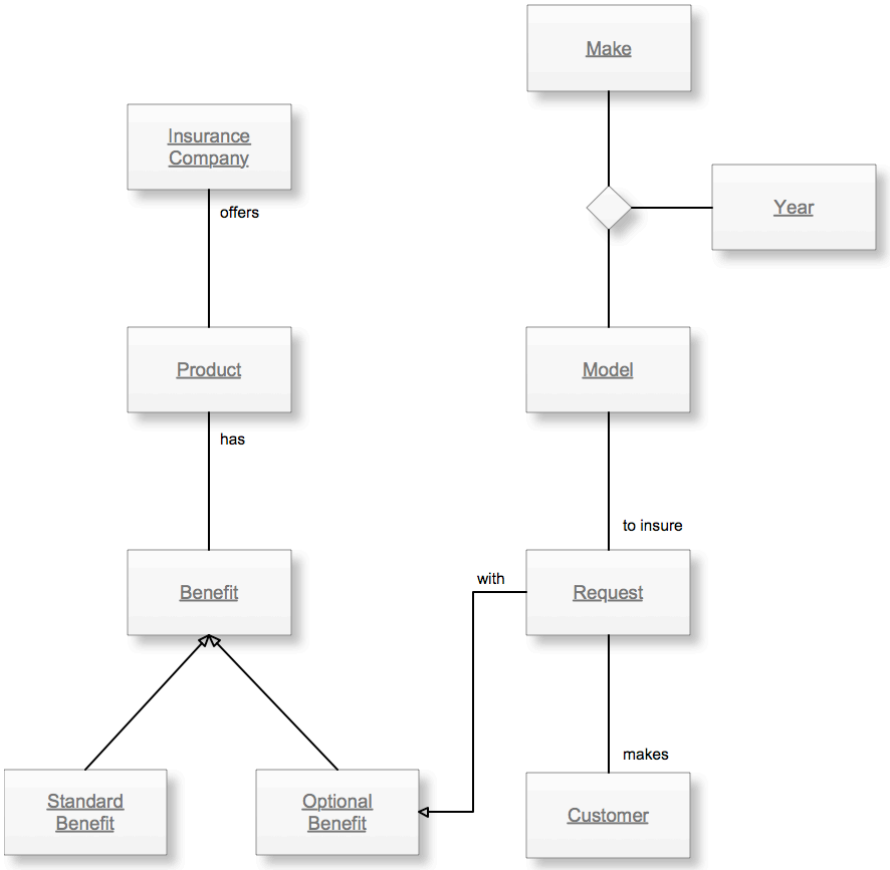
Other Use Cases

If the time permits, the developer can further extend the functionality of this application by providing features like

1. Insurance officer can view
    a. Status of the requests being worked upon
    b. Sending reminders for any clarifications required.
    c. View overall performance per Insurance officer of Request processed within 5 days.
    d. Expired requests as the SLA is missed.

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. An Insurance Company offers Insurance products on the Aggregator's website.
2. The product offerings are created for customers to select from.
3. The product has two components i.e. 1) Car Specifications & 2) Benefits
4. Car Specifications are a function of Make, Models' of a Make and the Year in which the Model is valid. The master for the car specification is derived from three separate masters.
5. Benefits are of two type 1) Standard i.e. provided by as a service without any additional charge e.g. 24*7 support, tow away etc. 2) Optional Benefits are provided at an additional cost like drivers's insurance, discounts on automobile bodies membership.
6. Customers on raise a request for insurance on Aggregator's website.
7. Request specifies the criteria for searching the available offerings for the specifications requested.

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

Common Checks	Validation Type
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/