LIC Market-Driven System LIC, an insurance company, wants to digitize a range of business processes and provide a complete solution that addresses all aspects of the agent-insurer relationship. Consider yourself as a part of the Requirement Analyst team at Retinodes Software Company, and your job is to gather and prioritize the set of requirements. In this new requirement of the project, there are no existing systems that can be analyzed for the development. Requirements have to be gathered, negotiated, validated and prioritized through multiple stakeholders which is a complex process because all stakeholders have different perspectives, requirements and priorities. Therefore, Retinodes want to have a requirements engineering framework that can be used in market-facing projects. To start with, you need to identify the set of stakeholders associated with the system, the domain information about the insurance market, and possible features. The first product LIC wanted you to develop consolidated insurance packages which can compete with the packages provided by other insurance companies. Another product is based on the customer priority, based on the insurance policies available the customer can create his/her own package and send a request for the review. The system has to automatically analyze the package, provide suggestions (if any), and at last give a competing price for the package. To understand the problem domain, existing packages have to be analyzed and the demands and restrictions from the insurance policy and agents have to be understood completely. The requirements and feasibility report generated by you, will be further used by the development team for implementation.

1. Identify all the stakeholders and users of the systems. Enlist all features of the LIC Market-Driven system by each user of the system, in the form of user stories. Can you prioritize them using the requirement prioritization techniques? (e.g., AHP, Numerical Assessment, MoSCoW method, etc.) How? Provide details.

Stakeholders:

- Government of India
- Policy Holders
- Customers
- Policy Makers
- Investors
- Insurance agents
- Employees, Manager
- Outsourcing companies
- Banks

Users:

- Insurance Holder
- Insurance Agent
- Employee, Manager, DB manager, AI
- Stakeholders and Investors, Government

Features of the LIC Market-Driven system:

1) As a user I should be able to create an account so that I can be a part of LIC.

2) As a user I should be able to log in.
3) As a user I should be able to see all the policies that are available.
4) As a policy holder I should be able to pay the premium of the policies periodically through the digital payment system.
5) As a GoI clerk I should be able to see the data and analytics in some time period so that I can submit the relevant data to the government.
6) As a stakeholder I should be able to suggest new policies to optimize the gain.
7) As an Insurance agent I should be able to create packages to bring new packages to the market.
8) As an Insurance agent I should be able to get suggestions to bring tempting packages to the market.
9) As a user, I should be able to create my own packages based on the available insurance policies so that I can send a request for the review of the packages selected by me.
10) As an insurance agent I should be able to track all the details of the policyholders that are under my assistance.
11) As a user, I should get notifications for the next premium due date.
12) As an agent, I must be provided with the updates as soon as there are any changes in the policy so that I can notify customers.
13) As a user, I should be able to see all the policies on the web application of LIC so that I can compare with other companies' policies.

Yes we can prioritize the features using MoSCoW prioritization technique.

MoSCoW

The four capitalized letters in the MoSCoW prioritization scheme stand for four possible priority classifications:

Must: The requirement must be satisfied for the solution to be considered a success.

Should: The requirement is important and should be included in the solution if possible, but it's not mandatory to success.

Could: It's a desirable capability, but one that could be deferred or eliminated. Implement it only if time and resources permit.

Won't: This indicates a requirement that will not be implemented at this time but could be included in a future release.

Ref: https://medium.com/analysts-corner/five-requirements-prioritization-methods-86f4c5e0433e

Must - 1, 2, 3, 4, 9, 10, 12
Should - 7, 8, 13
Could -5,
Won't - 6, 11

2. Prepare a list of market-facing technologies helpful for this project. According to you, would market-facing technologies be helpful in the proper deployment of the product? Why?

Front-end Technologies: A-js, HTML framework
Back-end Technologies: DB, Django (acid properties to be followed in DB)
Deployment: Heroku, Dev-ops

Or

WordPress - Website builder

Marketing Attribution & Management, Digital(SEO/PPC) Marketing System
Customer Experience Software

If the digital system comes to the market it will have a great impact on the customer base of LIC, because the main thing is that it will become more easier for the policy holder to do the payments of their policies, to the new users to see the different types of available policies according to their needs, insurance agents related to the system to maintain all the data of the policyholders under their assistance, and there are many other advantages as well, so the thing is that due to the digitalization in the system and the ease of the process it will have an impact on the number of customers.

3. Suggest an effective requirement engineering framework that can be used in market-facing projects because there are no existing systems that can be analyzed for the development so we need to consider all requirements from the core.

→ Here we're unsure about the requirements, we can use a simple prototype model and focus more on requirements elicitation of the provided prototype for betterment and addition of features in our original prototype.
→ We will use the framework functionality as below;
Firstly to Develop requirements correctly by prototype upgrading, then retrieving the Document Requirements and check completeness, we will also include analysis, refinement, and decomposition of requirements and at last we will validate and manage requirements.
We can also do some market research and stakeholder selection based on expertise of the product, sponsors and deliverables and also gathering the requirements by interviewing people.

4. List out the possible features those are not feasible to consider. Can you provide justification for each of them in detail?

→ Online Verification of people (Aadhar card, Photograph, Fingerprint) we will have to rely upon KYC centers.
→ Complete credit rating is not possible to be done completely online with higher accuracy as it will involve high risk and less accuracy.
→ In some health insurance policies the agents have to check all the health issues that are to the customer by doing some physical test or checking the health reports of the customer, which is not possible digitally.
→ Loans can't be provided to a customer without checking his/her credit rating and all the proper documents related to him, i.e. we have to rely upon the KYC centers.

5. Let us assume that the customized package developed by the customer (using your second product) is similar to the package available in your pre-defined package. What is the possible reason behind this defect? How can it be ensured that this would not happen? In which requirements engineering activity, this defect can be handled? Please provide a scenario to justify.

There can be several reasons for the customized package being similar to the predefined package:

1) The pre-defined package is optimized to the financial capacities of the customer and any more customization to it would not yield any significant benefits to the customer which in turn would render the customization feature useless.

2) The analysis of the system can be poor which as a result will not provide a competing prize to the customers and will not interest them.
One of the ways to go around this is, we can provide users with different types of short term or long term benefits which would interest our customers to choose and customize their package thus, providing flexibility to the customers.
Yes, we can take the data in certain formats to validate the package accurately. In the validation stage we will first check the package with the existing one from our database. And already the suggestion system is asked to be developed, so we will suggest it as already existing even if it is not the same but similar. As far as the development is concerned we can include it in a new prototype.

6. Identify three different use cases where the conflicts between the requirements occur? Do you think that the conflicts can be resolved? How?

→Here one conflict can occur when the company is competing for some package which is very common for people and some how to tackle the market The system provides the price which can get the company to lose, then the system has to take care of this kind of scenario and alert the company.This can be solved by leaving the price as it is or remove the package from system.

→There is also some possibility that the package the company provides has more price and the same package made by some customer can have less price.So companies here have to take care of customized packages and they should select the price on the basis of some basic packages.They can redirect the customer to the same package company has.Or they can give more price than the actual price so that customer will choose the company's package.

7. Considering the set of features you have identified, what are the non-functional aspects associated with this system? Explain rationale behind the selection of each of them.

Durability: All transactions or any policyholder or customer's data should stay intact even in case of any hardware or software fault.
Performance: System should respond within some milliseconds of time.
Reliability: System should be available 24/7.
Scalability: Even if the number of users of the system increases, the system should remain reliable. Also, it should remain stable and available while adapting to changes, upgrades, overhauls, and resource reduction.
Privacy: Insurance policies and documents which the customer has uploaded must be kept in an encrypted manner and only accessible to the customer.
Authentication: Customers can request policies only after verification of his /her documents.

8. Can there be 'Open Issues'- issues those are identified but not taken care of? If yes, what are they? Are there some alternative ways for their resolution, such that no requirements conflict will happen?

→ Loan related work will be difficult to do in online mode because of the security or duplicate documents issues.
→ Online document verification is not possible. Life insurance claim or accident claim, this kind of claims require proofs like many different documents related to death,accident certificates etc. But documents can be fake and the system can't check if the documents are valid or not. So, this can be an open issue of LIC.
→ Trust issues on companies