Case-Study : LIC Market-Driven System

Group-7 Members

- 201801031-SRISHTI KALRA
- 201801045-TANAYA ALPESHBHAI SHAH
- 201801055-KHATRI RAHUL NARAIN
- 201801083-NANAVATI SWAPNIL DIPANSHU
- 201801088-SOLANKI ABHISHEK PRIYAKANT
- 201801115-JIMIL DIPESHKUMAR
- 201801167-SHAH DHANVI SWETAL
- 201801173-MANGUKIYA ARSHIT DIPAKBHAI
- 201801227-PANCHAL KRUTARTH BHAVESHBHAI
- 201801242-PATEL PRIYA HARSHADBHAI

---

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:

- **Customers** - These are the individual persons or the business companies that tend to buy various insurance policies.
- **Employees** - These are the people who work in the organisation and are incharge of providing some of the base policies/packages.
- **Policy Makers -** These are the individuals that provide policies to the customers with the help of the system.
- **Backend Database** - This holds the information about the current customers and its details and also has the data of the policies.

Users:

- **Customers -** These are the individuals who would be utilizing the software to buy and personalise different insurance policy packages.
- **Agents -** These are the people who will be using the software to provide the customers with their needs.
- **Administration -** These are the individuals who will be handling the backend database and would be amending the database with respect to the customer interactions.

User Stories:

- As a user of the system, I should be able to create an account on the software.
- As a user of the system, I should be able to login into the system.
- As an agent in the system, I should be able to create an account in the software.

- As an agent in the system, I should be able to login in to the system.
- As a user of the system, I should be able to review different prices to avail the best price for a particular policy.
- As a user of the system, I should be able to customize packages best suited for me based on the existing policies.
- As an agent in the system, I should be able to create new packages.
- As an employee, I should be able to provide policies/packages that best suit the customer's needs and requirements.
- As an admin, I should be able to access the details of the customers and the policies as and when required.
- As a user, I would like to check the status of the policies.

⇒ Yes, we can prioritize them using the requirement prioritization techniques like the MoSCoW method. In this method, we can prioritize the user stories in four ways:

- Must have
- Should have
- Could have
- Will not have

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?

Market Facing Technologies :

- These are basically the use of technology to attract and target the customers.
- The following ways can be adopted-
    1. Customers can be given special discount offers if they buy a package (more than one policy). For eg- 10% discount on buying 2 policies and then additional 5% discount for adding each subsequent technology.
    2. Customers can be given a facility to customize policies on their own, as per their own requirement, and they can request a review for the policy that they have created.

- Study of the existing Market Facing technologies play a major role in the successful deployment of the project as we need to compete with these existing technologies as well, so a careful study of these technologies would help us get a better idea of the market trend/ requirements and help us build a better software that would be able to compete in the market.

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.

➔ Taking into consideration the scenario, we can use the Agile Method and SCRUM process model as requirements are not clear from the beginning and need to gather from the beginning.

Requirement Engineering Framework for Market Facing Technologies:

- This is basically the aggregation of steps for the developers in order to proceed further accordingly.
- As we don't have the already existing system for analysis, to start with :
    - Initially develop and gather requirements
    - Write the proper documentation depicting the requirements in detail
    - Then check for the completeness of the documentation, and make changes if needed.
    - Then, Analyze the requirements mentioned, think of an alternative if any, refine the requirements and finally validate the requirement documentation
    - Work for a Market Research and interviewing people, this would help in better understanding of the requirements from the perspective of users as well

  In conclusion, this software serves as a better alternative to attract the users.

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 the documents (Eg. Aadhar Card, Fingerprint, Pan Card details, etc.), but for that we might need the KYC option which is not feasible.
- Complete credit rating is not possible as with online systems, it lacks accuracy and involves high risk.
- Considering some health insurance policies, insurance company needs to verify all the health issues, though can be done through certificates, but still needs in person verification as people might get some wrong certificates, thus this is a feature which might be lacking.
- For providing loans also, proper verification of person must be done, which is not quite possible working with online system.
- Above mentioned are some features, which we might want to include, but cannot specifically include due to certain restrictions.

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 are multiple reasons why the pre-defined package doesn't match with the customer-defined package. One of the reasons can be a mismatch in the amount to be paid per month. As a customer, he/she wants to control the amount of spending he/she does on a policy/package. Another reason could be the maturity date of the policies in the package. These small details are of utmost importance to the user and thus, even though similar to predefined packages exist, these details cannot be ignored.
In order to avoid the scenario of almost-same packages of the user and pre-existing ones to clash with one another, one can create a checkpoint for the user where the user can be suggested certain packages based on the requirements selected. If the user agrees with that package, the pre-defined package is selected. If the user does not like certain details of existing packages, the user continues creating his/her own policy package.

6. Conflicts in requirement and its solution:

1) There can be conflict while understanding what is a discount price and how the discounts are being applied when we take a package.
Solution: This conflict can be resolved by providing more information regarding how discounts are being applied.

2) If a user creates a package that is already provided in the system of LIC, then the price for that particular package should be the same at both the places, if the prices are not matched then there can be conflict.
Solution: This can be solved by setting proper price distribution techniques for various parameters that determine the price of a policy.

3)There can be conflict in consideration of the user's benefit or agent's benefit while the system is analyzing the packages created by users or the agents.
Solution: This can not be resolved because everyone has their own priorities.

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.

Scalability :

a. System should be scalable and handle multiple requests at the same time.
b. Customers should be able to use the software from any device.

Reliability :

a. If any time a failure occurs to the system the data should be backed up in a failsafe storage.
b.
c. The System should be available 24x7.
d. The Response time should be quick so when a customer creates a package, the system should provide the competing price for the package as soon as possible.

Security:

a. There should be 2-way authentication during login to prevent any fraudulent transactions.
b. The insurance policies and related important documents must be kept in an encrypted manner and only be accessed by the customer.

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?

- Online verification is problematic -
  There is no way to check the authenticity of the documents that are submitted online.
- There is no way to make sure that a large number of people can access the system at the same time without the system getting crashed.
- A customer wants to consolidate a group of policies into a certain package but the company's policies do not allow the user to do so.