1.Question:-

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.

Stakeholder:

- Firm itself
- Policy Manager
- Policy Holder
- Agents
- Database Admin
- Bank
- Normal users
- Premium users
- Guest User
- NRI User

USERS:

- Normal users
- Premium users
- Guest User
- NRI User
- Employees
- Agent
- Policy Manager

Functional Requirements :

Normal User Features

- Display policies which are premade by the firm.

Premium User Features

- Can create customizable policies

Common Features

- Users should be able to create a account
- Users should be able to log-in or log out of the system
- System should be able to assign agents to the customers who are new and dont have much idea about the policy system.
- Review all the policies a user has already purchased
- Able to compare policies according to their inputs.

Requirements Prioritization (MoScoW Technique)

User Stories:
MUST
As a user I want to view my policies
As a user I want to be able to create policies
As a Policyholder I should be able to pay a premium.
As a user I should be able to create or login account

SHOULD
As a user I want to be able to compare between different policies
As a user I want the system to protect my personal data.
As a Policy Manager I should be able to add,update policy details.
As an Agent I should be able to access all details of the policyholder.
As a policyholder I should be able to get notifications of premiums.

COULD
As an insurance agent, I should be able to view my performance analytics.
As an insurance agent, I should be able to view the amount I would be receiving this month as commision.
As a user, I should be able to see suggestions based on my search regarding packages.

WOULD
As a user I should be able to get notifications for new policies

2.Question :

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 like Agile could be very useful, in preparing the proper deployment path as it is a robust method to employ and include all the ideas and views of the market.

Also the technologies like digital signage and kiosks would be useful for users who want to pay their premiums using cash or cheques.

3.Question:-

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.

We can use Agile methodology for our requirements engineering technique, as it is robust and can provide the perfect base for developing such applications which have

no predefined structure, also it involves all the users that are a part of the general application, it allows us to incorporate the requirements properly .
So user stories are the way to elicitate the requirements from the users

4.QUESTION :-

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

- There might be a case where customers might not be satisfied by the packages provided by the software.
- There can arise a case where custom policies may clash with the market guidelines.
- There can be a division of policies based on age groups and implementing can be very much difficult.
- 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.
- Automatic validation of documents is a valid requirement but its implementation is not possible
- Having 24x7 customer support is not feasible
- Having group policies with people of different ages would be tough to implement

5. QUESTION:-

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.

Since the product is market driven, and the competition in the market is such that the maximum benefits with the lowest prices is the package or product that is preferred by the customers, hence the premade packages already made according to market demands will be the ones which give reasonable low prices for maximum benefits and the customers whilst customizing this would prefer this sort of packages or deals which according to them give them maximum benefits for lowest prices and hence their customizations will be similar to the premade packages.

In order to avert this sort of a thing we can enhance the customization tool itself by providing a functionality of comparing it with similar premade packages so that before they submit their customized package, they can come to know about other similar packages that fulfill their requirements in a better way and hence they would not develop packages that are similar to premade packages. This comparison tool inside the customizing interface can also show how adding or removing an insurance in a package can be profitable or lossy, and make the package stronger or weaker.

6. Question :-

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

1)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.

2)Another conflict can be that some package is giving a different price with some feature and if any customer makes a customized package from that package because (s)he does not need all feature and if this package has more price than the basic package that (s)he uses as basis than the system has to be changed.the system should calculate price on the basis of basic package and if it has defect than they should change the basis package.

3)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.

4) The automated pricing model can sometimes create unfair prices because of reasons related to the intrinsic algorithm and such rare cases should be handled individually having another layer of verification on top thus ensuring that the pricing model does not exploit the customer due to exceptions in our learning/adaptive model

7.Question:-

Non Functional Requirements:

System should be scalable
System should be easy to use by the end users
System should be able to handle multiple users at the same time
System should be available 24x7
System should be secure as to protect the private data of the users
System should be flexible to run on any platforms like android, iOS and PC.
System should be reliable and robust.
System should be able to categorize the policies to suit the custom needs to various customers
System should be quickly responsive in order to give swift notifications to the customers regarding reminders of upcoming and overdue payments.
System should have very less a latency

8.Question:-

Can there be 'Open Issues'- issues that 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?

Issue: There might be a case when a fraud document is sent for document verification.

Issue: There may be a very optimized package that could be created by a user or a popular package that many users may be customizing, would that package be available to other users?
Resolution: The user/s can submit their package to the package developers and the company can decide whether or not to that package will be profitable to the company and hence implement them and make them open to the general populace