**IT314: Software Engineering**

**LAB 6**

**Project: Dynamic Time Table**

**Group 30**

**Group Members:**

Margi Hingrajia: 201801014

Priyal Raj: 201801106

Yash Patel: 201801134

Pratik Parmar: 201801211

Harshil Goti: 201801130

Maharshi Vaghela: 201801216

Nishant Shah: 201801403

Jaydeep Machhi: 201801452

Manish Khandar: 201801222

Jaykumar Darji: 201801460

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:
  + Life Insurance Company
  + Investors
  + Agents
  + Brokers
* Users:
  + Agents/Brokers
  + Insurers
* User stories:
  + Agent:
    - As an agent, I should be able to develop consolidated packages.
    - As an insurance agent I should be able to track the details of all the policyholders under my assistance.
  + Insurer:
    - As an insurer, I should be able to create a customized package and send it for review.
    - As an insurer, I should be able to get a competing price for the package.
    - As an Insurer I should be able to pay the premium of the policies periodically through the digital payment system or any other payment method.
    - As an Insurer I should be able to compare the different policies so that I can choose best for me.
    - As an Insurer I should be able to bookmark some policies.
    - As an Insurer I should be able to renew my old policies.
    - As an Insurer I should be able to add my contact details so I will get the latest update.
* requirement prioritization techniques:
  + We can use MoSCoW model (Must, Should, Could and Won’t)

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?

* Content Management System
* Email Marketing, Social-media Marketing
* Customer Experience Software
* Marketing Attribution Software
* Various insurance technologies like insurance dekho.com and policy related technologies like policybazaar.com

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, since the requirements are not static. We prefer using incremental model for this project. Whenever a new set of requirements comes, it can easily be reflected using an incremental model very efficiently. [CRISP](https://ibsintelligence.com/wp-content/uploads/2017/01/Requirement-Engineering-CRISP.pdf) requirement engineering framework can be used to gather requirements from the customer. CRISP framework can help us in ensuring the following things:

**C**overage

**C**larity

**R**epresentation

**R**eferenceability

**I**ndependent

**I**ndexing & Cross referencing

**S**tability

**S**pecificity

**P**rioritisation

**P**ragmatism

We prefer using the CRISP model because it’s a very high level framework and the team works in a very systematic way. Team has a predefined set of requirements for each domain which can be helpful here because we don’t know anything about the software beforehand. Team uses best practices for eliciting requirements and uses proper tools for documenting and indexing requirements. Team also uses different techniques to accelerate the process of requirement gathering so that software can be made as early as possible.

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

Analysis of the packages created by the customer that are to be checked by the system automatically is not always feasible to consider, as the implementation of such a feature which has to provide suggestions and an efficient price comparison might not be accurate for all the cases. It is a little difficult to generate and maintain such a complex system.

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.

⇒ This is possible when the system prioritises custom packages more than predefined packages. Means system gives customers to choose custom packages before predefined packages.Customers do not know about predefined packages.We can ensure by showing predefined packages before customization.we think Prototype is the best technique to identify this defect.We can detect by analysing present system. Because in Prototype, we can get a demo and by analysing it, we can find it.

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

1. Determining the price for the user generated package : the system which determines the price of the package and the packages which are already there should match the price and any additional scheme should increase the price.
2. The way users want to customize the package might not be consistent with the restrictions provided by the company.
3. There can be conflicts in prioritizing the packages which are already in the system and are user generated.

The conflicts which lie between different stakeholders can be resolved through a group meeting between conflicting parties and discussing the requirements. Other conflicts can be handled by considering all the edge cases by rigorous testing.

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.

1) Insurers must awer about terms and conditions while choosing a LIC packege.

It’s necessary that every insurer is read term and condition because it will help to

understand the package as well as not getting any problem in the feature.

2) insurers have to understand demand and restriction of insurance policy and agents.

It helps to go ahead good insurer-agent relationship and also take good policy by

looking All policies demand.

3) The system has to change the database according to the regular insurance policy and their package.

4) Customers have some limitations to create their own packages. So that anyone does not suggest any package that’s not possible.

5) Systems have to be reliable, modifiable. So the system may not fail while you are taking policy. If some change in requirement then the system has to be modified.

6) There has to be data-back of all insurers, their accounts, details of all policies. So if the system fails at some time then we can’t lose data.

7) System should have a reliable transaction platform for payment of insurance policy.

8) Privacy must be there, so information of the insurer can’t get out of the database.

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?

* Suspicions(trust issues on companies)
* Documents can be forged and so we can fully trust the clients. This is one big open issue related to identity theft.
* To always approve customized packages might be complicated