

## **Lab 6 Software Engineering Case-Based Learning Exercise**

### **Group 26**

Group members:

201801103 Aaditi Roy  
201801128 Zenil Kothari  
201801129 Hiren Mangroliya  
201801174 Shreya Joshi  
201801186 Maulina Raina  
201801231 Manish Parmar  
201801236 Asutosh Sangada  
201801407 Pratvi Shah  
201801435 Sarthak Patel  
201801444 Bhavya Ganatra

Flow: The customers are the prime users of the system and on the company side the proper functioning is maintained by the agents in terms of regular updates and ensuring proper functioning of the system in terms of policies and packages that are produced as the system output. The admin is the supervisor of the agents and the whole system.

Questions:

**1. Stakeholders:**

- a. Owner of the insurance company (LIC)
- b. Agent
- c. Insurer (customer)
- d. Employees of LIC
- e. Investors
- f. Manager

Users:

- a. Customer
- b. Agents
- c. Admin

User stories:

- a. As a customer, I would like to be able to login with the given credentials so that I can view my profile.
- b. As a customer, I would like to be able to view the different Insurance packages so that I'm able to compare and choose the best option.
- c. As a customer, I should be able to create my own package based on the available Insurances so that I can customize it according to my needs.
- d. As a customer, I should be able to send a request to review my package.
- e. As a customer, I could be able to prioritize some policies as per my requirements.
- f. As a customer, I would like to give feedback on different agents.
- g. As an agent, I should be able to login with the given credentials so that I can view my profile.
- h. As an agent, I should be able to see the policies to which I am assigned to.
- i. As an agent, I would like to input the different policies.
- j. As an agent, I would like to be able to update the already existing policies.
- k. As an agent, I should be able to track the details of the customers and the policies that they have taken that are under my supervision.
- l. As an agent, I should be able to analyze the details of all of the customers and the policies that they have taken.
- m. As an admin, I want to be able to see all customer's profiles.
- n. As an admin, I want to be able to analyze all of the users that are interacting with the system.

→ We can prioritize the given requirements using the MoScow technique.

- MUST (Mandatory) :

- As a customer, I would like to be able to login with the given credentials so that I can view my profile.
- As a customer, I would like to be able to view the different Insurance packages so that I'm able to compare and choose the best option.
- As an agent, I should be able to login with the given credentials so that I can view my profile.
- As an agent, I should be able to see the policies to which I am assigned.
- As an agent, I would like to input the different policies.

- SHOULD (Of high priority) :

- As an agent, I should be able to track all of the details of the customers and the policies that they have taken under my supervision.
  - As an agent, I would like to be able to update the already existing policies.
  - As a customer, I should be able to create my own package based on the available Insurances so that I can customize it according to my needs.
  - As a customer, I should be able to send a request to review my package.
  - As an admin, I want to be able to analyze all of the users that are interacting with the system.
- COULD (Preferred but not necessary)
    - As a customer, I should be able to prioritize some policies as per my requirements.
    - As an agent, I should be able to analyze all of the details of all the customers and the policies that they have taken.
- WOULD (Can be postponed and suggested for future execution) :
    - As an agent, I should be able to give my email/contact details to the customers.
    - As a customer, I would like to give feedback on different agents.
    - As the admin, I should be able to access the feedback of customers for each agent so that efficient workers can be rewarded
    - As an agent, I would like to get customer feedback regularly.

## 2. Market Facing Technology:

- Existing technologies/ platforms like policybazaar.com, insuredhondho.com, etc can form a starting point to our solution
- As we use cloud servers and different server technologies to store the data and manipulate the data, the cost of operation goes down.

- As we use new technologies and frameworks such as react js or react native for the frontend, express js like helping frameworks for the backend, the speed of development can be increased.
3. Requirement engineering should be done very efficiently and the logs for various users should be maintained. The user stories would be an efficient way for requirement gathering and then addressing the requirements while creating a prototype would be an efficient and better way. It is the best way to club similar types of requirements together and try to implement it in a systematic way. Agile SCRUM development would be a suitable model for this kind of project. Daily Scrum would also ensure transparent and efficient work.
  4. Possible features that are not feasible to consider
    - It is not possible to satisfy the budget constraints of the customers at all times. The first product will give us the most optimal price for a given package so if a similar model was suggested by the customer with lower pricing then that would not be feasible
    - The user might wish to have some package that has each requirement from a totally different package but providing every permutation and combination of requirements is not a feasible option.
  5. There can be possibly many reasons for the customized package being similar to the predefined package. The predefined packages are optimized for benefits to companies as well as the customers. And, customization to it made by the customer would lead to more benefits to the customer and maybe in some cases a huge loss to the company's profit. And so, customization features will be useless in this case as it will be a loss to the company.  
 One way to overcome this problem is to provide different types of benefits related to the short term and long term. So, customers will choose packages from already available packages and will customize some functionality based on his/her requirements. And, also if a customer ignores an already available package and makes a request to a directly customized package then the system should suggest a similarity with the already available package, and in addition to that a list of such packages should also be made available for the customer to explore.
  6. User Case Conflict:

- a) The definition of the optimal package should be the same for agent curated package and website-generated package. As the system-generated package to compete would not yield maximum profits for the company(Investors) and that could lead to conflicts.
- b) One of the main requirements of the customer is that the latency should be as small as possible but that may be costly and that is contradictory to the Investors' interest since their goal is profit.
- c) Customer wants that the competing price that is given by the machine should be as low as possible (they would want that the system accepts the price that they have provided) but the system would give the most optimal price after analyzing the customizable package. So requirements would clash as the customer would want the lowest price possible, but the company's requirements would differ.

The above conflicts are difficult to resolve.

#### 7. Non Functional Requirements:

- The system should have minimum maintenance time as this would allow efficient functioning of the system
- The first product model should be able to learn from the second model in order to accommodate the most popular package choices of the customers.
- There should be two-phase Authentication during login to prevent any fraudulent transactions
- The system should have a reliable and efficient transaction platform for payment.
- In case of login from more than one device or log in from a new device a notification should be sent to the customer's phone.
- The system should be scalable and should be able to accommodate a large number of customers.
- This system should work on all types of devices.
- The system should work on all types of browsers and operating systems.

#### 8. Open Issues:

- It may be possible that a certain permutation of features in a package may not be feasible for the company but is requested by the customer in the second product. Thus a situation like this will need to be handled carefully.
- Categorization of customers on the basis of their financial documents will be required to put a limit on the number of (or amount of) insurance allowed. The verification of these documents can be a tedious task.

- The documents needed for verification can be forged so the authentication of those documents can not be ensured if the system does not have any offline mode.