Software Engineering Case-Based Learning Exercise

LIC Market-Driven System

201801092 SHAH ARPIT JAYESHKUMAR
201801116 HARSH ASHOK DESHPANDE
201801028 AKSHYA GARG
201801210 VALA SKAND ARSIBHAI
201801153 PATEL KIRANKUMAR MAHESHBHAI
201801262 PAGHADAR HET KISHORBHAI
201801052 HITESH GOYAL
201801025 HEMANG SHRIMALI
201801455 MAISURIYA VARSHITKUMAR BAKULBHAI
201801165 VAZIRANI DIVYA RAJESH

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 and Users**
  - Customers
  - Employee
  - Investors
  - Supplier
  - Vendor
  - Agent

- **User Stories**
  - As a customer, I can login in my account, so that I can view my insurance policies.
  - As a customer, I want to see all the insurance that I have taken So that I can keep track of those.
  - As an employee, I should be able to suggest a good insurance package to the customer.
  - As an employee, I should be able to provide competing prices for the package of the user.
  - As a customer, I can create my own insurance package.
  - As a customer, I can send a request to review my own package.
  - As a customer, I should be able to upload my documents.

- **Features of LIC Market-Driven System**

- Registration
- Login
- View Policy / Insurance
- Track the Records of the policies taken by the customer
- Payment
- Review/Analyze User Insurance Package
- Suggestions
- Upload documents

We can use MoSCoW method for prioritizing the given tasks where there are four possible priorities Must (must be satisfied), Should(important and should be included if possible), Could(Desirable capability, implemented if time and resource permit) and Won't (will not be implemented at this time but could be included in future).

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:

- Predictive analytics
  - Pricing and risk selection
  - Identifying customers as a risk of cancellation
  - Anticipating trends
- Machine Learning
  - Process data and learn on their own, without supervising
- Artificial Intelligence (AI)
- Internet of things (IoT)
  - Can automate much of data sharing
  - For better-determining risks, rates etc
- Blockchain data
  - To build customer trust
  - Fraud detection and prevention
- Social media data
  - For marketing strategy and clever advertising

Yes, market-facing technologies would be helpful for the proper development of this product.

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.

1. Develop requirements
2. Document Requirements
3. check completeness-
4. Analyze, refine, and decompose requirements
5. Validate requirements
6. Manage requirements
7. Market Research
8. Stakeholder selection based on the expertise of the product, sponsors and deliverables
9. Interviewing people

We can follow an Incremental approach to develop market-facing projects

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

5) Let us assume that the customized package developed by the customer (using your second product) is similar as 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 defect can be caused due to poor View Policy as we are not showing the packages that are already there.
This defect can be handled by giving all the packages the User is requesting before so that the user does not request the same package which is already there in the System
This possibility occurs because the predefined package will be somehow optimized regarding customer requirement, So it is highly possible that an average customer's package will match with the predefined.

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

- View Policy is not - Customers sending packages that already exist in the system . This conflict can be resolved by providing a better UI and proper view facility so that all the customers can see all the packages before sending their own suggestion

- Customer doesn’t like the suggestion provided by the system about the insurance package - This can be resolved by giving users the freedom to customize his/her insurance package as many times he/she wants

- Payment - Customers don't like the price the system has provided for the package. This conflict can be resolved by giving /suggesting multiple packages to the users with different prices so that the user can choose the price that is suitable for him /her.

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

- Availability: 24x7
- Reliability: Mean time between failures should be very low.
- Privacy: customer data should not leak from the database

8) 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?

1. Whether the System is a Physical system (Insurance bank) or mobile/web app
2. Users can take multiple insurance policies at a time ?
3. Whether the system will be available for 24 hours ?
4. Is there an alternate suggestion that the system can provide so that a user can choose the best package to his/her interest ?
5. User wants the package according to his/her price.