5/10/2018

Analysis and Design Project: Information System for Sentimental Analysis

Group 9

Gayathri Etraj Janaki

Sathya Pramod Bhat Muniyooru Nataraja

Sneha Prakash Naik

Analysis and Design Project: Information System for Sentimental Analysis

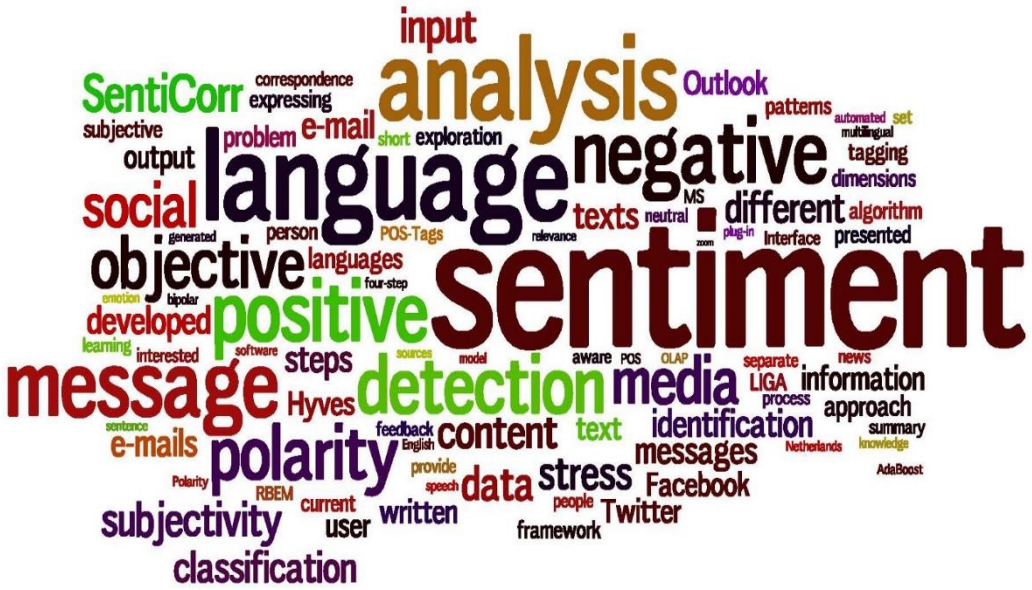


Table of Contents

Introduction:

We plan on designing a system with Sentiment Analysis for Product Rating framework that identifies hidden emotions in remarks and rates the product accordingly. The framework utilizes sentiment analysis keeping in mind the end goal to achieve wanted usefulness. This project is an E-Commerce web application where the enrolled client will see the product, product features and based on which he/she will comment about the product.

The Framework will break down the remarks of different clients and will rank the products. We will be using a database of sentiment-based keywords along with their positivity or negativity weight in database. The client comments will be dissected by contrasting the comment and the keywords put away in database. The System will take remarks of different clients, in view of the remark, framework will indicate whether the product is great, awful, or most noticeably bad. When client login to the framework he/she can see the product and its features. In the wake of review product, client can remark about the item. Client can likewise see remarks of other users. The part of the administrator's job is to add product to the framework and to include related keywords in the database. Client can without much of a stretch discover and amend the product for his/her utilization. This application additionally acts as a promotion which makes numerous individuals mindful about the product. This framework is likewise valuable for the clients who need an audit about the product they are interested in.

System Request: Sentiment Analysis based on Product Reviews

Project Sponsor:

The sponsors for this project is the team of managers who had come up with this plan of utilizing the reviews and providing a proper experience to the customers, at the same time investigating how the business could make profit out in this scenario.

Business Need:

- **Improvement in the sales of a product**:

  Based on the user reviews we will show which product needs improvement in the area required for the growth of sales of that product.

- **Analysis of reasoning behind developing or performing changes to a product.**

  The products on sale are taken into consideration without proper reasoning and this gets rejected at high level management resulting in continued lower sales. Analytical reasoning will help the manufacturer understand the motive behind the changes that have to be brought about for the project. The reasoning will be based on the user reviews on your own e-commerce website.

Business Requirements:

- **We need the reviews of the product**.

  The separation of positive and negative reviews is done with the help of keywords from the reserved keywords set. The split-up of the reviews will further help us decide the improvement on the product under consideration.

- **The details of the product which must be analyzed.**

  The product chosen for analysis will have its own specification, this will reflect on the selection of the reserved keywords. Even though we would have a generic set of key words, with the help of the details of the selected product the reserved keyword selection can be done in a more concise manner.

Business Value:

- This system will help improve the product quality or any faultily which might have been present in the initial release of the product. This in turn will improve the sales of the product.
- The improvement in sales of every product is a major plus for the economy of the company.

Feasibility Analysis:

Technical Feasibility:

Familiarity with the functional area.

The functional area for the development team will be the same IDE (Integrated Development Environment) as the webpage. In return helping in increasing the familiarity of the functional area.

The products used for analysis can be of varied range, the usage of the products will help the analyst to get a better understanding of the reviews to sort the keywords for the analysis. As we are dealing with already existing products and reviews the system will find less risk while developing a new system for sentimental analysis.

Familiarity with the technology.

The reviews obtained from the webpage already have a web platform which we will only be extracting the text from. This makes it easier for the developers to proceed as they have already worked on the development of the webpage.

The increased familiarity decreases the risk in development of the product.

Project size.

A team of 10 developers

The number of developers required will be based on the number and the variety of products involved in the analysis. We will conform ourselves to a limited number of products to see the feasibility and testing phase of the product in real time. For a smaller product set, the number of developers required will be less, with the team size varying from five to ten will be sufficient for the initial production of the system.

Compatibility**.**

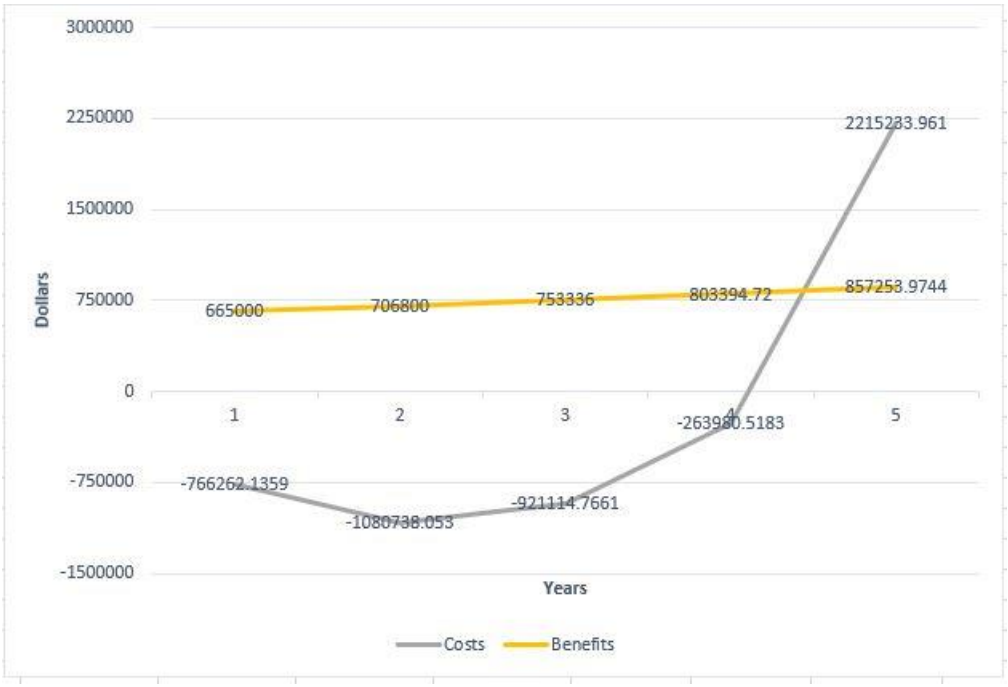
The compatibility of the new system will be co dependent with the technology.

Economic Feasibility:

The costs can be broadly classified into two categories i.e., **Development Cost and Operational Cost**. While the Development cost is a one-time initial investment, the Operational/progression cost is a periodic expenditure to keep the organization functional. The following expenditures can be considered as development cost, which are: Office setup and maintenance, Application software's, Development Labor. Whereas the operational costs are software licenses, Server software's, hardware's and operational labor.

The Tangible benefits include Improvement in subscription and Advertisements

Year	1	2	3	4	5	Total
Improvement in subscription	500000	540000	583200	629856	680244.48	
Advertisments	170000	171800	175236	178740.72	182315.5344	
Total Benefits	670000	711800	758436	808596.72	862560.0144	
PV Benefits	650485.4369	670939.7681	694076.3796	718427.7127	744051.8455	3477981.143
PV of all Benefits	650485.4369	1321425.205	2015501.585	2733929.297	3477981.143	
2 servers @ $150000	300000	0	0	0	0	
Office Setup [Help desk support]	200000	0	0	0	0	
Software Licences	125000	0	0	0	0	
Server Licences	9250	0	0	0	0	
Development Labour	600000	0	0	0	0	
Total Development cost	1234250	0	0	0	0	
Hardware - Server	50000	50000	50000	50000	50000	
Server Software	25000	25000	25000	25000	25000	
Operational Labour	150000	157500	165375	173643.75	182325.9375	
Total Operational Cost	225000	232500	240375	248643.75	257325.9375	
Total Cost	1459250	232500	240375	248643.75	257325.9375	
PV Costs	1416747.573	219153.5489	219977.1764	220916.7514	221971.6141	2298766.664
PV of all Costs	1416747.573	1635901.122	1855878.298	2076795.049	2298766.664	
Total Project Benefits Costs	-789250	479300	518061	559952.97	605234.0769	
NPV of all costs	-766262.1359	451786.2192	474099.2032	497510.9613	522080.2313	
Yearly NPV	-766262.1359	-314475.9167	159623.2865	657134.2478	1179214.479	915233.9609
Cumilative NPV	-766262.1359	-1080738.053	-921114.766	-263980.518	915233.9609	
Return on Investment	39.81413057					
Break even point	4.223861327					



Organizational Feasibility:

Proposed project is beneficial only if it can be turned into information systems that will meet the operating requirements. This test of feasibility asks-

- If the system will work when it is developed and installed.
- Are there any major barriers to Implementation?
- If there will be any resistance from users that will affect the possible benefits, or they will welcome the change and the new system.
- If the system will work when it is developed and installed.

The project was proposed as we wanted to make to make a simplified web application that provides the customers with a better shopping experience by providing them a proper reviews and feedbacks of what other customers of similar sense feel. It is simpler to operate and can be used in any webpages. It is free and not costly to operate.

Requirement Definition:

Functional requirement:

Creation of accounts:

a. Customers and sellers sign up for accounts.
b. Guest access facility provided for anonymous reviews.

Entry login page:

a. Customers and sellers have separate sign in pages.
b. Project related employees/admin will have separate sign-in as their access requirements are different.

Process reviews and verify the customer:

a. Check the credibility of the customers.
b. Classify the reviews according to its rating with the related keywords.
c. Fake reviews will be filtered regularly.

Display ratings and benchmarking:

a. The reviews are weighted against their positivity/negativity.
b. The customer's intention to purchase the product will also be collected.
c. The scores are updated as such and benchmarked against the competitor's similar products.

Non-functional requirements:

Operating Requirement:

a. The system will operate on all platforms.
b. The database would automatically log all the data entered.

Performance requirement:

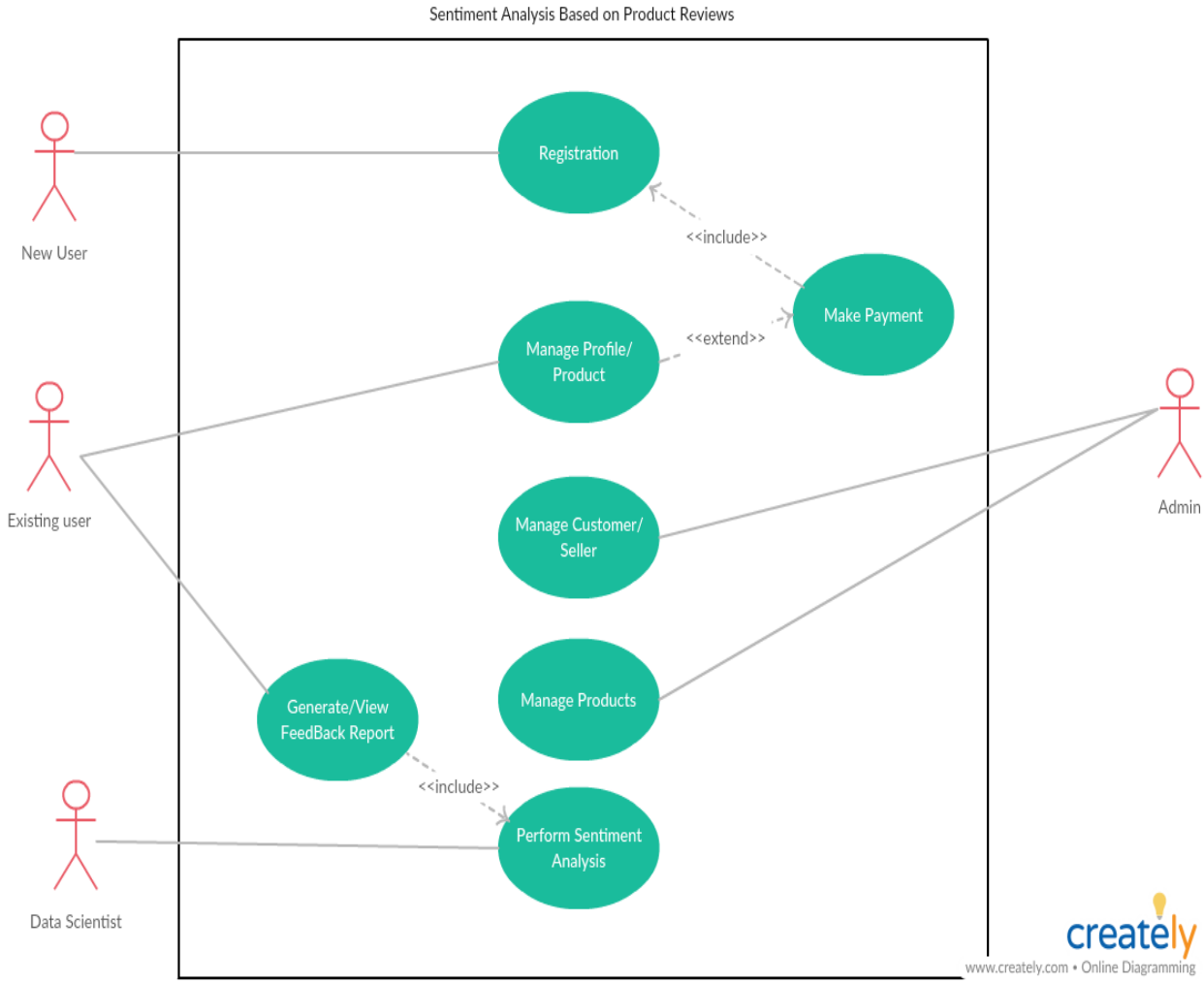
a. The rating of the product would be updated as soon as the reviews are being verified against the products.

Security requirement:

a. Only system administrators can access the pages and make amendments.
b. All the review related data and comments will be transparent – thus assuring customers with dependability.
c. All access is password protected.

Functional Models:

6.1 Use-Case Diagram:



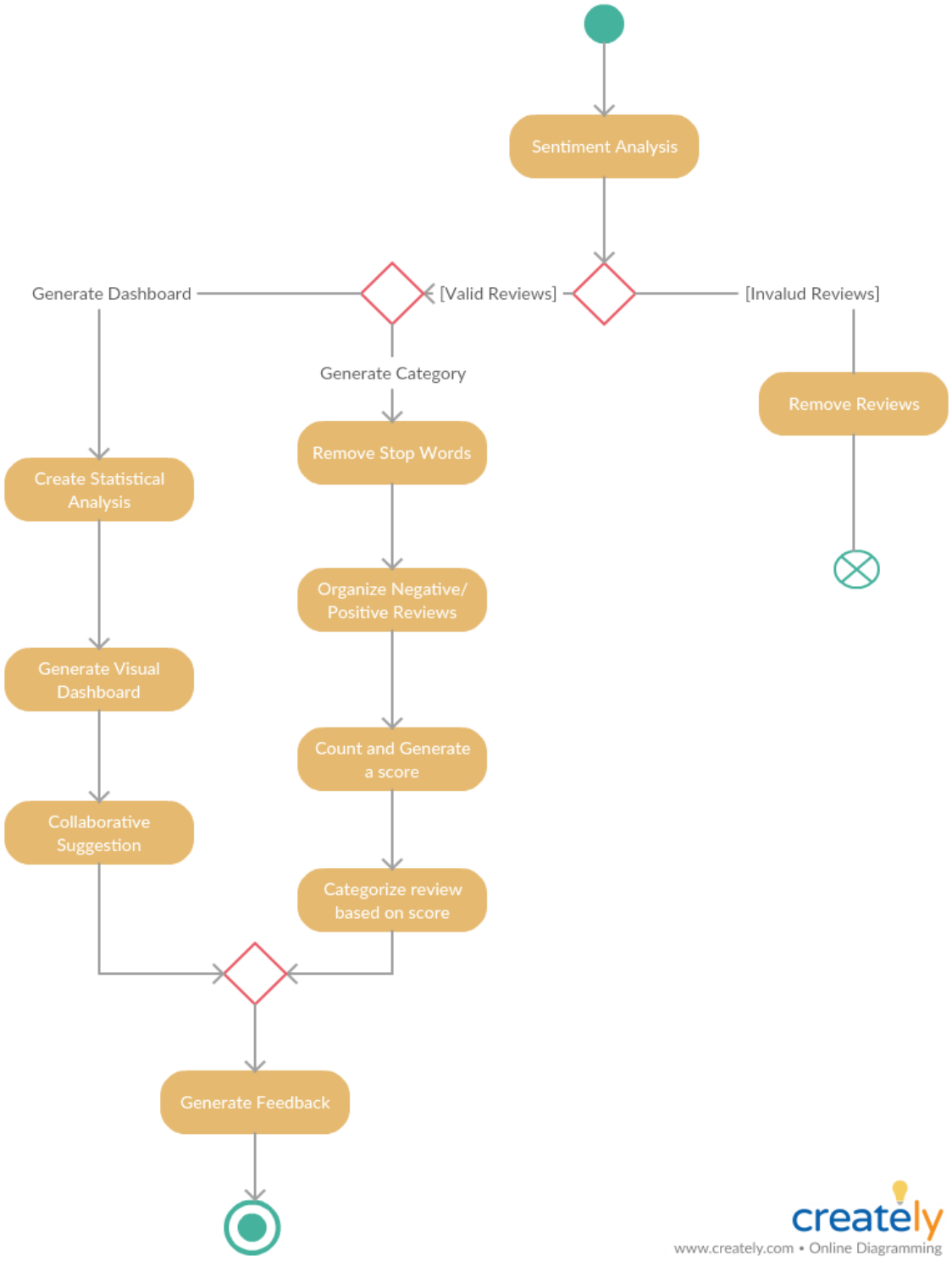
Use Case for performing Sentiment Analysis

Based on Product Reviews

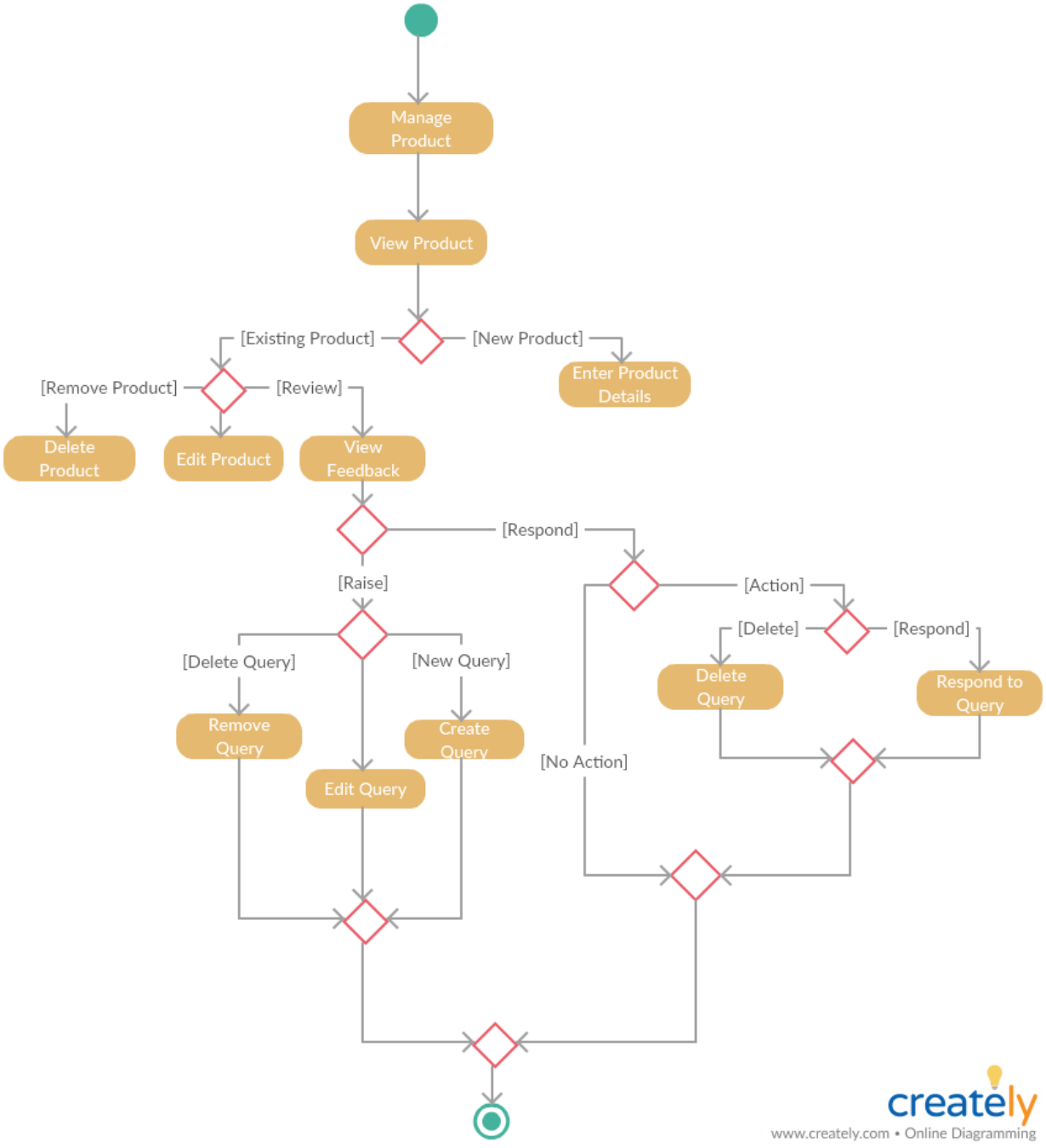
7. Activity Diagrams:

7.1 Activity Diagram 1:

Perform Sentiment Analysis

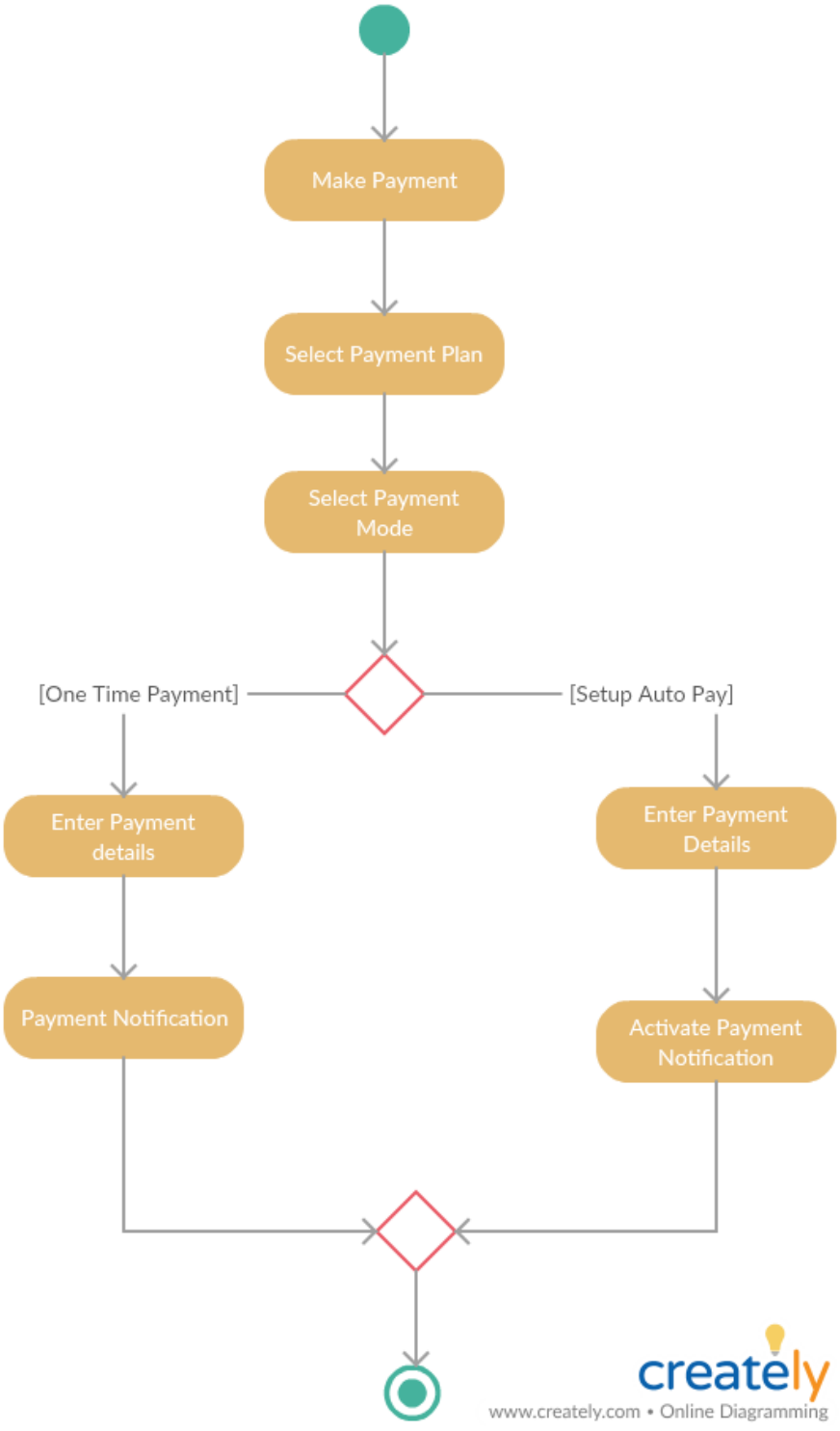


7.2 Activity Diagram 2: Manage Products



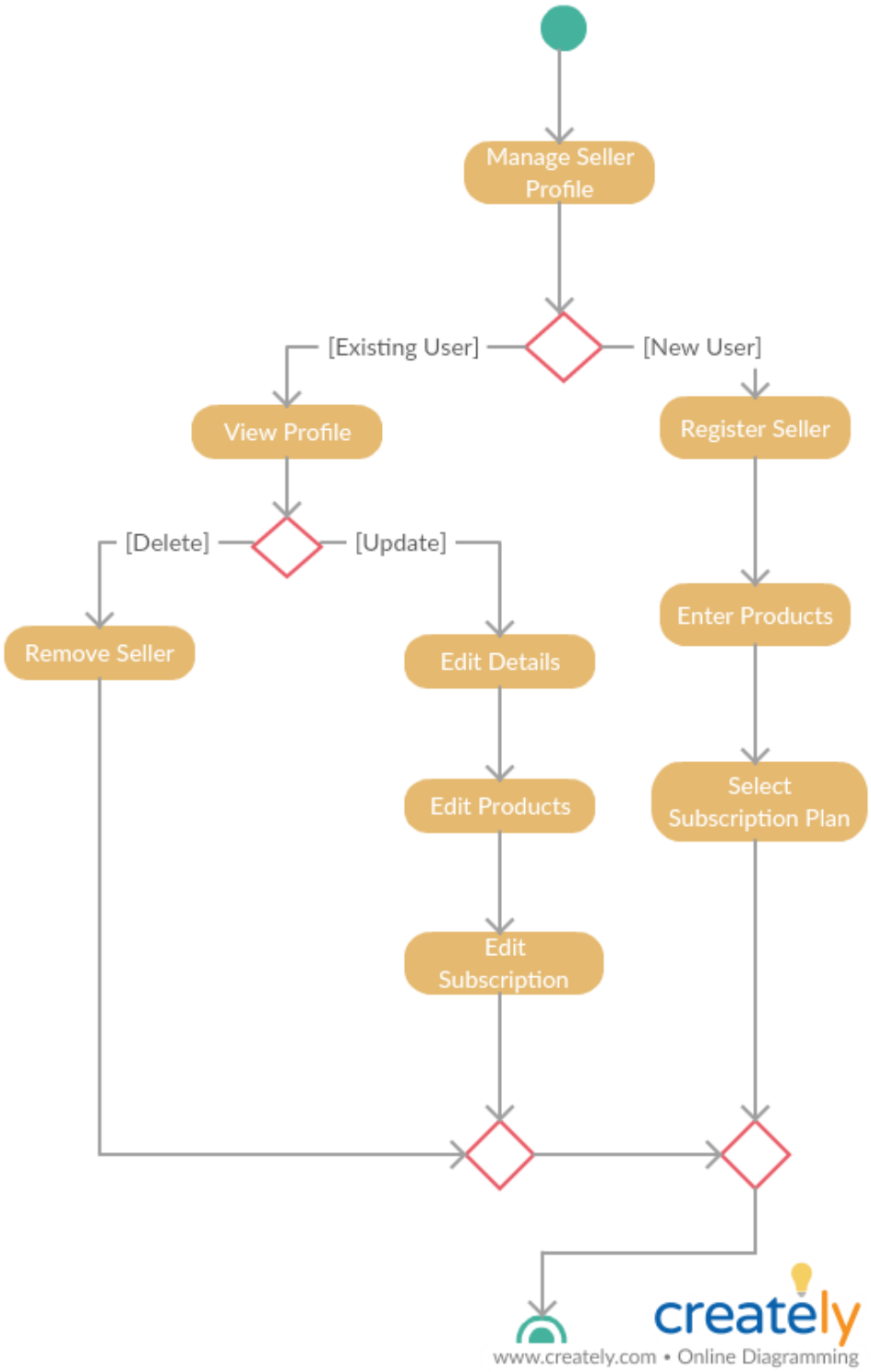
Activity Diagram for Manage Products

7.3 Activity Diagram 3: Make Payment



Activity Diagram for Make Payment

7.4 Activity Diagram 4: Manage Sellers



Activity Diagram for Manage Sellers

8. Use-Case Descriptions:

8.1 Perform Sentiment Analysis

Use Case Name: Perform Sentiment Analysis	ID: 001	Importance Level: High
Primary Actor: Data Scientist	Use Case Type: Detail, Essential	
Stakeholders and Interests: Seller: Who would benefit from the Sentiment Analysis Results. Data Scientist: Would execute and generate Analysis Report.		
Brief Description: This use case describes how a Data Scientist extracts useful feedback for the sellers.		
Trigger: When Sellers are looking for vital product Feedback. Type: External		
Relationships: Association: Data Scientist Include: Generate/View Feedback Report Extend: Generalization:		
Normal Flow of Events: 1. The algorithm would filter out the genuine review and remove the fake ones. 2. Data Scientist will remove stop words from the text feedback using NLP. 3. Data Scientist will aggregate the positive and negative keywords obtained from the feedback and organize the reviews. 4. Data Scientist will run a sophisticated algorithm to obtain an overall score. 5. Data Scientist will categorize the Product based on the aggregated scores.		
Sub Flows:		
Alternate/Exceptional Flows:		

8.2 Manage Product (For Admin)

Use Case Name: Manage Product	ID: 002	Importance Level: High
Primary Actor: Admin	Use Case Type: Detail, Essential	
Stakeholders and Interests: Seller: When seller is unable to update his own profile or products or has raised a query. Admin: When admin has to view/respond to the open queries or want to edit seller information.		
Brief Description: This use case describes how admin can manage the products and respond to queries raised by sellers.		
Trigger: When seller has raised a query or when the seller is unable to access their products. Type: External		
Relationships: Association: Admin Include: Extend: Generalization:		
Normal Flow of Events: 1. Admin will view all the active products. 2. Admin can edit existing product details. 3. Admin can view feedbacks posted by the sellers. 4. Admin will respond to the raised query.		
Sub Flows: 4b. Admin will not respond to the query and continue using the system.		
Alternate/Exceptional Flows: 1a. Admin will be able to add a new product into the system under a specific seller. 2a. Admin can remove an existing product. 4a. Admin will delete an existing query.		

8.3 Manage Product (For Existing User)

Use Case Name: Manage Product	ID: 003	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: Seller: When the seller desires to add new products or manage existing products or when the seller has a query towards the admin. Admin: Admin gets the queries raised by the sellers.		
Brief Description: This use case describes how Seller can manage the products and raise queries towards the admin.		
Trigger: When seller wants to add new products or manage existing products or has an issue or concern. Type: External		
Relationships: Association: Existing User Include: Extend: Make Payment Generalization:		
Normal Flow of Events: 1. Existing User will view all the active products. 2. Existing User can edit existing product details. 3. Existing User can view feedback responses posted by the admin. 4. Existing User will raise a new query.		
Sub Flows: 4b. Admin will not respond to the query and continue using the system.		
Alternate/Exceptional Flows: 1a. Existing User will be able to add a new product into the system under a specific seller. 2a. Existing User can remove an existing product. 4a. Existing User will edit raised query. 4b. Existing User will remove raised query.		

8.4 Generate/View Feedback Report

Use Case Name: Generate/View Feedback Report	ID: 004	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	

Stakeholders and Interests:

Seller: When the seller desires to get vital analytical report about the product based on customer reviews.

Brief Description: This use case describes how the feedback dashboard is generated for the sellers.

Trigger: When seller wants to view a personalized feedback for their product based on customer reviews.

Type: External

Relationships:

Association: Existing User

Include:

Extend:

Generalization:

Normal Flow of Events:

1. Existing User will view all the active products.
2. Existing User can edit existing product details.
3. Existing User can view feedback responses posted by the admin.
4. Existing User will raise a new query.

Sub Flows:

4b Admin will not respond to the query and continue using the system.

Alternate/Exceptional Flows:

1a. Existing User will be able to add a new product into the system under a specific seller.

2a. Existing User can remove an existing product.

4a. Existing User will edit raised query.

4b. Existing User will remove raised query.

8.5 Perform Registration

Use Case Name: Registration	ID: 005	Importance Level: High
Primary Actor: New User	Use Case Type: Detail, Essential	
Stakeholders and Interests: Seller: New Seller wants to register to the system to avail the facilities.		
Brief Description: This use case describes how a new first-time seller can register themselves into the system.		
Trigger: When seller is new to the system and feels the need to create a profile. Type: External		
Relationships: Association: New User Include: Make Payment Extend: Generalization:		
Normal Flow of Events: 1. New User will create a new profile with the user details. 2. New User will enter the product details. 3. New User will be able to select subscription plan based on the facilities the user wants to avail.		
Sub Flows:		
Alternate/Exceptional Flows:		

8.6 Manage Sellers

Use Case Name: Manage Sellers	ID: 006	Importance Level: High
Primary Actor: Admin	Use Case Type: Detail, Essential	
Stakeholders and Interests: Seller: Seller is unable to update their profile and requires support of the admin. Admin: When admin requires to view or update the sellers' profiles.		
Brief Description: This use case describes admin can view/edit the profiles of the sellers.		
Trigger: When admin requires to view all the sellers or gets a request from the seller to update some profile information. Type: External		
Relationships: Association: Admin Include: Extend: Generalization:		
Normal Flow of Events: 1. Admin will be able to view the profile of all the registered sellers. 2. Admin can update the seller profile details. 3. Admin can update the seller products. 4. Admin can update the seller subscription.		
Sub Flows:		
Alternate/Exceptional Flows: 2a. Admin can remove a seller from the systems completely.		

8.7 Make Payment

Use Case Name: Make Payment	ID: 007	Importance Level: High
Primary Actor: New User, Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: Seller: New User during registration will require to make a payment or existing user during renewals.		
Brief Description: This use case describes how new or existing user would be able to make payment towards their subscription.		
Trigger: When a new user signs up or when an existing user has to renew their subscription, they are required to make a payment. Type: External		
Relationships: Association: Include: Extend: Generalization:		
Normal Flow of Events: 1. User will be able to select their payment plan as monthly or yearly. 2. Users will be prompted for the mode of payment of their choice. 3. User can opt for one-time payment and enter Payment details. 4. User can opt for payment notification and provide details regarding the same.		
Sub Flows:		
Alternate/Exceptional Flows: 3a. User can opt for Auto Pay as part of their payment plan. 3b. User will Enter the payment details for which the auto pay is required to be set up. 3c. User will activate payment notification for each payment made towards the organization.		

9. Effort Estimation:

Unadjusted Actor Weighting Table:				
Actor Type	**Description**	**Weight**	**Number**	**Weighted Value**
Simple	External System with well-defined API	1	3	3
Average	External System using a protocol-based interface, eg,. HTTP,TCP/IP or a data!	2	2	4
Complex	Human	3	4	12
			Unadjusted Actor Weight Total (UAW)	19

Unadjusted Use-Case Weighting Table:

Use Case Type	**Description**	**Weight**	**Number**	**Weighted Value**
Simple	1-3 transactions	5	5	25
Average	4-7 transcations	10	3	30
Complex	>7 transactions	15	2	30
			Unadjusted Use Case Weight Total (UUCW)	85
			Unadjusted Use-Case Points (UUCP) = UAW + UUCW	104

Technical Complexity Factors

Factor Num	Description	Weight	Assigned Value	Weighted Va	Notes
T1	Distributed System	2.0	1	2	
T2	Reposnse time or throughout performance objectives	1.0	2	2	
T3	End-user online efficiency	1.0	4	4	
T4	Complex internal processing	1.0	4	4	
T5	Reusability of code	1.0	5	5	
T6	Easy to install	0.5	2	1	
T7	Ease of use	0.5	2	1	
T8	Portability	2.0	3	6	
T9	Ease of change	1.0	1	1	
T10	Concurrency	1.0	1	1	
T11	Special security objectives included	1.0	5	5	
T12	Direct access for the third parties	1.0	3	3	
T13	Special user training required	1.0	2	2	
			Technical Factor Value (Tfactor)	37	
		Techincal Complexity Factor (TCF) = 0.6 + (0.01 * Tfactor)		0.97	

Enviornmental Factors

Factor Num	Description	Weight	Assigned Value	Weighted Va	Notes
E1	Familiarity with system development process being used	1.5	4	6	
E2	Application experience	0.5	4	2	
E3	Object-oriented experience	1.0	5	5	
E4	Lead analyst capability	0.5	5	2.5	
E5	Motivation	1.0	3	3	
E6	Requirements stability	2.0	4	8	
E7	Part time staff	-1.0	1	-1	
E8	Difficulty of programming language	-1.0	4	-4	
			Enviornmental Factor Value (Efactor)	21.5	
	Enviornmental Factor (EF) = 1.4 + (-0.03 * Efactor)	0.755			
	Assigned Use Case Points (UCP) = UUCP * TCF * ECF	76.16			
	PHM	20			
	Effort in Personal Hours = UCP * PHM	1523			

CRC Cards:

Front:

Class Name: New User	ID: 1	Type: Concrete, Domain
Description: An individual or organization in general who wants to register to the system to avail its facilities.		Associated Use Cases: 3
Responsibilities	Collaborators	
Register	System	
Make Payment	System	
Manage Product/ Profile	System	

Back:

Attributes: Name(Text) Age(Integer) Gender (Text)
Relationships: Generalization (a-kind-of): Aggregation (has-parts): Other Associations: Existing Customer, System

Front:

Class Name: Existing User	ID: 2	Type: Concrete, Domain
Description: An individual in general who pays for services using the system		Associated Use Cases: 3
Responsibilities	Collaborators	
Manage Profile/ Product	System	
Make Payment	System	
Generate/ View Feedback Report	System	

Back:

Attributes: Name(Text) Age(Integer) Gender (Text)
Relationships: Generalization (a-kind-of): Aggregation (has-parts): Other Associations: New User, System

Front:

Class Name: Admin	ID: 3	Type: Concrete, Domain
Description: An individual in general who manages products and users using the system		Associated Use Cases: 2
Responsibilities Manage Products Manage Customers/ Sellers	Collaborators New, Existing User New, Existing User	

Back:

Attributes: AdminName(Text) ProductID(Integer) UserName (Text)
Relationships: Generalization (a-kind-of): Aggregation (has-parts): Other Associations: Existing User, New User, System

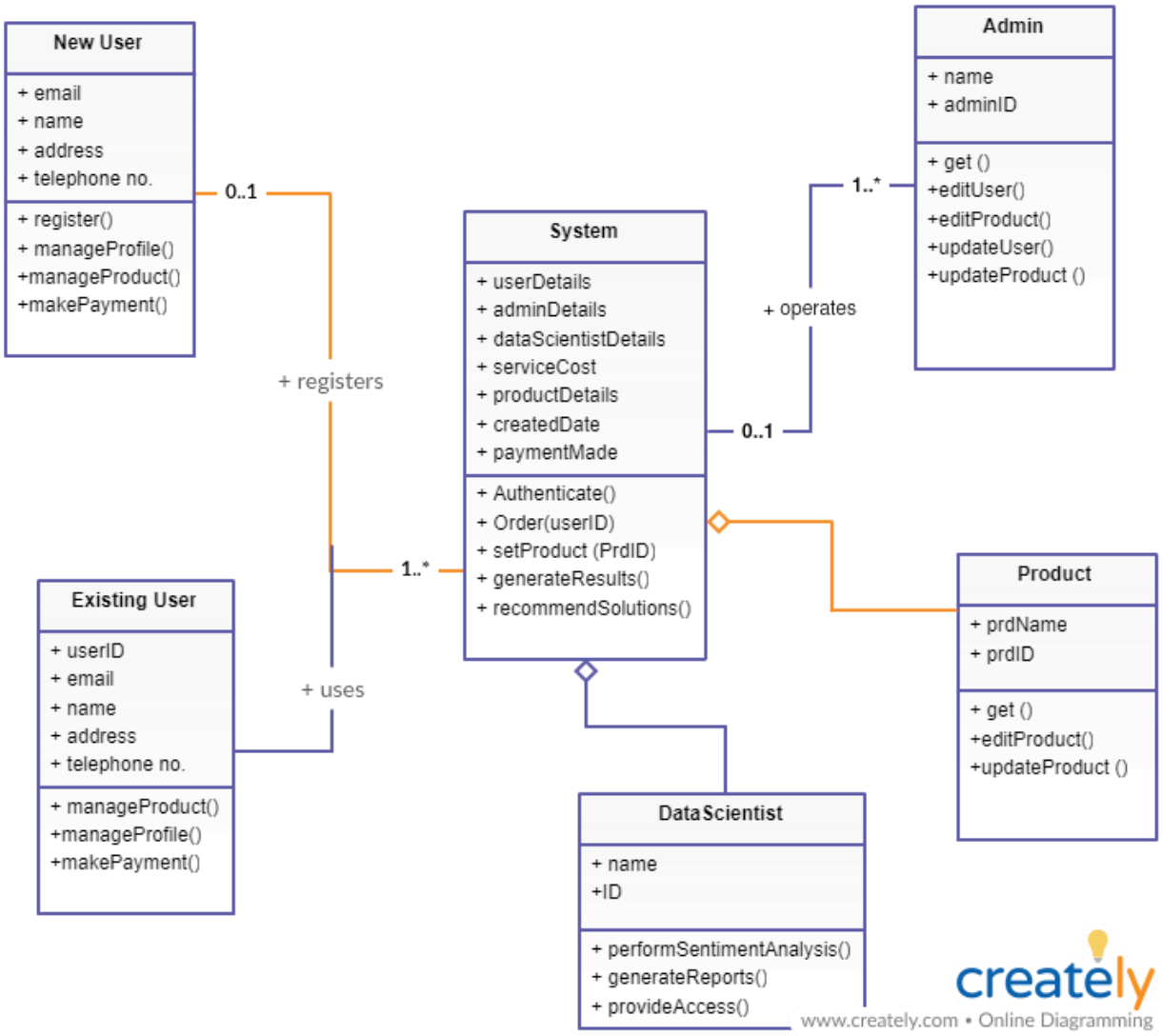
Front:

Class Name: Data Scientist	ID: 4	Type: Concrete, Domain
Description: An individual in general who performs sentiment analysis for users		Associated Use Cases: 2
Responsibilities Performs Sentiment Analysis Uploads Analysis Results	Collaborators Existing User Existing User	

Back:

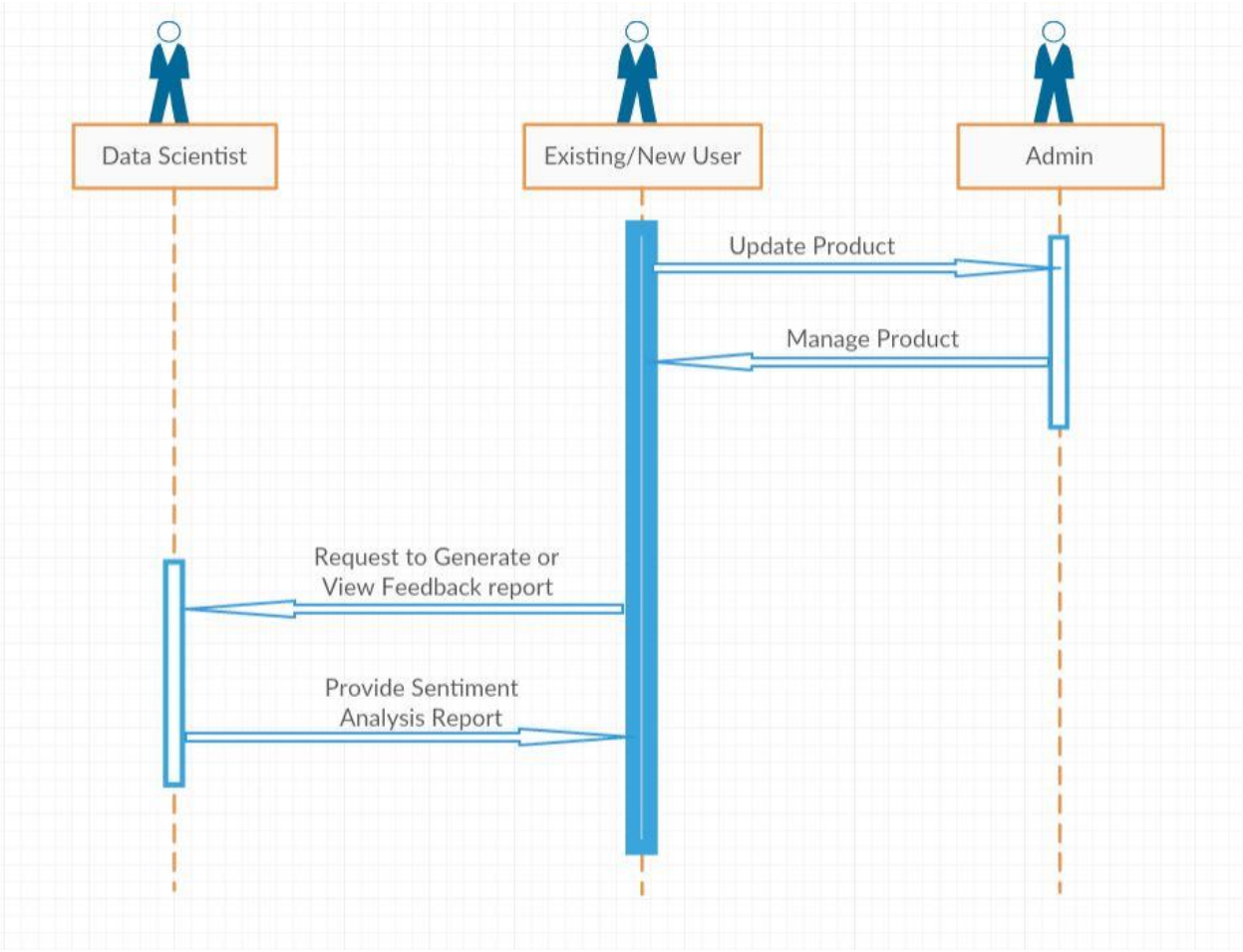
Attributes: UserName(Text) DataScientistName(Text) Results(Text)
Relationships: Generalization (a-kind-of): Aggregation (has-parts): Other Associations: Existing User, System

CLASS DIAGRAM

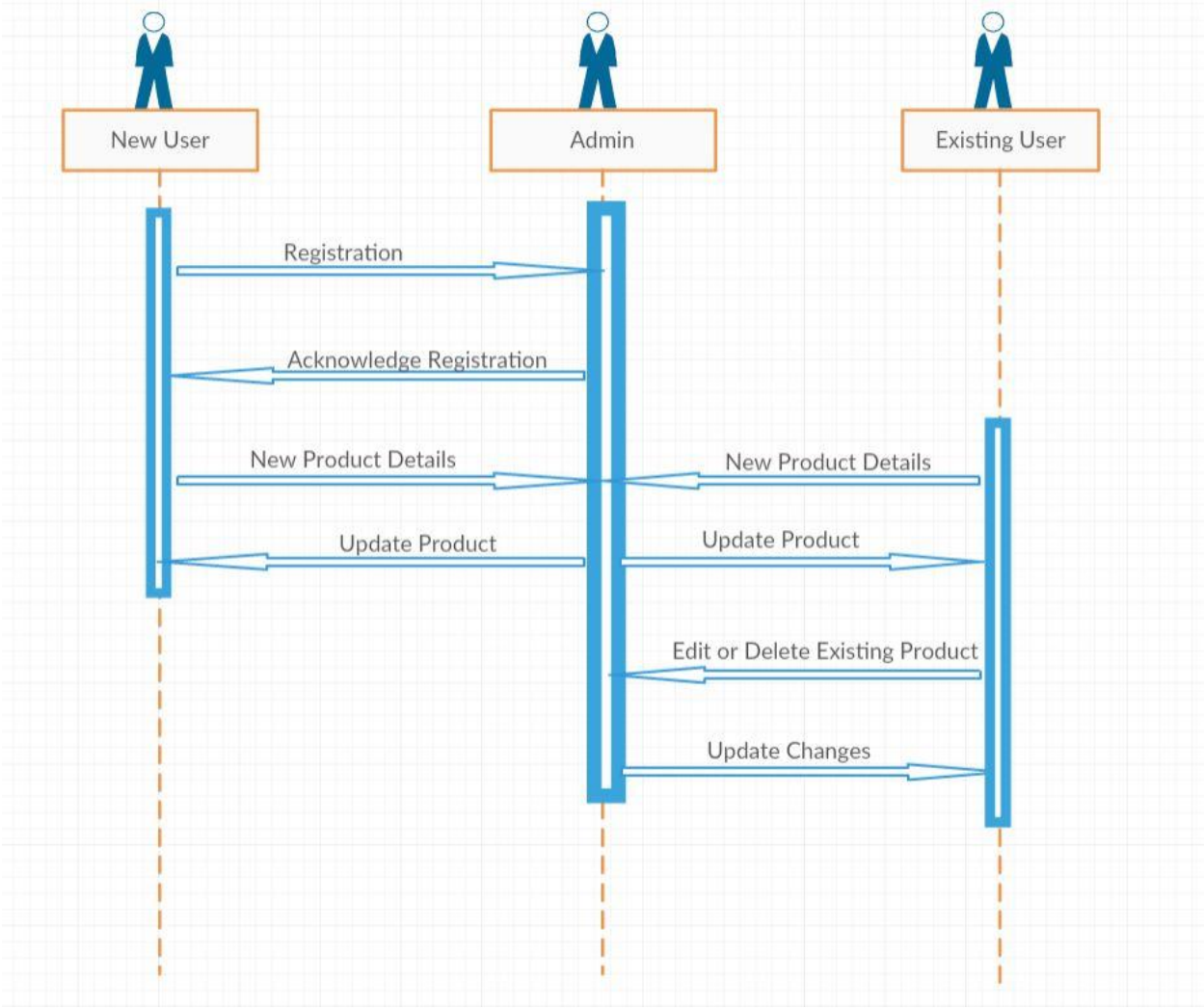


Sequence Diagram

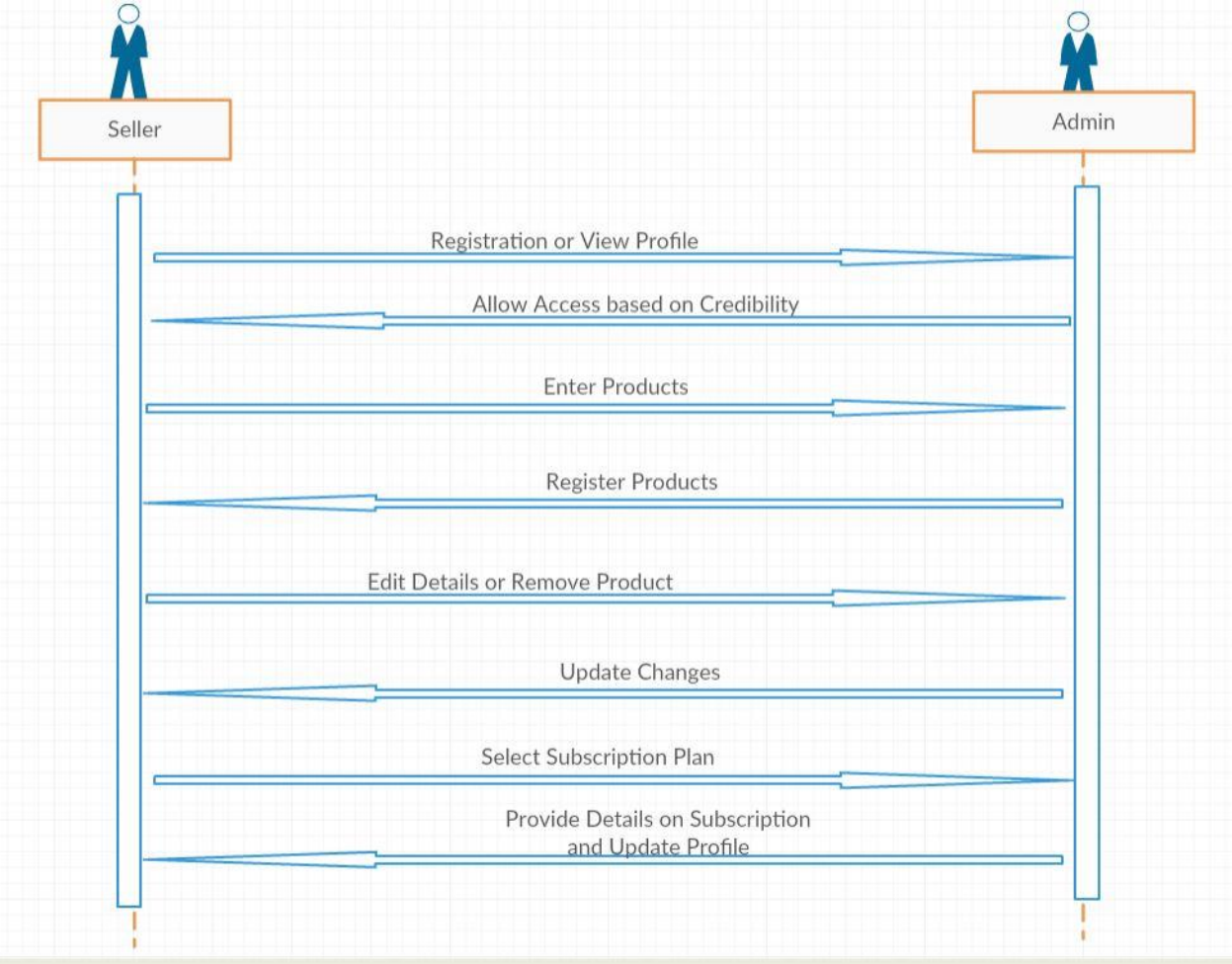
Sentiment Analysis



Users and Admin

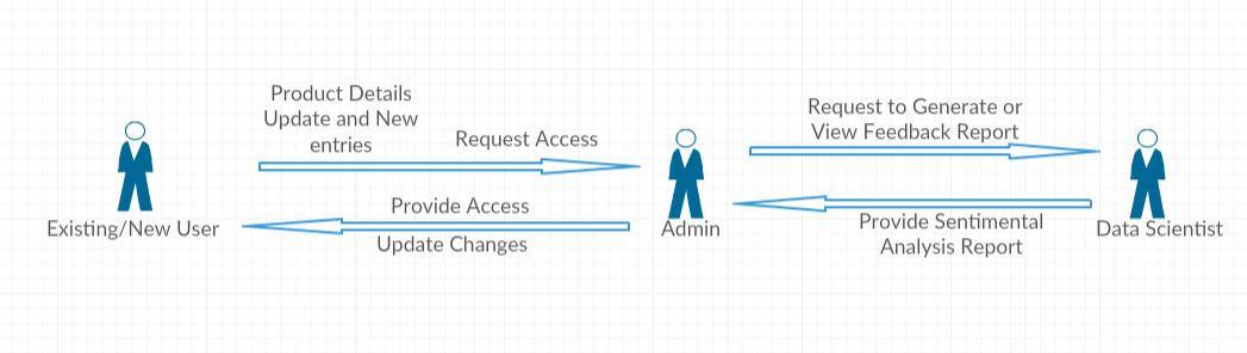


Seller and Admin

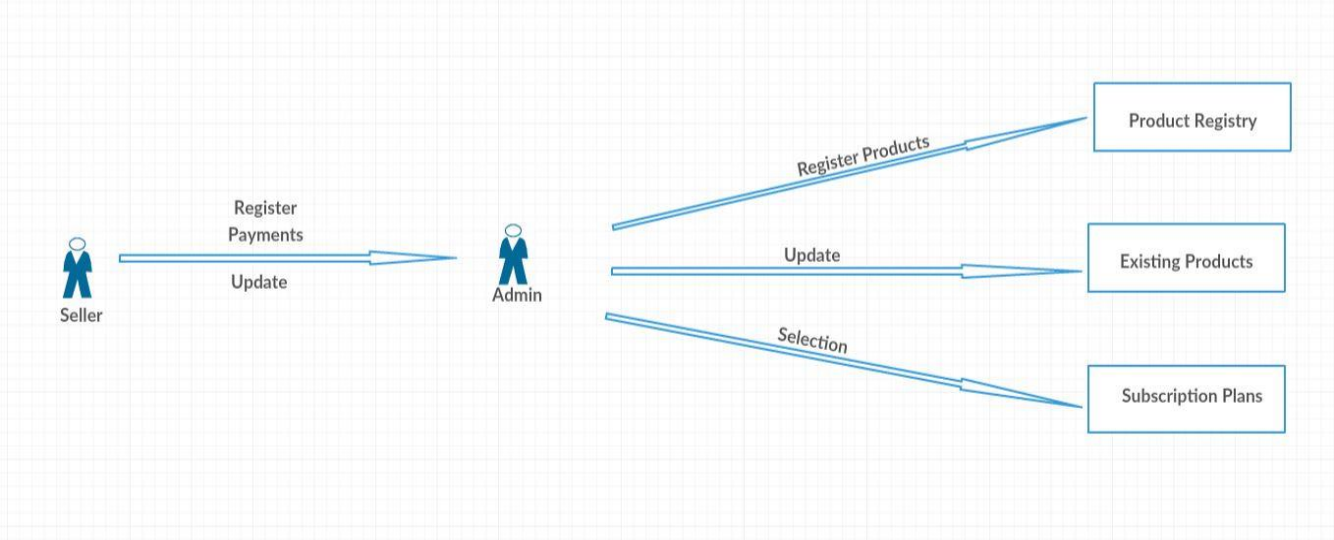


Communication Diagrams

Admin and Users



Seller and Admin

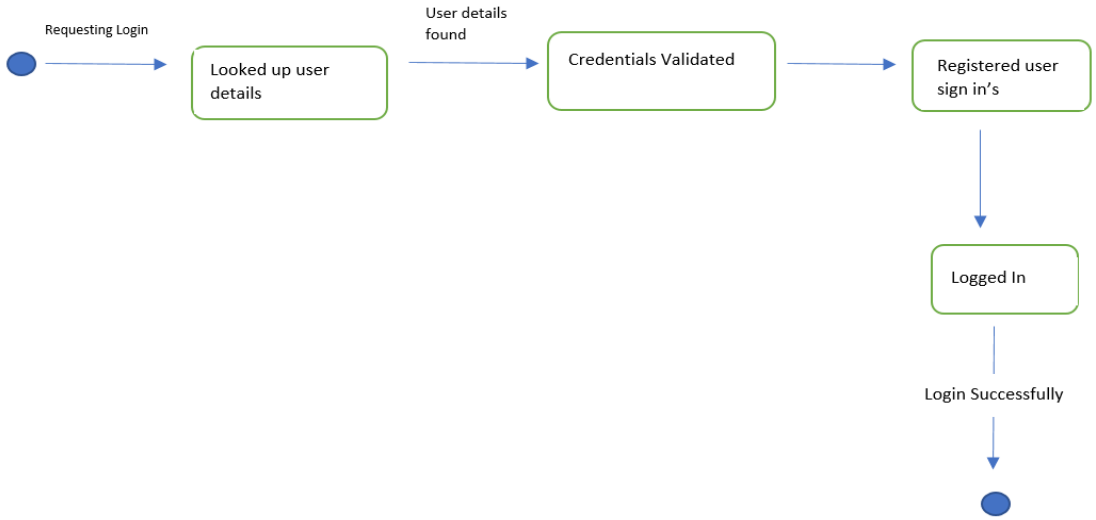


CRUDE Matrices:

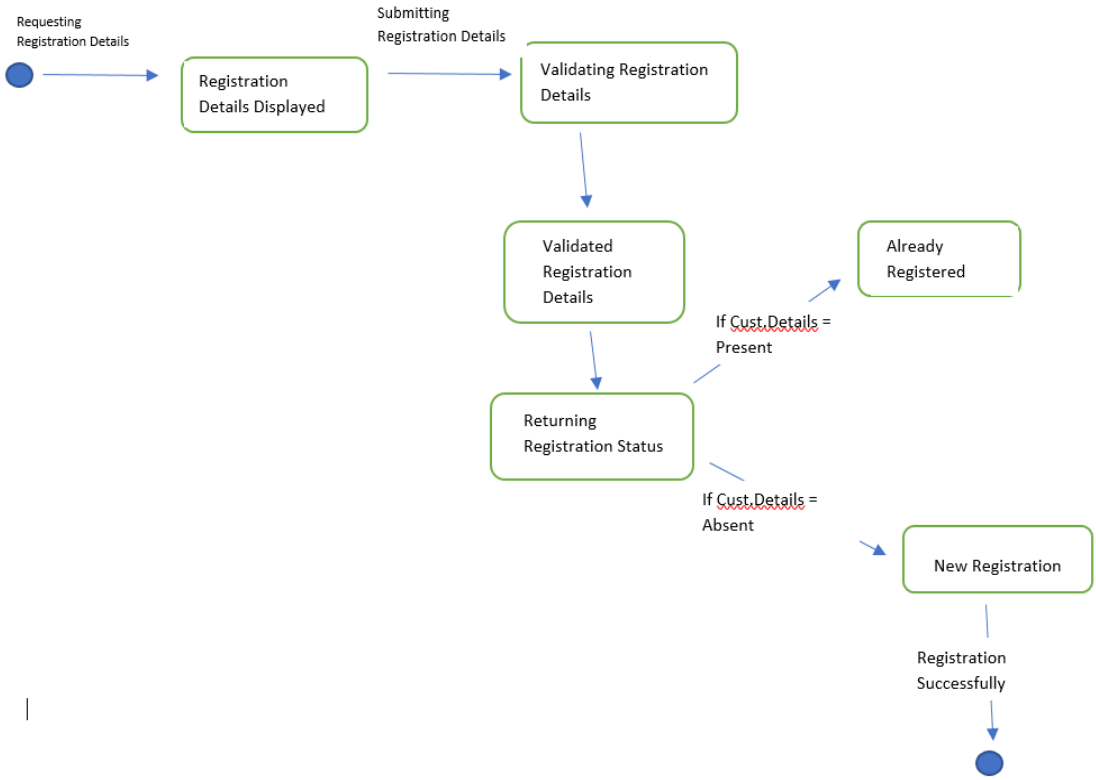
	Data Scientist Actor	User Actor	Admin Actor	Seller Actor	Manage Product Status	Analysis Report
Data Scientist Actor			E		UDE	UDE
User Actor		R			CRUD	R
Admin Actor	CDUE	CDUE		E	CRUDE	CRUDE
Seller Actor					R	R
Manage Product Status						
Analysis Report						

Behavioral State Diagrams:

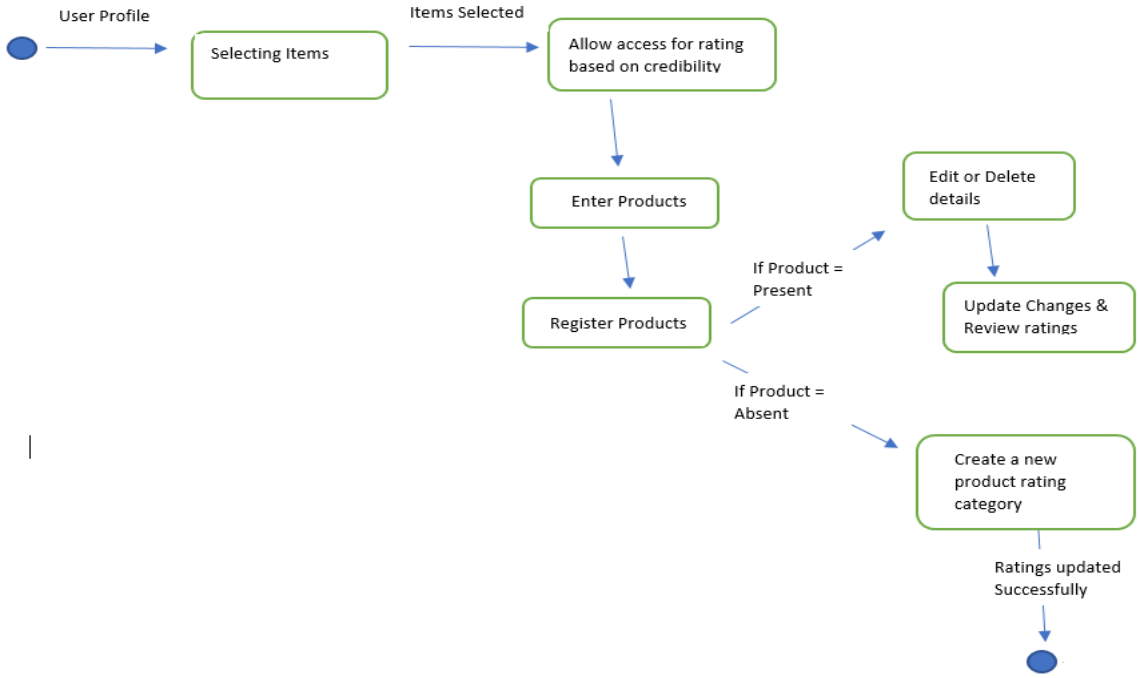
Login:



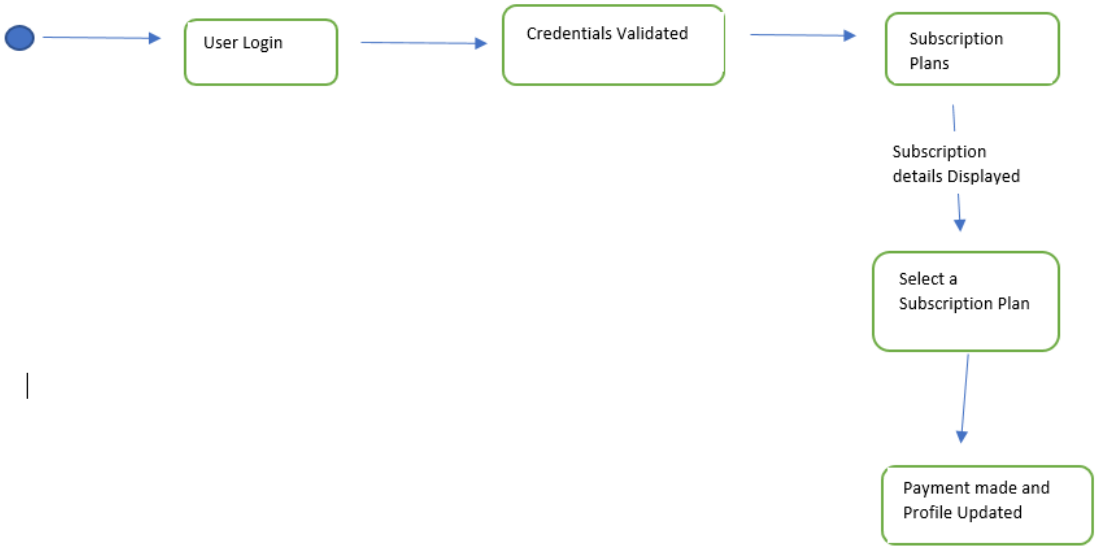
User Registration:



Ratings & Reviews:



Subscription:



APPENDICES:
Results of verification and validation of functional model, structural and behavioral models performed by your partner user group on your deliverable.

Verification and Validation Of Functional Model

1) **At least one event recorded in the normal flow of events, subflows, or alternative/exceptional flows of the use-case description for each activity or action that is included on an activity diagram, and each event should be associated with an activity or action.**

Activity diagram	**Missing events in use-case description**
Perform Sentiment Analysis	Create Statistical Analysis, generate Visual Dashboard, Collaborative Suggestion, Remove Reviews
Manage Products (Admin) Manage Products (User)	Enter Product Details
Manage Sellers	Register Seller, Enter Products, Select Subscription plan

2) All objects portrayed as an object node in an activity diagram must be mentioned in an event in the normal flow of events, sub-flows, or alternative/exceptional flows of the use-case description: Table in 1) above describes all the object nodes that are not mentioned in the use-case description

3) Sequential order of the events in a use-case description should occur in the same sequential order of the activities contained in an activity diagram: Refer Table in 1) which talks about missing objects that result in sequential order not being followed.

4) When comparing a use-case description to a use-case diagram, there must be one and only one use-case description for each use case, and vice versa: The number of use-cases in the use-case diagram matches the number of use-case descriptions and vice-versa

5) All actors listed in a use-case description must be portrayed on the use-case diagram: The actors Data Scientist, New user, Existing user and Admin are all listed in the use-case descriptions. They are portrayed on the use-case diagram too.

6) Include the stakeholders listed in the use-case description as actors in the use-case diagram: *Done*

7) All other relationships listed in a use-case description (include, extend, and generalization) must be portrayed on a use-case diagram: All other relationships listed in the use-case description are portrayed successfully in the use-case diagrams

8) Diagram-specific requirements that must be enforced: A control flow is missing in the activity diagram 'Perform Sentiment Analysis'. *Done*

Verification and Validation Of Structural Model

First, every CRC card should be associated with a class on the class diagram, and vice versa.
Done

Second, the responsibilities listed on the front of the CRC card must be included as operations in a class on a class diagram, and vice versa.
Done

Third, collaborators on the front of the CRC card imply some type of relationship on the back of the CRC card and some type of association that is connected to the associated class on the class diagram.
Done

Fourth, attributes listed on the back of the CRC card must be included as attributes in a class on a class diagram, and vice versa.
Done

Fifth, the object type of the attributes listed on the back of the CRC card and with the attributes in the attribute list of the class on a class diagram implies an association from the class to the class of the object type.
Done

Sixth, the relationships included on the back of the CRC card must be portrayed using the appropriate notation on the class diagram.
Done

Seventh, an association class, should be created only if there is indeed some unique characteristic (attribute, operation, or relationship) about the intersection of the connecting classes.
Done

Verification and Validation Of Behavioral Model

First, every actor and object included on a sequence diagram must be included as an actor and an object on a communication diagram, and vice versa.
Done

Second, if there is a message on the sequence diagram, there must be an association on the communications diagram, and vice versa.
Done

Third, every message that is included on a sequence diagram must appear as a message on an association in the corresponding communication diagram, and vice versa.
Done

Fourth, if a guard condition appears on a message in the sequence diagram, there must be an equivalent guard condition on the corresponding communication diagram, and vice versa.

Fifth, the sequence number included as part of a message label in a communications diagram implies the sequential order in which the message will be sent.
Done

Sixth, all transitions contained in a behavior state machine must be associated with a message being sent on a sequence and communication diagram, and it must be classified as a (C)reate, (U)pdate, or (D)elete message in a CRUDE matrix.
Done

Seventh, all entries in a CRUDE matrix imply a message being sent from an actor or object to another actor or object.
Done

Results of balancing across the three different models performed by your partner user group on your deliverable. (Total – 3 such verifications)

Verification and Validation of Analysis Models

Balancing Functional and Structural Models

First, every class on a class diagram and every CRC card must be associated with at least one use case, and vice versa.
Done

Second, every activity or action contained in an activity diagram and every event contained in a use-case description should be related to one or more responsibilities on a CRC card and one or more operations in a class on a class diagram and vice versa.

Done

Third, every object node on an activity diagram must be associated with an instance of a class on a class diagram (i.e., an object) and a CRC card or an attribute contained in a class and on a CRC card.
Done

Fourth, every attribute and association/aggregation relationships contained on a CRC card (and connected to a class on a class diagram) should be related to the subject or object of an event in a use-case description.
Done

Balancing Functional and Behavioral Models

First, the sequence and communication diagrams must be associated with a use case on the use-case diagram and a use-case description.
Done

Second, actors on sequence diagrams, communication diagrams, and/or CRUDE matrices must be associated with actors on the use-case diagram or referenced in the use case description, and vice versa.
Done

Third, messages on sequence and communication diagrams, transitions on behavioral state machines, and entries in a CRUDE matrix must be related to activities and actions on an activity diagram and events listed in a use-case description, and vice versa.
Done

Fourth, all complex objects represented by an object node in an activity diagram must have a behavioral state machine that represents the object's lifecycle, and vice versa.
Done

Balancing Structural and Behavioral Models

First, objects that appear in a CRUDE matrix must be associated with classes that are represented by CRC cards and appear on the class diagram, and vice versa.
Done

Second, because behavioral state machines represent the life cycle of complex objects, they must be associated with instances (objects) of classes on a class diagram and with a CRC card that represents the class of the instance.
Done

Third, communication and sequence diagrams contain objects that must be an instantiation of a class that is represented by a CRC card and is located on a class diagram.
Done

Fourth, messages contained on the sequence and communication diagrams, transitions on behavioral state machines, and cell entries on a CRUDE matrix must be associated with responsibilities and associations on CRC cards and operations in classes and associations connected to the classes on class diagrams.
Done

Fifth, the states in a behavioral state machine must be associated with different values of an attribute or set of attributes that describe an object.
Done