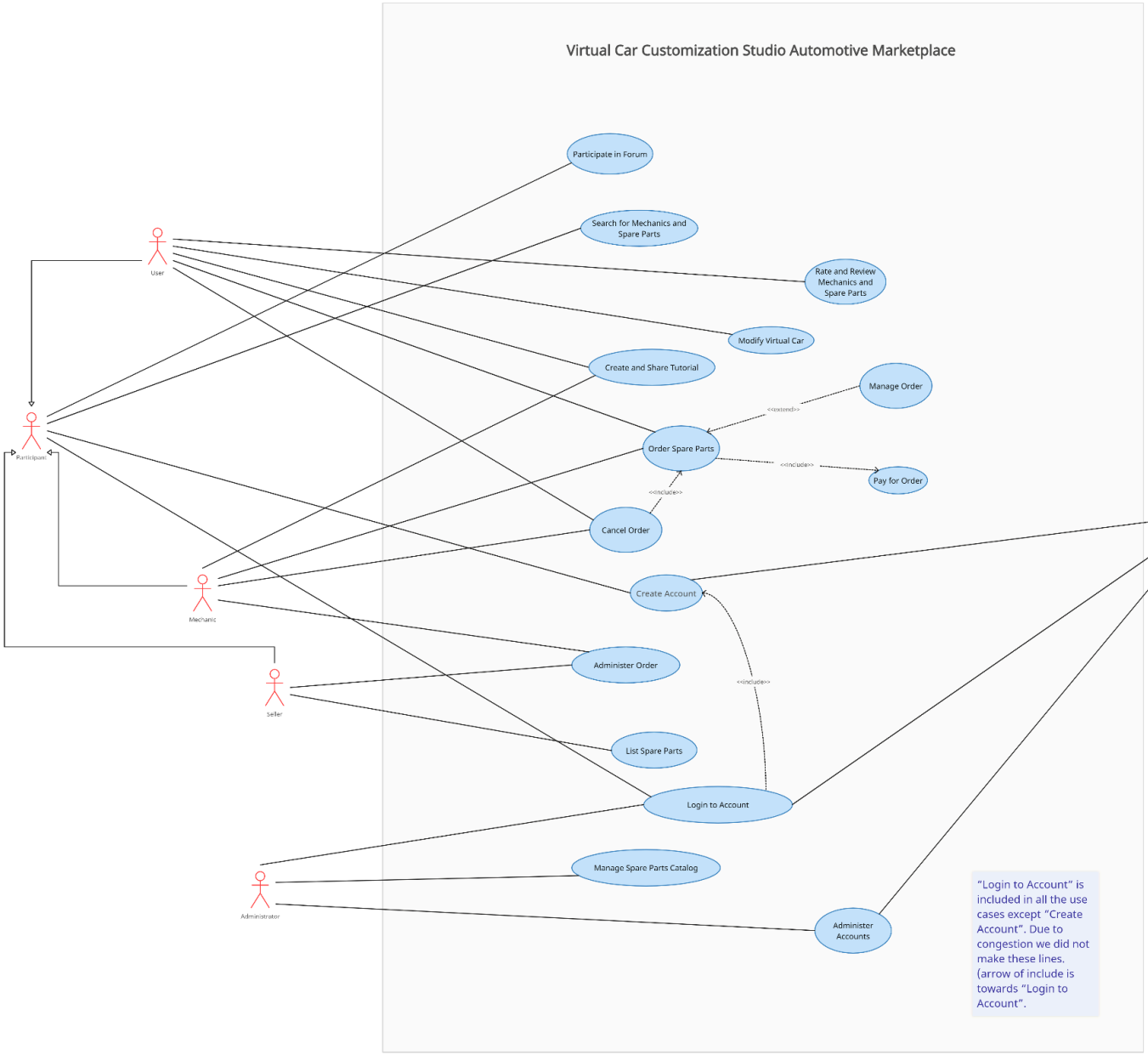
Project Deliverable-1

Virtual Car Customization Studio and Automotive Marketplace

Group Members (Section 5F-Group # 4):

- **Raabia Baig (21L-1831)**
- **Ariba Arshad (21L-5381)**
- **Rafia Karim (21L-5192)**
- **Khadeeja Wasif (21L-5177)**



Use Case Diagram

Use Case Descriptions

Use Case #1: Search for Mechanics and Spare Parts

Identifier	UC001
Name	Search for Mechanics and Spare Parts
Purpose	This use case describes the procedure to search for trusted mechanics and available spare parts.
Priority	High
Actors	User, Mechanic, Seller
Pre-conditions	1. The account should exist. 2. The account should be logged in. 3. The input data should have been entered.
Post-conditions	The searched mechanic or spare part will be displayed.
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Enter the mechanic's name or spare part to be searched	1.1	Display the list of matched mechanics or spare parts

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Enter the mechanic's name or spare part to be searched	1.1	Display "Not Found" Error
Exceptions: The system displays an error message and suggests trying again if the user, mechanic or seller has not logged in, tries to search with incomplete or missing information, or technical issues or system errors occur that prevent the search for mechanics or spare parts.			

Use Case #2: Order Spare Parts

Identifier	UC002
Name	Order Spare Parts
Purpose	This use case describes the procedure to place an order for spare parts from the marketplace.
Priority	High
Actors	User, Mechanic
Pre-conditions	1. The account should exist. 2. The account should be logged in. 3. The spare parts should be in stock. 4. The spare parts should have been added to the cart. 5. The payment method should be selected.

Post-conditions	1. “Order has been placed” message will be shown. 2. The system will be updated.
Dependencies	UC003, UC012, UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Select spare parts from the catalog.	1.1	The spare parts are added to the cart.
2	Select the payment method.	2.1	Selected payment method is displayed.
3	Click “Place Order”.	3.1	Confirmation message is displayed and system is updated.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Click “Back” button	1.1	Displays previous page
2	Decide not to place the order.	2.1	The system retains its state.
3	Enter payment information	3.1	Payment failure message is displayed

Exceptions: The system displays an error message and suggests trying again if the user or mechanic has not logged in, tries to place order with incomplete or missing information, or technical issues or system errors occur that prevent placement of order for the spare parts.

Use Case #3: Manage Order

Identifier	UC003
Name	Manage Order
Purpose	This use case describes the procedure to track and manage spare part orders.
Priority	High
Actors	User, Mechanic
Pre-conditions	1. The account should exist. 2. The account should be logged in. 3. The spare part(s) should have been added to the cart.
Post-conditions	None
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**

1	Select spare part from the catalog.	1.1	The spare part is added to the cart.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Remove spare part from the cart.	1.1	The spare part is removed from the cart.
2	Click “Back” button.	2.1	Display previous page
Exceptions: The system displays an error message and suggests trying again if the user or mechanic has not logged in, or technical issues or system errors occur that prevent order management.			

Use Case #4: Rate and Review

Identifier	UC004
Name	Rate and Review Mechanics and Spare Parts
Purpose	This use case describes the procedure to provide ratings and reviews.
Priority	Medium
Actors	User

Pre-conditions	1. The account should exist. 2. The account should be logged in. 3. The spare part or mechanic to be rated and reviewed should be selected.
Post-conditions	The rating and review will be posted.
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Rate and review a spare part or mechanic.	1.1	Store the user's input.
2	Submit the rating and review.	2.1	The rate and review section is updated.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Cancel the rating and review.	1.1	Discard the rating and review.
Exceptions: The system displays an error message and suggests trying again if the user has not logged in, information is incomplete or missing, or technical issues or system errors occur that prevent rating and reviewing.			

Use Case #5: Cancel Order

Identifier	UC005
Name	Cancel Order
Purpose	Allow users to cancel an existing order for spare parts or services under certain conditions.
Priority	Medium
Actors	User, Mechanic
Pre-conditions	1. The account should exist. 2. The account should be logged in. 3. The spare parts should be in stock. 4. The spare parts should have been added to the cart.
Post-conditions	1. “Order has been canceled” message will be shown. 2. The system will be updated.
Dependencies	UC002, UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Access list of orders.	1.1	The system displays a list of the user's orders with the option to cancel.

2	Select the specific order to cancel.	2.1	The system presents details of the selected order including order status and items.
3	Confirm the cancellation of the order.	3.1	Confirmation message is displayed and the system is updated.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Click “Back” button	1.1	Displays previous page
2	Decide not to cancel the order.	2.1	The system retains the order status and details.
Exceptions: The system displays an error message and suggests trying again if the user or mechanic has not logged in, tries to cancel order with incomplete or missing information, or technical issues or system errors occur that prevent canceling order for the spare parts.			

Use Case #6: Modify Virtual Car

Identifier	UC006
Name	Modify Virtual Car
Purpose	Allow Users to customize virtual cars with different modifications
Priority	High
Actors	User
Pre-conditions	1. The user must be logged into their account. 2. The user has selected a virtual car for modification.

Post-conditions	1. The user's selected virtual car is customized with the chosen modifications. 2. The user can order the chosen modifications from the sellers
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	The User selects a virtual car for modification.	1.1	The system presents customization options, including paint, rims, body kits, and performance upgrades.
2	The User chooses modification options and makes selections.	2.1	The system applies the selected modifications to the virtual car in real-time, providing a visual representation.
3	The User reviews the modifications and confirms.	3.1	The system saves the customized virtual car to the user's profile.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	The User cancels the modification process.	1.1	The system discards the changes, returning to the previous state.
Exceptions: The system displays an error message and suggests trying again if the user has not logged in or if technical issues prevent the modifications.			

Use Case #7: Participate in Forums

Identifier	UC007
Name	Participate in Forums

Purpose	Allow Participants to participate in discussions on the community forum.
Priority	Medium
Actors	User, Seller, Mechanic
Pre-conditions	1. Account should exist. 2. Account must be logged in.
Post-conditions	None
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Navigates to the forum section of the platform	1.1	The system displays a list of available forum topics and discussions.
2	Selects a forum topic or starts a new discussion.	2.1	The system presents a text editor for the user to compose their post or comment.
3	Writes and submits their post or comment	3.1	The system posts the user's contribution to the forum, and it becomes visible to others.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Edits or deletes their forum post or comment	1.1	The system updates or removes the post accordingly
Exceptions: If there are technical issues preventing the posting of the user's contribution, the system displays an error message and suggests trying again later.			

Use Case #8: Create and Share Tutorials

Identifier	UC008

Name	Create and Share Tutorials
Purpose	Allow users and mechanics to create and share car-related tutorials.
Priority	Low
Actors	User, Mechanic
Pre-conditions	1. Account should exist. 2. Account must be logged in.
Post-conditions	The user's/mechanic's car-related tutorial is published and accessible to others.
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	User/Mechanic accesses the tutorial creation section.	1	The system presents a tutorial editor with options to add text, images, and videos.
2	The User/Mechanic creates a tutorial by adding content and providing a title and description.	2	The system saves the tutorial.
3	User/Mechanic chooses to publish the tutorial.	3	The system makes the tutorial accessible to other users in the tutorial section.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	User/Mechanic decides not to publish the tutorial	2	The system discards the tutorial, and it is not saved or published.
Exceptions: If there are technical issues preventing the creation or publication of the tutorial, the system displays an error message and suggests trying again later.			

Use Case #9: Administer Orders

Identifier	UC009
Name	Administer Orders
Purpose	Allow sellers to manage orders for spare parts.
Priority	High
Actors	Seller, Mechanic
Pre-conditions	1. Account should exist. 2. Account must be logged in.
Post-conditions	None
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Seller/Mechanic accesses the "Manage Orders" section.	1	The system displays the order management dashboard.
2	Seller/Mechanic selects a specific order to manage.	2	The system provides details of the selected order, including order status, customer information, and order items.
3	Seller/Mechanic updates the order status, adds comments, or takes relevant actions.	3	The system records the changes and updates the order information.
4	Seller/Mechanic saves the changes and confirms the update.		The system displays a confirmation message and updates the order status

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Seller/Mechanic decides not to update the order.	2	The system retains the order status and details.
Exceptions: If there are technical issues preventing the update of an order, the system displays an error message and suggests trying again later.			

Use Case #10: List Spare Parts

Identifier	UC010
Name	List Spare Parts
Purpose	Allow sellers to enter information about the spare parts they are selling.
Priority	High
Actors	Seller
Pre-conditions	1- Internet connection should be good. 2- The seller’s account should exist. 3- The seller’s account should be verified. 4- The seller has already signed-up.
Post-conditions	The seller's spare parts information is uploaded and is accessible to the users.
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Seller logs into their account.	1.1	The system authenticates the seller’s credentials.

2	Seller selects the "List spare parts" option.	2.1	The system navigates the seller to the spare parts listing page.
3	Seller enters information about the spare parts.	3.1	The system provides fields to input spare parts details.
3	Seller uploads any necessary images/documents.	3.2	The system allows the seller to upload images/documents.
4	Seller submits the spare parts listing.	4.1	The system processes and saves the listing data.
		4.2	The system confirms the successful submission.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Seller decides not to submit the spare parts listing.	1.1	The system cancels the listing process.
		1.2	The system does not save the information and it is discarded.
Exceptions: The system will display an error message and suggest trying again later if the seller attempts to list spare parts without being logged into their account, seller tries to list spare parts with incomplete or missing information or technical issues or system errors occur during the listing process.			

Use Case #11: Create Account

Identifier	UC011
Name	Create Account
Purpose	Allow sellers, users and mechanics to create their profile.
Priority	High
Actors	Seller, User, Mechanic
Pre-conditions	Internet connection should be good.
Post-conditions	The profile is created with the provided information.
Dependencies	None

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Seller, user or mechanic accesses the "Create account" page.	**1.1**	The system displays the account creation form.
2	Seller enters their personal information, business details,account information, product offerings and any necessary verification documents.	**2.1**	The system provides fields for the seller to input their information.

2	User enters their personal information.	**2.2**	The system provides fields for the user to input their information.
2	Mechanic enters their personal information, business details,account information and any necessary verification documents.	**2.3**	The system provides fields for the mechanic to input their information.
3	Seller, user or mechanic confirms the accuracy of the information and submits the account creation request.	**3.1**	The system validates the provided data for completeness and correctness.
		3.2	The system creates the account with the provided information.
		3.3	The system confirms the successful creation of the account.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Decides not to submit the information entered in the form.	**1.1**	The system cancels the process.
		1.2	The system does not save the information and it is discarded.

Exceptions: The system will display an error message and suggest trying again later if the actor tries to submit the form with incomplete or missing information or technical issues or system errors occur during the process. The system will also display error message if the format of the required information (for example email) is incorrect.

Use Case #12: Pay for Order

Identifier	UC012
Name	Pay for Order
Purpose	Allow payment transfer between user and mechanic
Priority	High
Actors	User, Mechanic
Pre-conditions	The buyer’s (user or mechanic) account should exist.
Post-conditions	The payment will be successfully transferred.
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Buyer (user or mechanic) logs into their account.	1.1	The system authenticates the user's or mechanic's credentials.

2	Buyer (user or mechanic) goes to the cart and checks out.	2.1	The system displays the checkout page.
3	Buyer (user or mechanic) enters the address and necessary information and selects the option to make a payment to a seller.	3.1	The system provides fields for the buyer (user or mechanic) to select the payment method.
4	The buyer (user or mechanic) selects the payment method.	4.1	The system displays the selected payment method.
5	The buyer (user or mechanic) clicks on the confirm button.	5.1	The system saves the selected payment method.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Buyer (user or mechanic) decides not to confirm the order.	1.1	The system cancels the process.
		1.2	The system does not save the information and it is discarded.
Exceptions: The system will display an error message and suggest trying again later if the buyer attempts to checkout without being logged into their account, buyer tries to confirm the order with incomplete or missing information or technical issues or system errors occur during the process.			

Use Case #13: Manage Spare Parts Catalog

Identifier	UC013
Name	Manage Spare Parts Catalog
Purpose	Allow the administrator to manage (add, update, remove) spare parts catalog.
Priority	Medium
Actors	Administrator
Pre-conditions	The spare parts catalog database must be available and up to date.
Post-conditions	Any changes made to the spare parts catalog, such as adding, updating, or deleting spare parts, are reflected in the system.
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Administrator logs into their administrator account.	1.1	The system authenticates the administrator's credentials.
2	Administrator selects the option to manage the spare parts catalog.	2.1	The system provides access to the spare parts catalog management interface.

3	Administrator updates (add, delete, modify) the catalog and submits the changes.	3.1	The system confirms the successful modification of the spare part catalog.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Administrator decides not to submit the changes entered in the form.	1.1	The system cancels the process.
		1.2	The system does not save the information and it is discarded.
Exceptions: The system will display an error message and suggest trying again later if the administrator tries to submit the changes with incomplete or missing information or technical issues or system errors occur during the process.			

Use Case #14: Administer Accounts

Identifier	UC014
Name	Administer Accounts
Purpose	Allow administrator to manage users' (buyers and sellers) profile.
Priority	Medium
Actors	Administrator

Pre-conditions	The accounts database must be available and up to date.
Post-conditions	Any changes made to the accounts, such as adding, updating, or deleting accounts, are reflected in the system.
Dependencies	UC015

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**
1	Administrator logs into their administrator account.	1.1	The system authenticates the administrator's credentials.
2	Administrator selects the option to manage the accounts.	2.1	The system provides access to the accounts management interface.
3	Administrator updates (add, delete, modify) the accounts and submits the changes.	3.1	The system confirms the successful modification of the accounts catalog.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Administrator decides not to submit the changes entered in the form.	1.1	The system cancels the process.

		1.2	The system does not save the information and it is discarded.
Exceptions: The system will display an error message and suggest trying again later if the administrator tries to submit the changes with incomplete or missing information or technical issues or system errors occur during the process.			

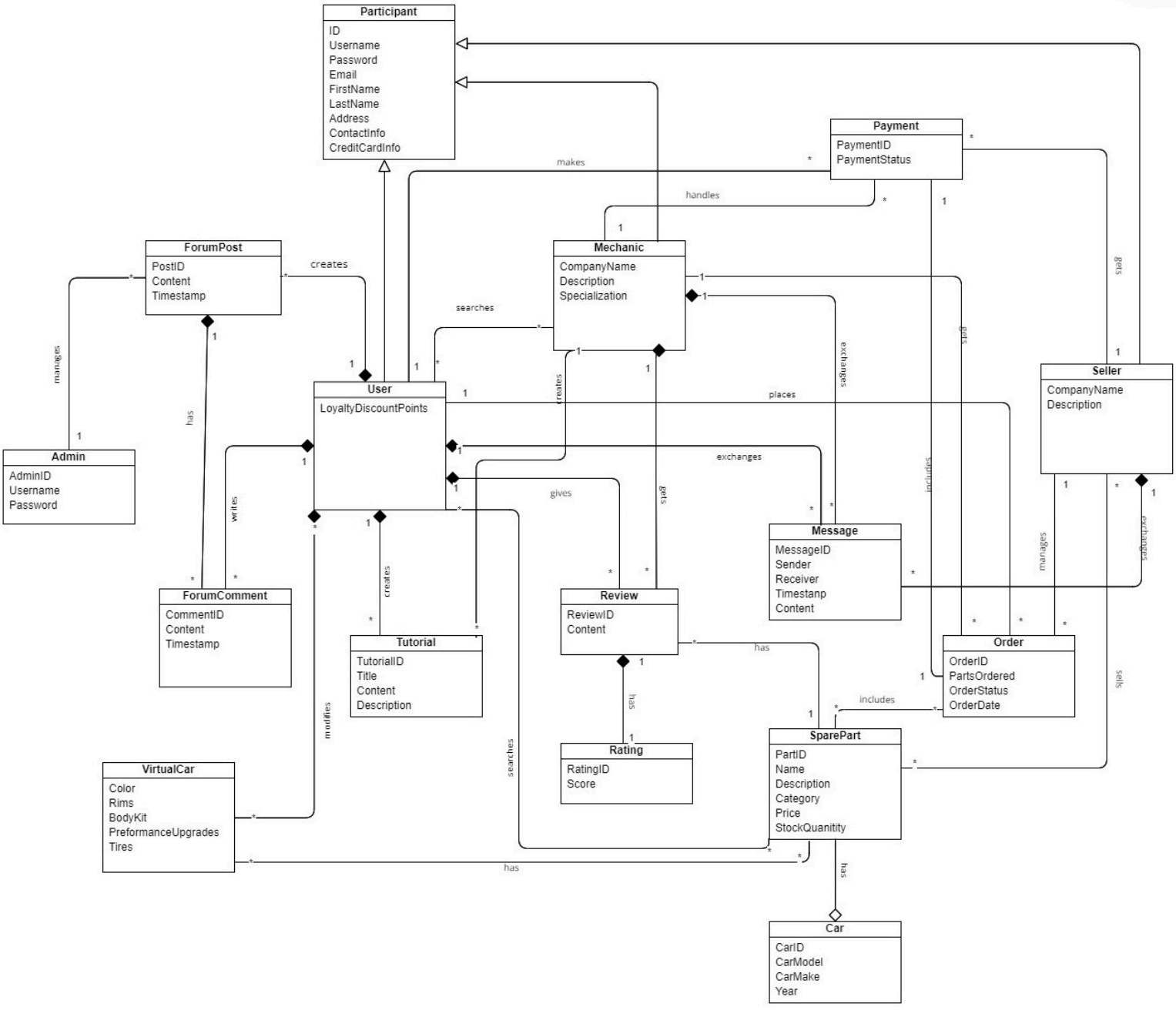
Use Case #15: Login to Account

Identifier	UC015
Name	Login to Account
Purpose	Allow Seller, User, Mechanic to login to their account
Priority	High
Actors	Seller, User, Mechanic, Administrator
Pre-conditions	Internet connection should be good.
Post-conditions	The profile is logged in.
Dependencies	UC011

Typical Course of Action (Primary Flow)			
S#	**Actor Action**	**S#**	**System Response**

1	Access the "Login" page.	**1.1**	The system displays the account creation form.
2	Enter username and password.	**2.1**	The system verifies the credentials and checks for account existence.
3	Click the "Login" button	**3.1**	The system logs into the account.
		3.2	System displays a successful login message.

Alternative Course of Action			
S#	**Actor Action**	**S#**	**System Response**
1	Decides not to log in.	**1.1**	The system remains on the login page.
Exceptions: If the provided username and password do not match, the system displays an error message. If the account does not exist, the system informs that the account is not found.			



Analysis Class Diagram