



# CARSHOP



The Perfect Car - In Your Cart

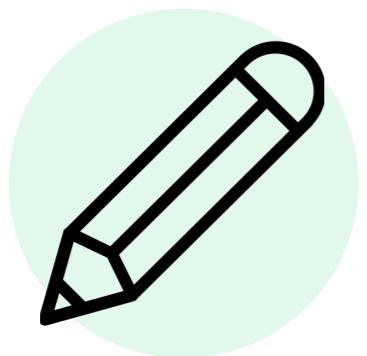


“CarShop is dé webshop voor auto's,  
met een ruim aanbod en scherpe prijzen.”

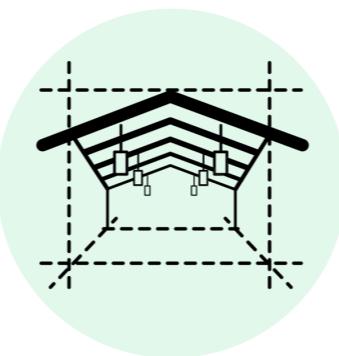
Team  
Alpha Q



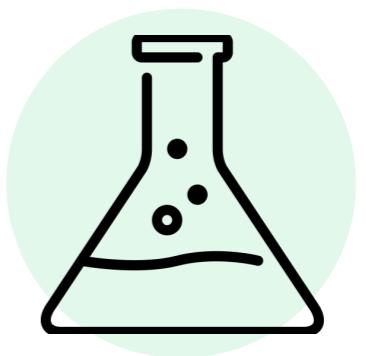
# Ontwikkelproces



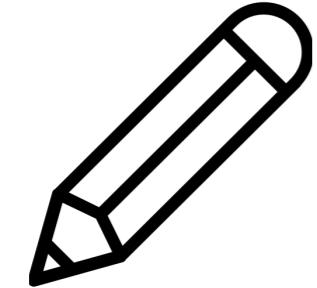
1. Sketching



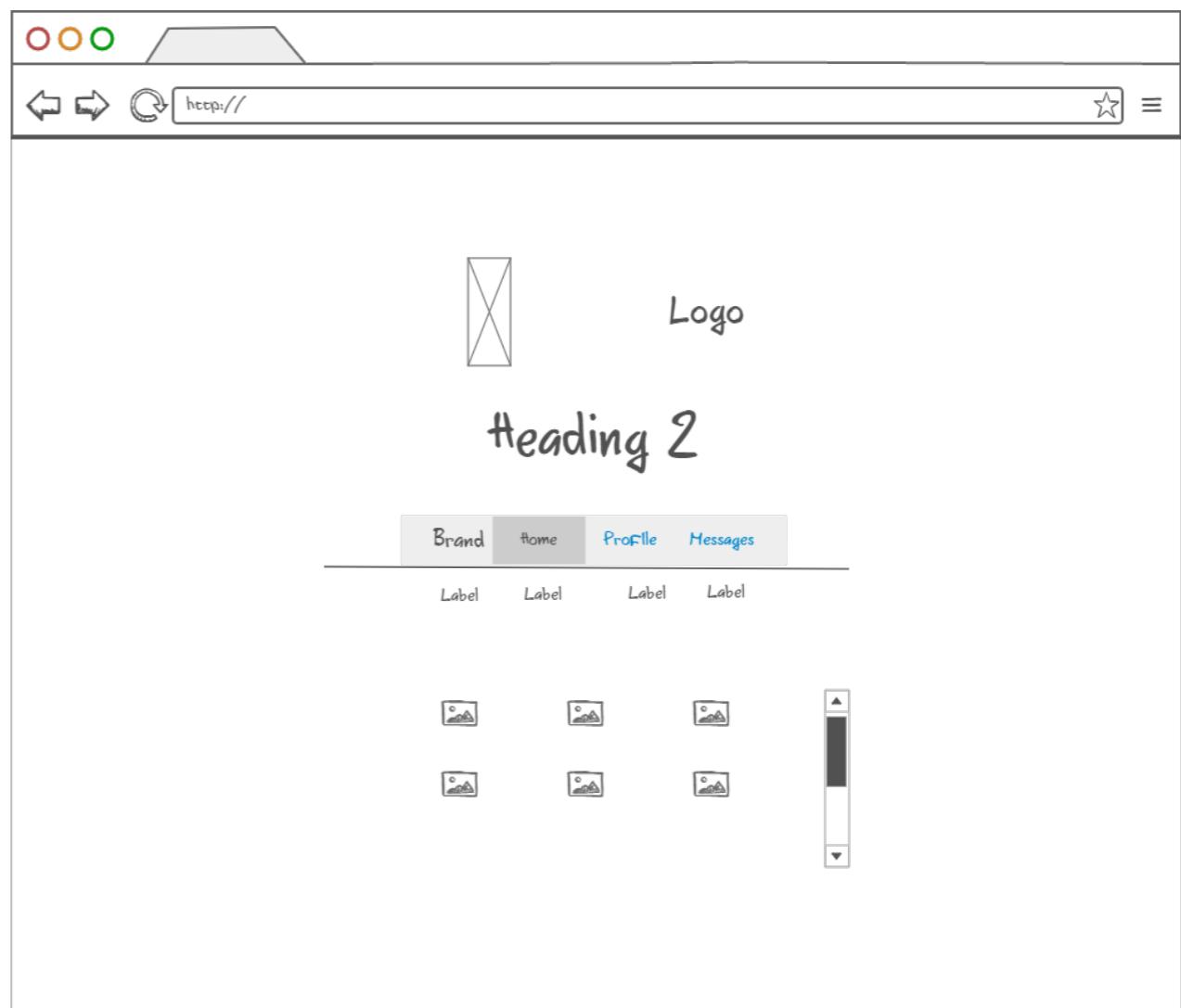
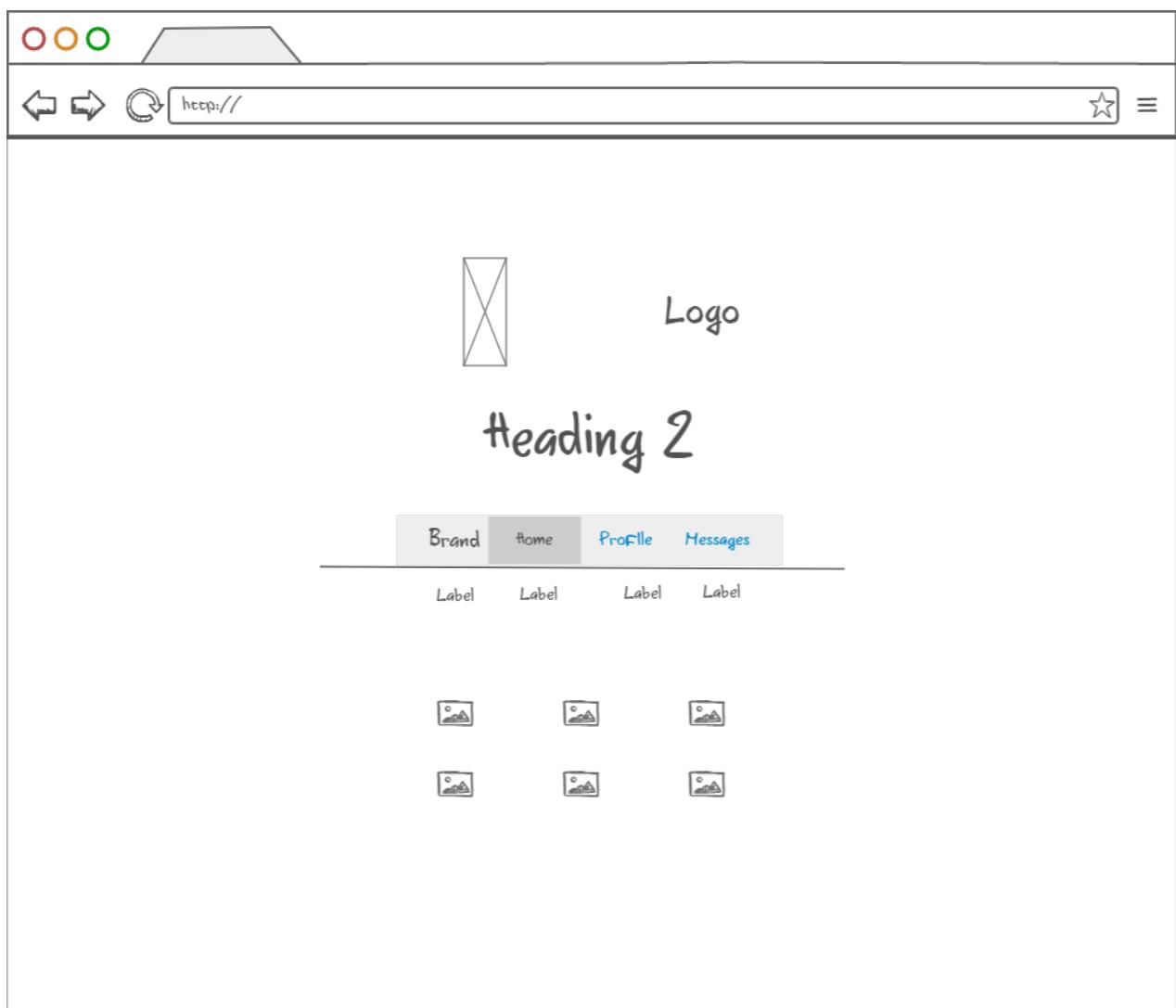
2. Architecting

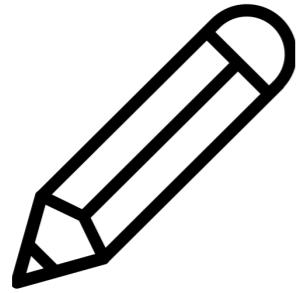


3. Testing



# Sketching: Home & Cars

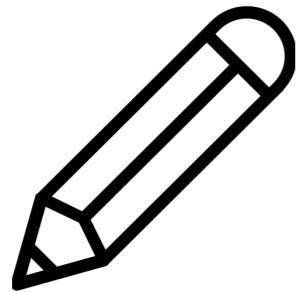




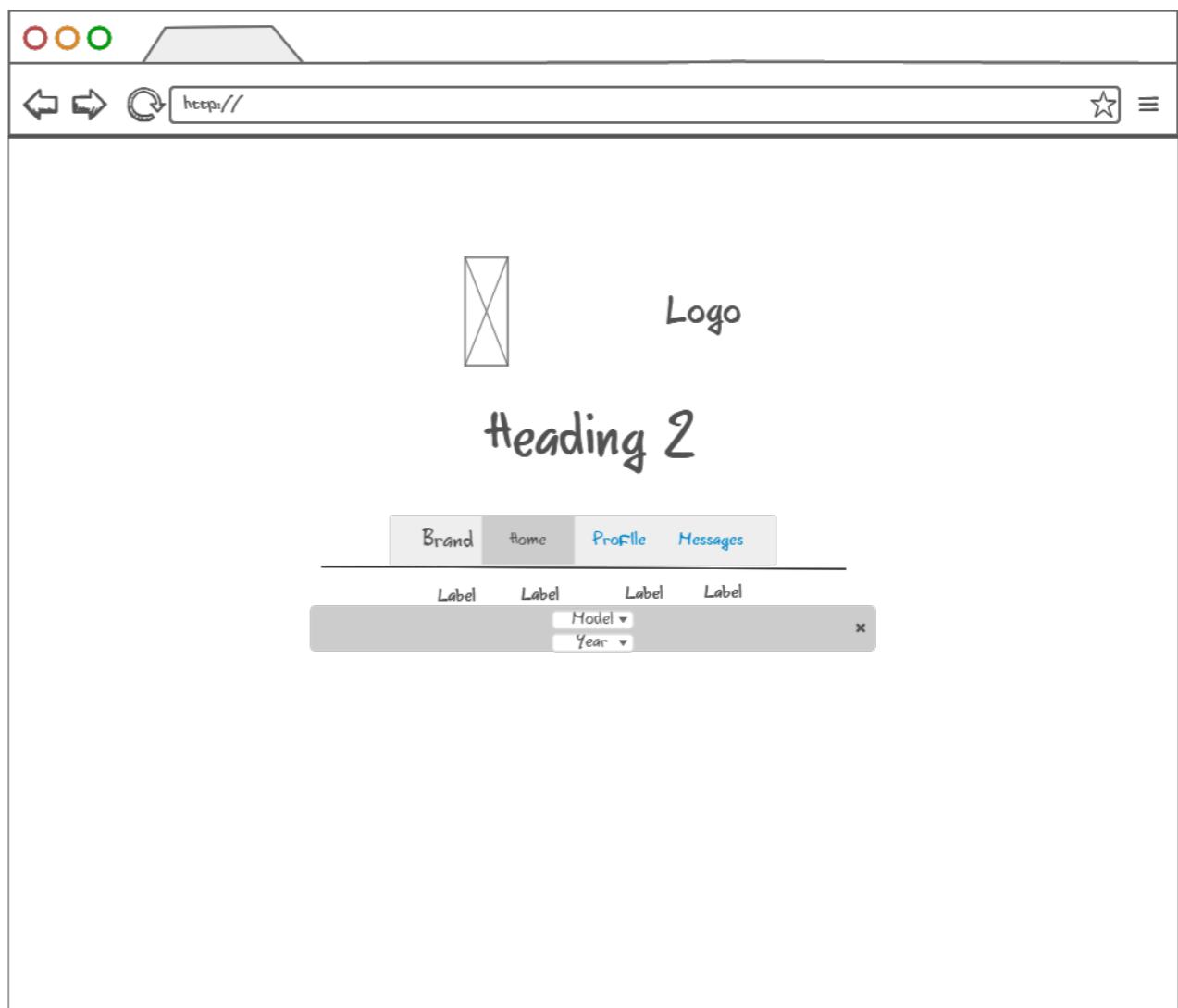
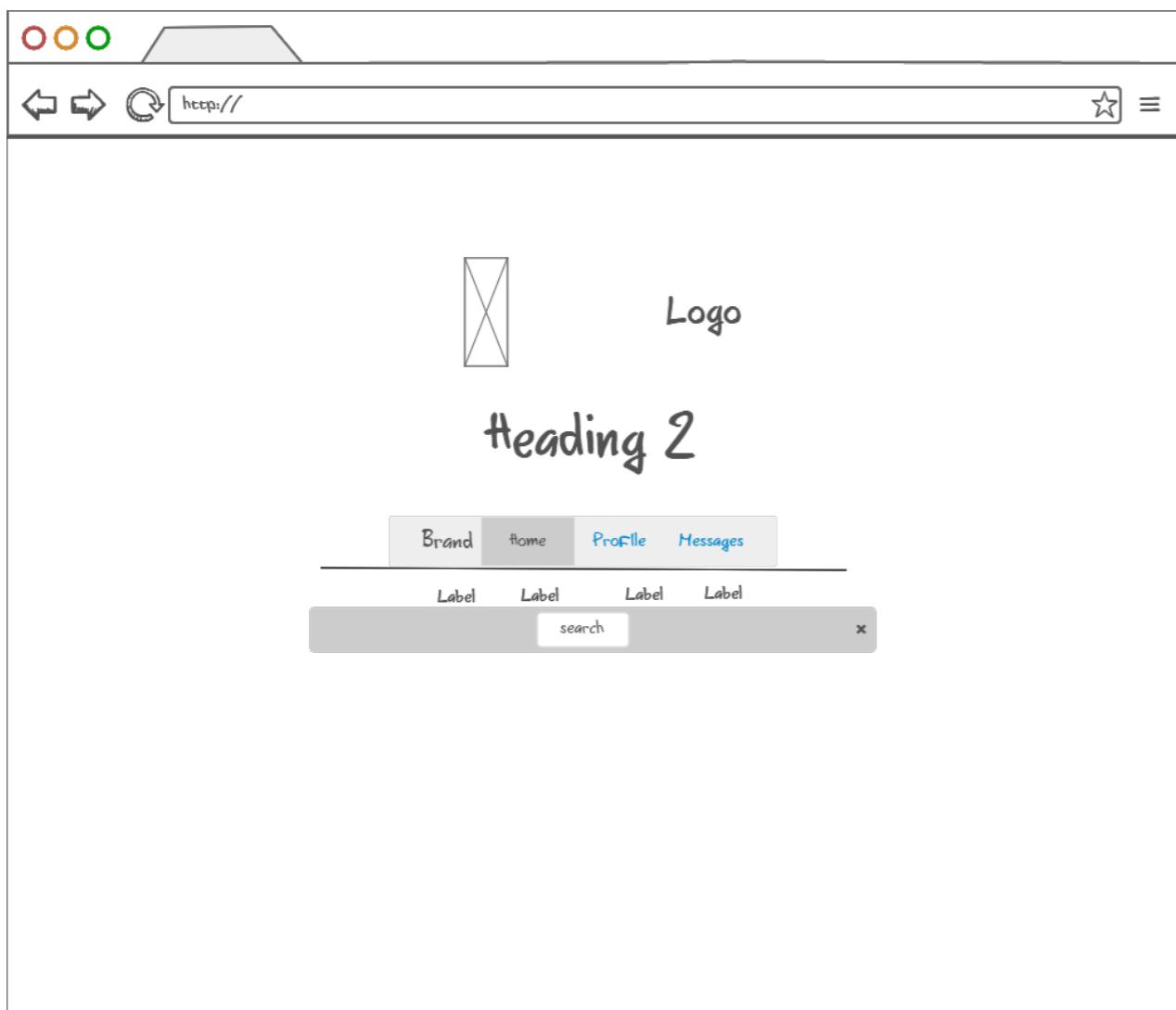
# Sketching: Login & Registration

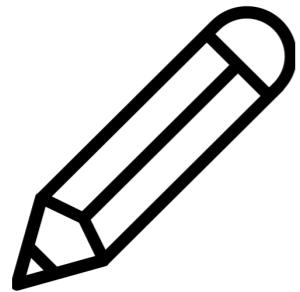
This sketch illustrates a user interface for a web application. At the top, there's a header bar with three colored circles (red, orange, green) and a search bar with a URL field containing "http://". Below the header is a logo placeholder and a "Heading 2" title. A navigation bar at the bottom includes tabs for "Brand", "Home", "Profile" (which is highlighted in blue), and "Messages". Underneath the navigation bar, there are four "Label" components. Below them is an "Inline form" containing fields for "Email" and "Password", along with a "Sign in" button and a checked checkbox.

This sketch illustrates a user interface for a web application. It features a header bar with three colored circles (red, orange, green) and a search bar with a URL field containing "http://". Below the header is a logo placeholder and a "Heading 2" title. A navigation bar at the bottom includes tabs for "Brand", "Home", "Profile" (highlighted in blue), and "Messages". Underneath the navigation bar, there are four "Label" components. A "Register form" box contains fields for "First name", "Last name", "Email", and "Password", each with a corresponding label and input field. A "Submit" button is located at the bottom right of the form.



# Sketching: Search & Filter





# Sketching: Cartlist & Contact

Logo

Heading 2

Brand Home Profile Messages

Label Label Label Label

Select Remove Set order x

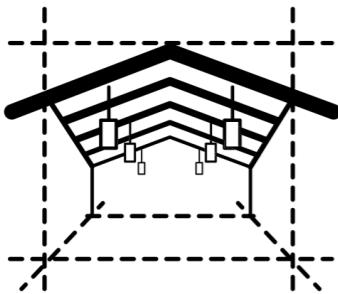
Logo

Heading 2

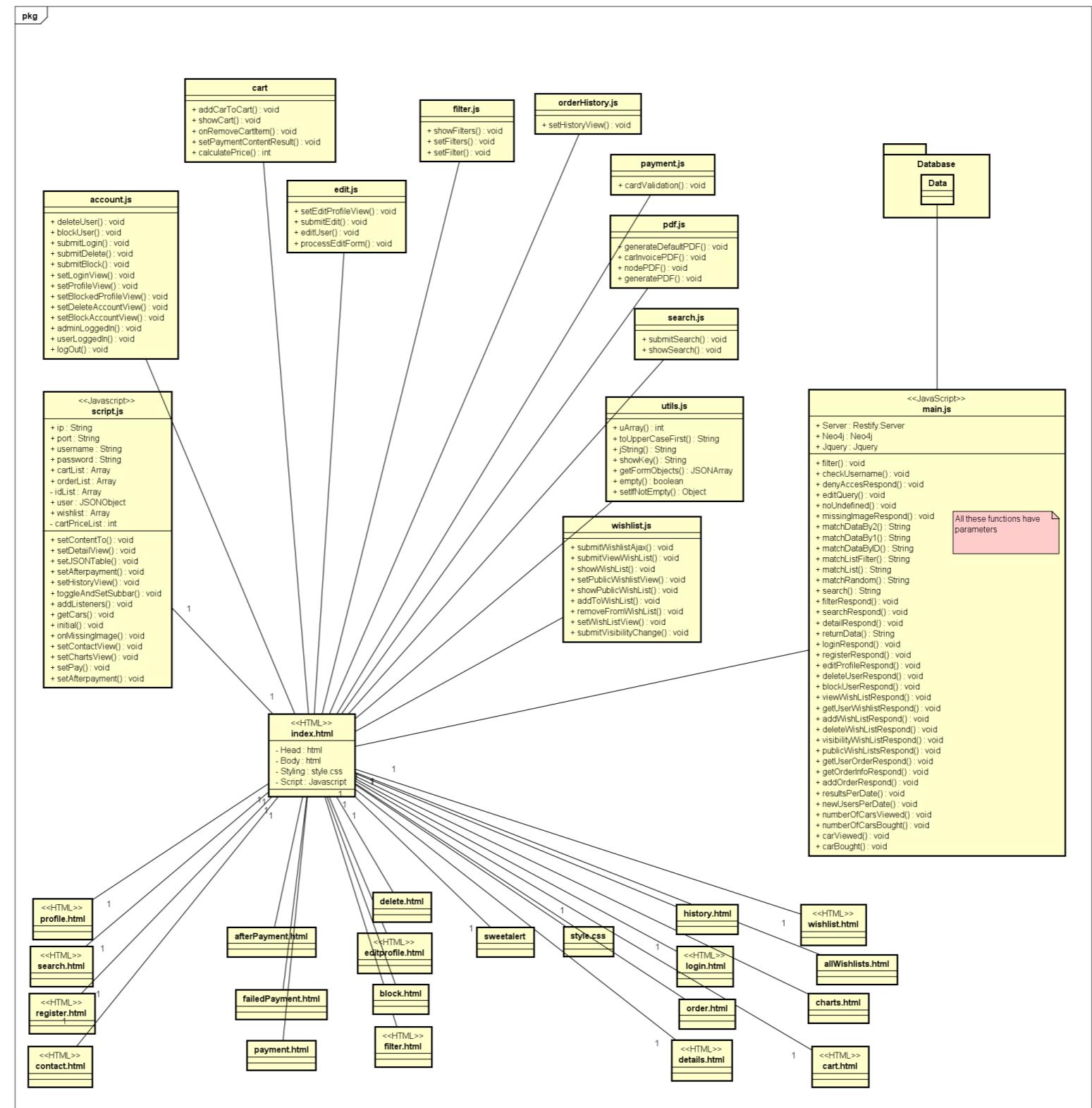
Brand Home Profile Messages

Label Label Label Label

We're four Rotterdam University students who have embarked on the exciting mission to conquer the vast interwebz with our data-driven web-application, AKA CarShop.

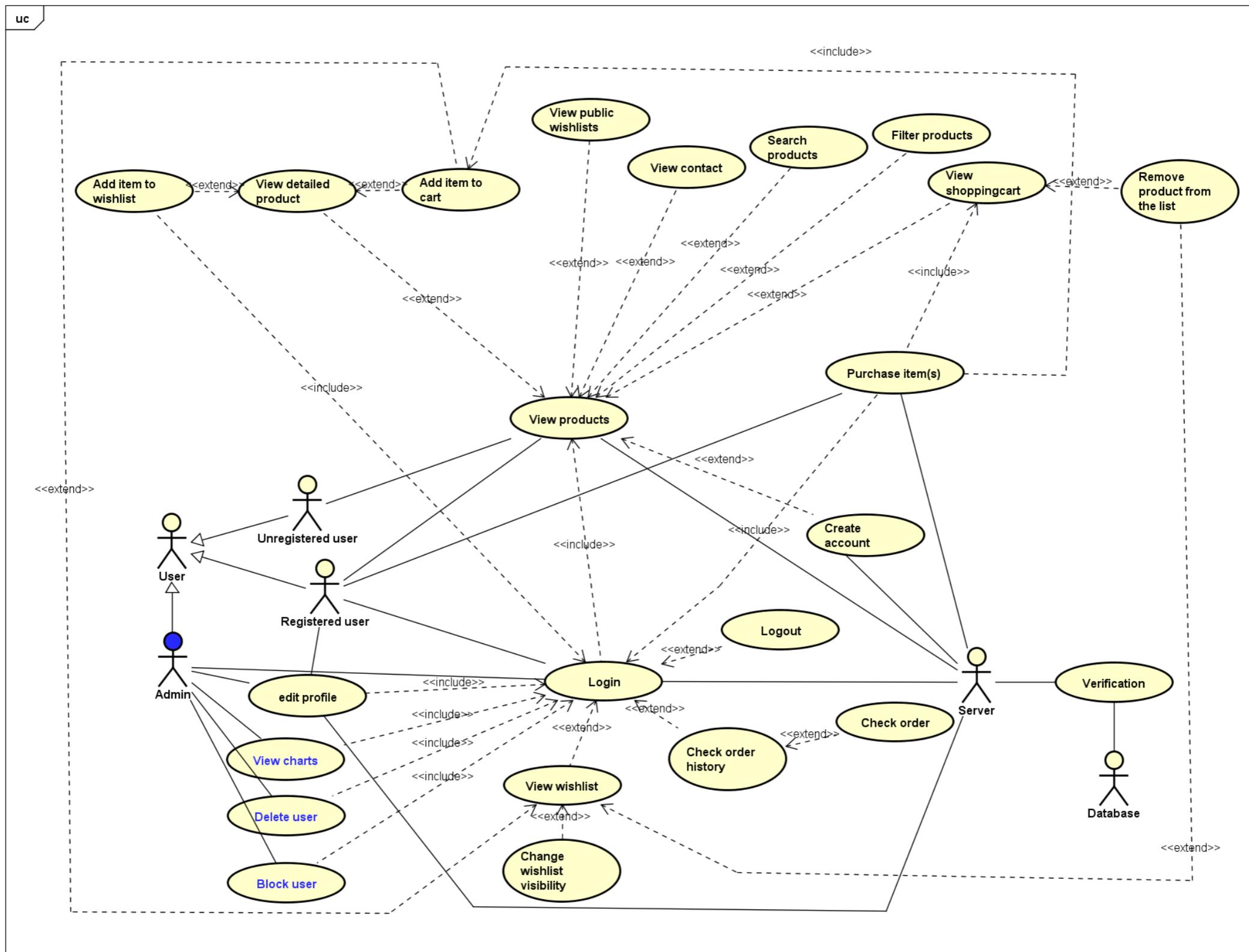
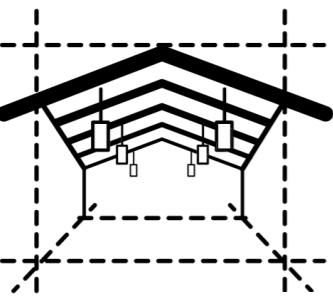


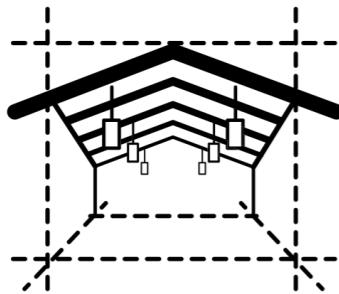
# Architecture: Class Diagram



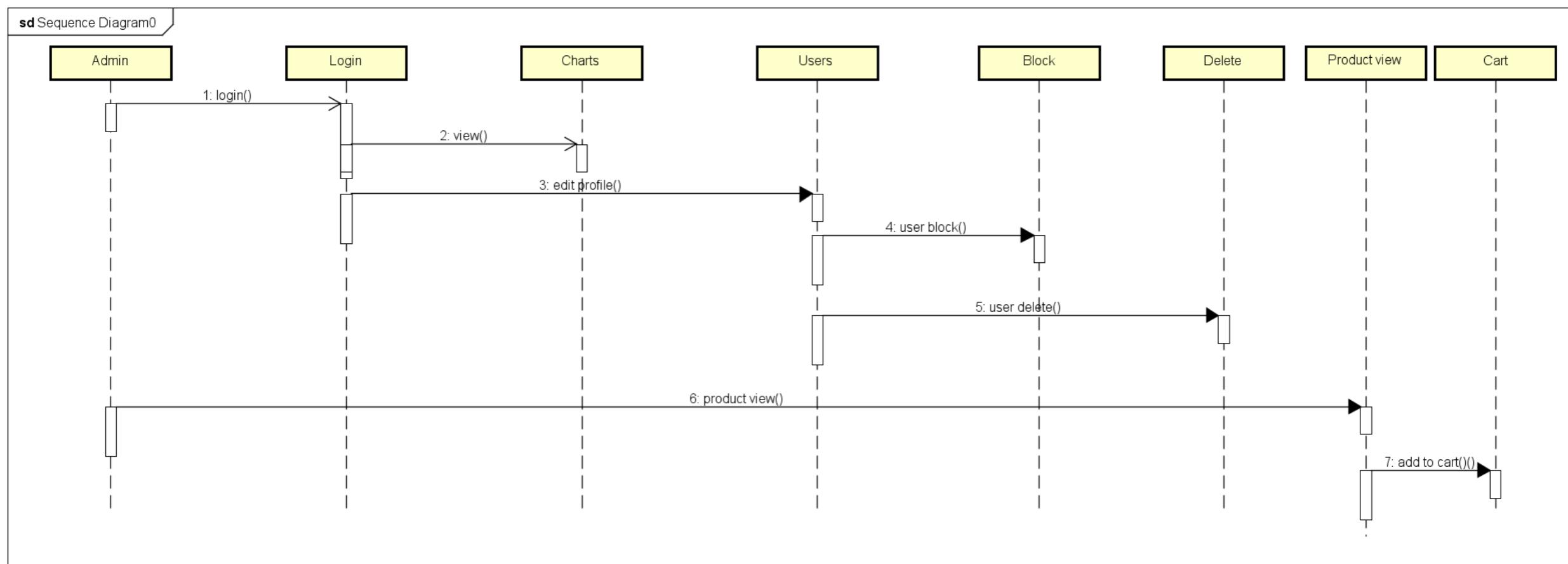


# Architecture: Use Case Diagram



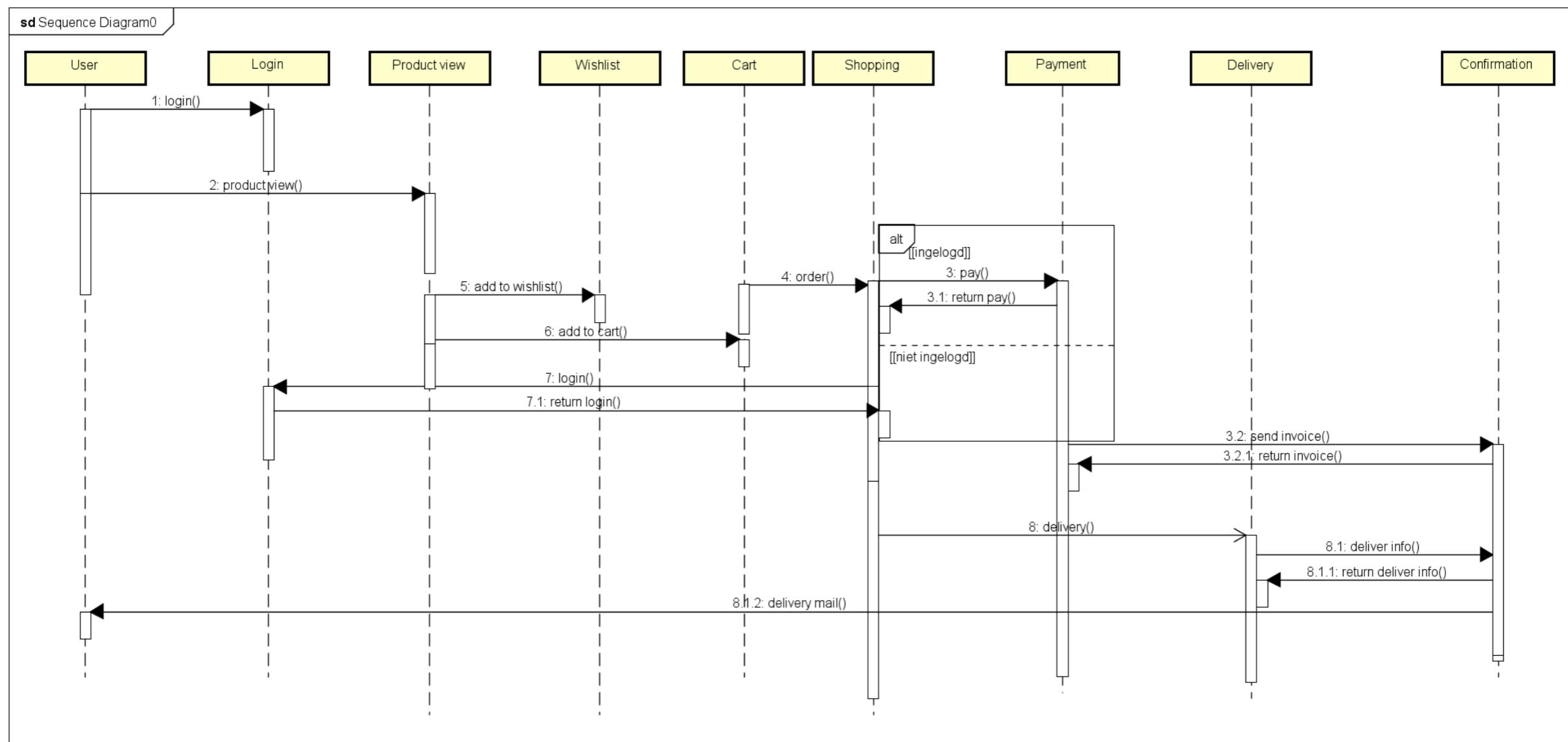


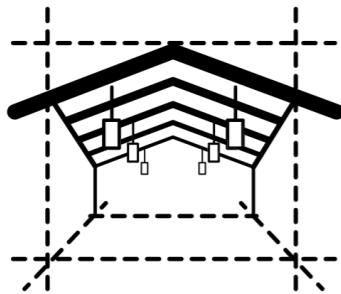
# Architecture: Sequence Diagram



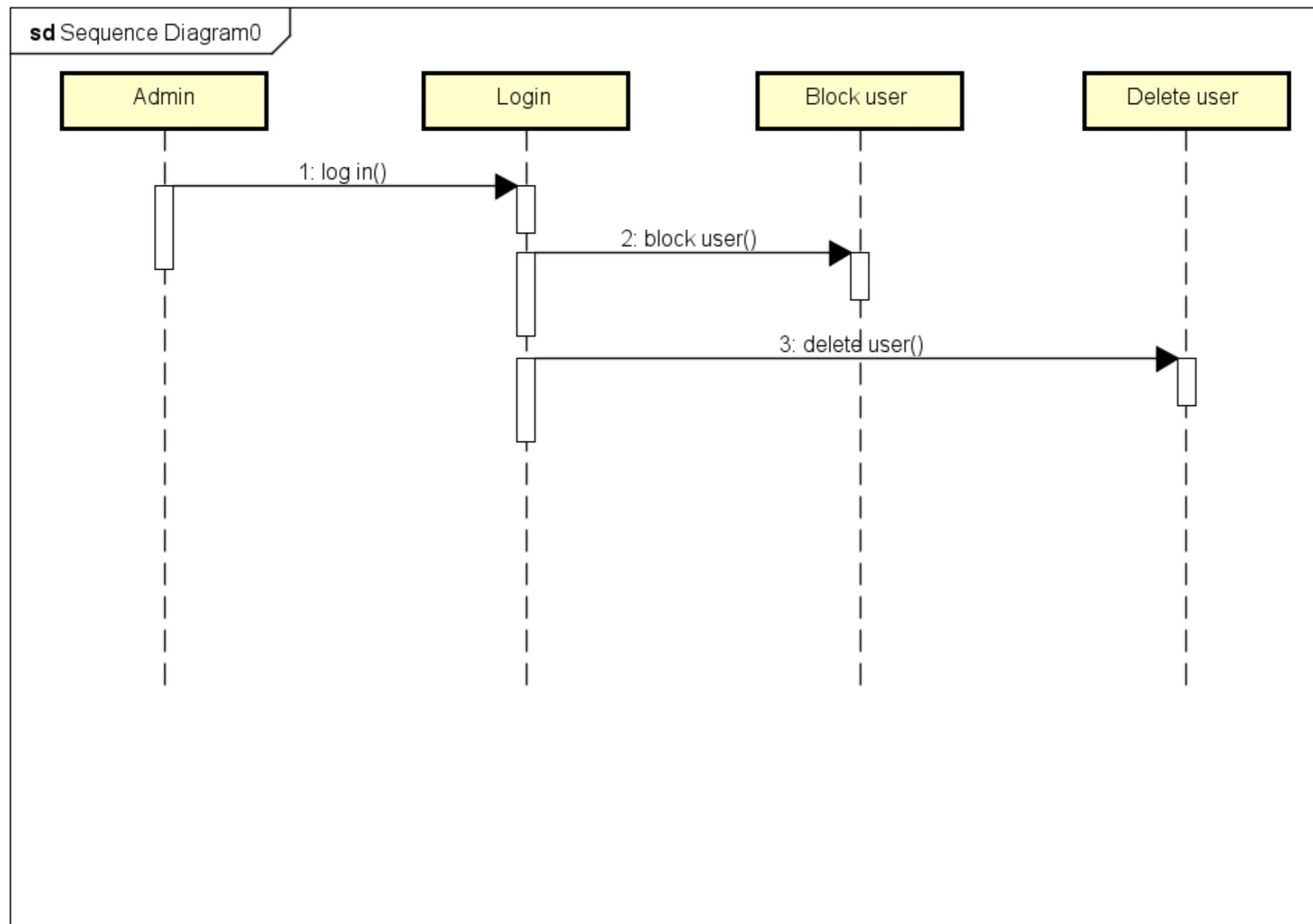


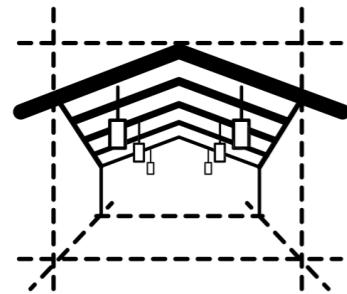
# Architecture: Sequence Diagram



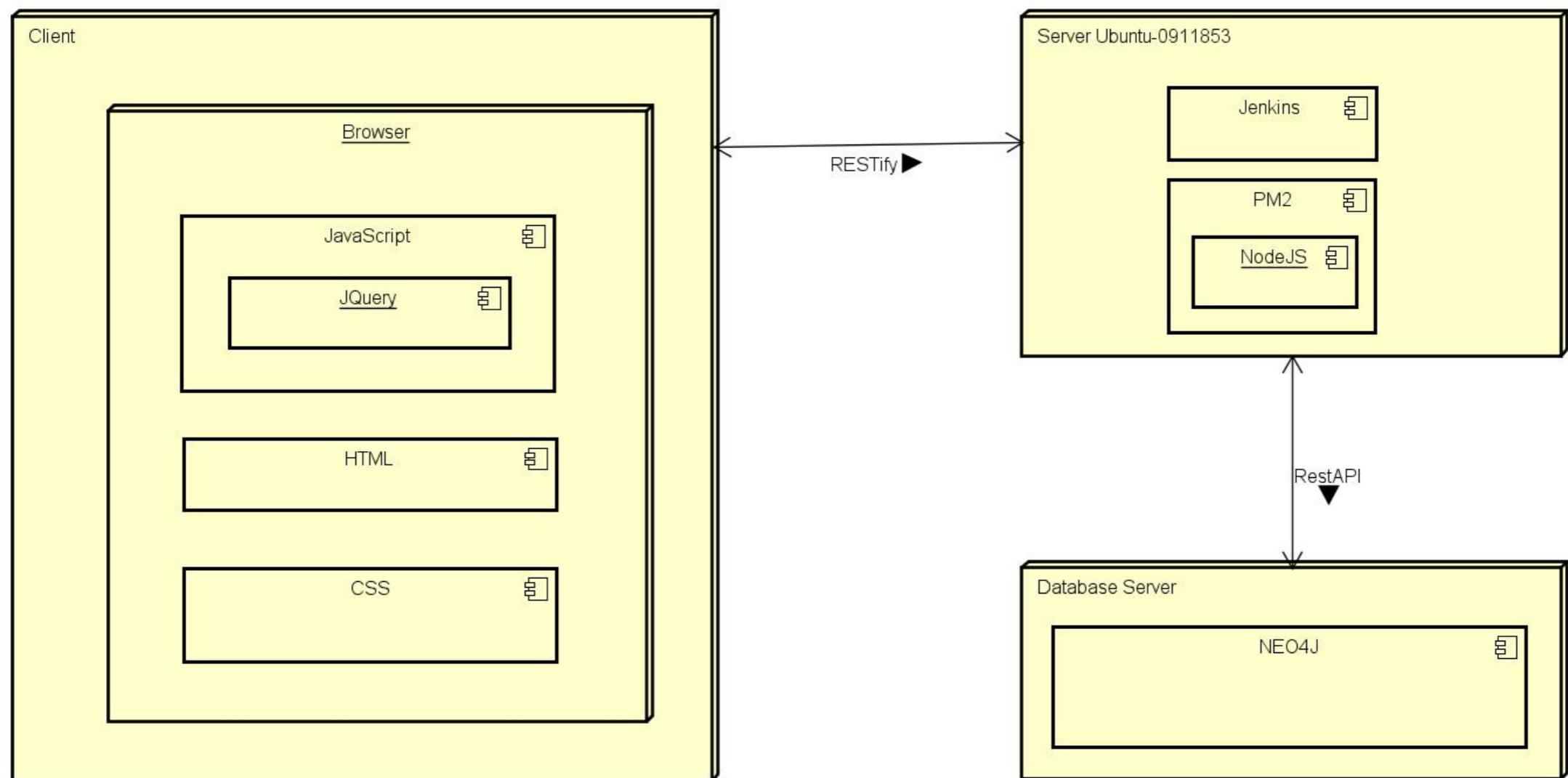


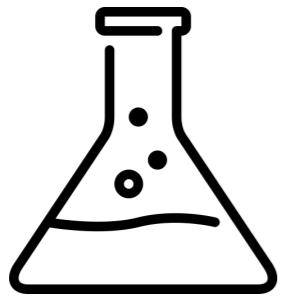
# Architecture: Sequence Diagram





# Architecture: Deployment Diagram





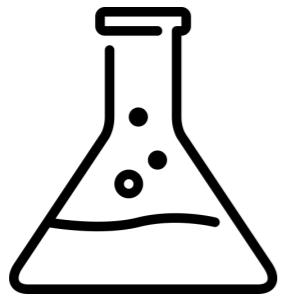
# Testing: Pipeline

All [+](#)

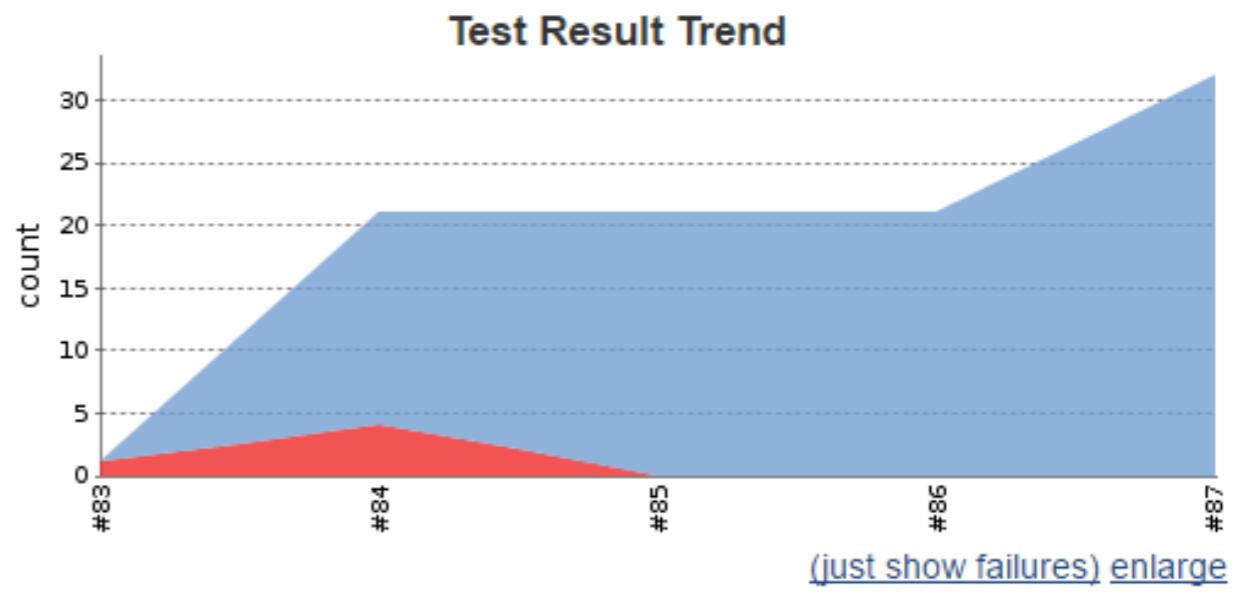
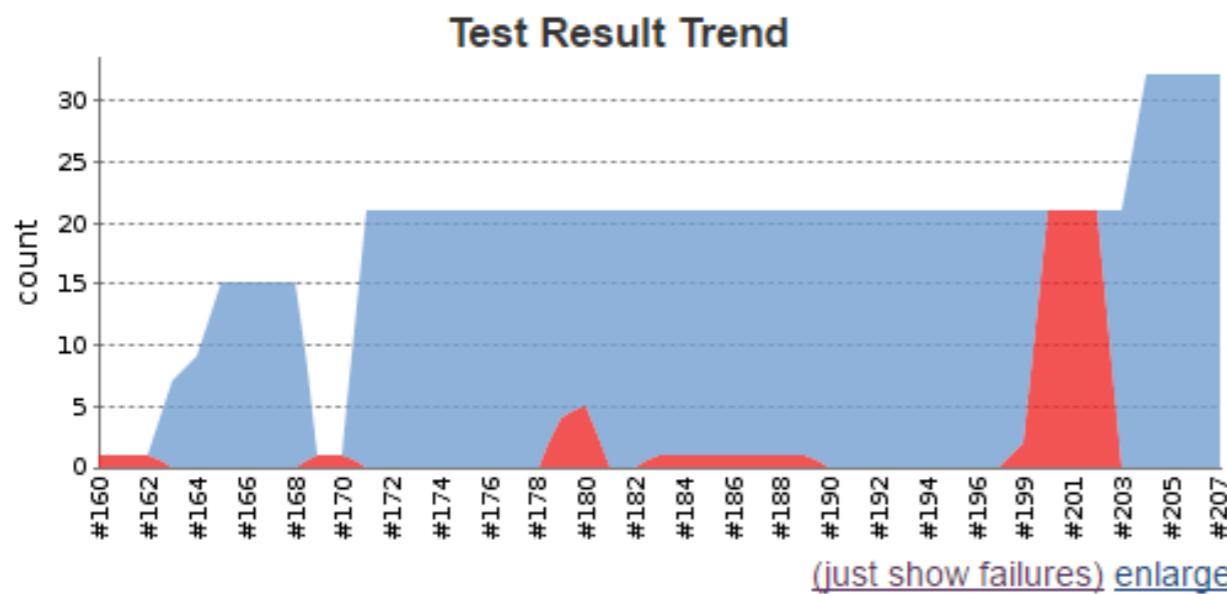
S	W	Name ↓	Last Success	Last Failure	Last Duration	
		<a href="#">CarShop</a>	2 hr 57 min - #87	1 day 7 hr - #84	1 min 5 sec	
		<a href="#">CarShop Testing</a>	1 hr 35 min - #207	1 day 3 hr - #202	35 sec	

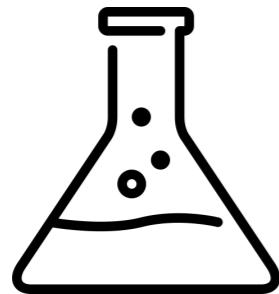
Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)



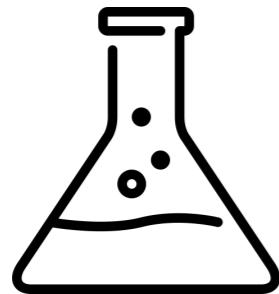
# Testing: Test Reports





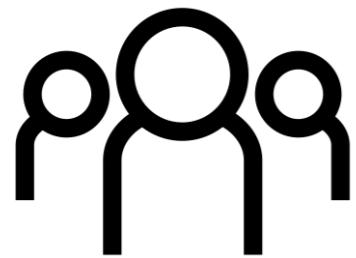
# Testing: Test Reports

Class	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
<a href="#">GET Requests Detail</a>	0.66 sec	0		0		2		2	
<a href="#">GET Requests Filter</a>	0.66 sec	0		0		3		3	
<a href="#">GET Requests Index</a>	0.66 sec	0		0		2		2	
<a href="#">GET Requests New Users Per Date</a>	0.66 sec	0		0		2		2	
<a href="#">GET Requests Number Of Cars Bought</a>	0.66 sec	0		0		2		2	
<a href="#">GET Requests Number Of Cars Viewed</a>	0.66 sec	0		0		2		2	
<a href="#">GET Requests Search</a>	0.66 sec	0		0		2		2	
<a href="#">POST Requests Add Order</a>	0.66 sec	0		0		2		2	
<a href="#">POST Requests Block</a>	0.66 sec	0		0		1		1	
<a href="#">POST Requests Delete</a>	0.66 sec	0		0		1		1	
<a href="#">POST Requests Edit User Edit User (correct)</a>	0.66 sec	0		0		1		1	
<a href="#">POST Requests Edit User Edit User (incorrect)</a>	0.66 sec	0		0		1		1	
<a href="#">POST Requests Login</a>	0.66 sec	0		0		2		2	
<a href="#">POST Requests Register</a>	0.66 sec	0		0		1		1	
<a href="#">Register Registration is Valid Form Object</a>	0.66 sec	0		0		1		1	
<a href="#">Utilities Beautified String</a>	0.66 sec	0		0		1		1	
<a href="#">Utilities Empty String</a>	0.66 sec	0		0		1		1	
<a href="#">Utilities Fallback-String if Initial is Empty</a>	0.66 sec	0		0		2		2	
<a href="#">Utilities Single-Quoted Escape String</a>	0.66 sec	0		0		1		1	
<a href="#">Utilities Unique Array</a>	0.66 sec	0		0		1		1	
<a href="#">Utilities Uppercase First Letter</a>	0.66 sec	0		0		1		1	



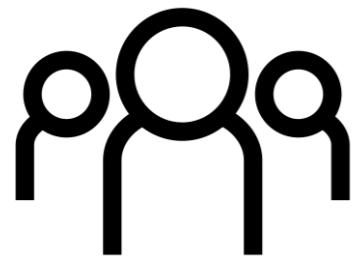
# Testing: Test Reports

Class	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
<a href="#">GET Requests Detail</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">GET Requests Filter</a>	4 sec	0		0		3 +3		3 +3	
<a href="#">GET Requests Index</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">GET Requests New Users Per Date</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">GET Requests Number Of Cars Bought</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">GET Requests Number Of Cars Viewed</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">GET Requests Search</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">POST Requests Add Order</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">POST Requests Block</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">POST Requests Delete</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">POST Requests Edit User Edit User (correct)</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">POST Requests Edit User Edit User (incorrect)</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">POST Requests Login</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">POST Requests Register</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">Register Registration is Valid Form Object</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">Utilities Beautified String</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">Utilities Empty String</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">Utilities Fallback-String if Initial is Empty</a>	4 sec	0		0		2 +2		2 +2	
<a href="#">Utilities Single-Quoted Escape String</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">Utilities Unique Array</a>	4 sec	0		0		1 +1		1 +1	
<a href="#">Utilities Uppercase First Letter</a>	4 sec	0		0		1 +1		1 +1	



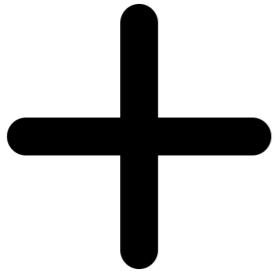
# SCRUM: Product Backlog

ID	Feature	Category	Priority	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Definition of done	User Story	Taken + points
1	Onderscheid tussen admin en gebruiker	Code	M			X				Een admin of gebruiker kunnen succesvol inloggen op de site.	Als gebruiker of als admin wil ik kunnen inloggen op de site om volledig gebruik te kunnen maken van alle functies.	Onderscheid maken tussen admin en gebruiker op dezelfde inlogpagina (29 punten)
2	Producten weergeven in een lijst	Code	M		X					Alle producten zijn op een pagina te zien op de site.	Als gebruiker wil ik producten kunnen zien, opdat ik kan zien wat ik wil gaan kopen	Producten laten zien op de homepage en de 'carspage'(13 punten)
3	Filter voor de producten	Code	M			X				Gebruikers kunnen de producten op een pagina filteren.	Als gebruiker wil ik producten kunnen filteren, om tot een juiste keuze te komen	Filterfunctie toevoegen met meerdere functies(21 punten)
4	Registratie voor nieuwe gebruikers	Code	M			X				Bezoekers van de site kunnen zich registreren en met die gegevens inloggen. De gebruikersnaam moet uniek zijn.	Als gebruiker wil ik me registeren opdat ik kan inloggen	Registratiformulier maken en zorgen dat er geen dubbele gebruikersnamen zijn(13 punten)
5	Kopen van producten	Code	M				X			Klanten kunnen producten kopen en afrekenen	Als gebruiker wil ik producten kunnen kopen en afrekenen, zodat ik een auto kan bemachtigen	Creditcardbetaalfunctie maken (21 punten)
6	Producten toevoegen in winkelwagen	Code	M			X				Gebruikers kunnen producten in een winkelwagen leggen en verder shoppen zonder dat de producten uit de winkelwagen verdwijnen.	Als gebruiker wil ik een auto in mijn winkelwagen kunnen toevoegen, opdat ik verder kan winkelen zonder deze uit het oog te verliezen	Winkelwagen maken, add to cart functie, zorgen dat deze tijdelijk wordt opgeslagen(8 punten)
7	Het weergeven van min 3 actuele statistieken, voor de admin.	Code	M				X			De admin krijgt een eigen pagina waarin minimaal 3 actuele statistieken te zien zijn	Als admin wil ik minimaal 3 statistieken zien over de website en de verkoop opdat ik kan inschatten wat ik beter kan doen aan de website of de verkoopstrategieën.	Bekende autos, gekochte autos, combinatie van beiden, aantal nieuwe users per dag (21 punten)
8	Wijzigingen aanbrengen bij een gebruiker vanuit een adminaccount	Code	M				X			De admin moet als enige wijzigingen kunnen aanbrengen aan de account van de gebruiker, danwel het account blokkeren en/of verwijderen	Als admin wil ik gebruikers kunnen blokkeren of verwijderen opdat ik illegale activiteiten tegen kan gaan	Admin login, statsknop, extra block, wijzig en delete buttons in account(13 punten)
9	Geplaatste orders bekijken(order history)	Code	M				X			De klant krijgt een lijst te zien van gekochte producten	Als klant wil ik een bestelling terugzien, opdat ik de factuur kan downloaden	Checkout view met totaalprijs maken(5 punten)
11	Producten opslaan in een lijst(wishlist)	Code	M			X				De klant kan alle producten opslaan in een lijst om later makkelijk de producten te kunnen terugvinden	Als gebruiker wil ik producten opslaan in een lijst die alleen voor mij of voor iedereen toegankelijk is en ook de wishlists van andere gebruikers kan zien	Producten opslaan in een wishlist(5 punten)
12	Keuze om wishlist openbaar of privé te maken	Code	M			X				De wishlist van de klant is normaal gesproken privé, de klant kan deze echter ook openbaar stellen	Als gebruiker wil ik mijn wishlist op private of public zetten wanneer ik wil.	Radio box maken met public en private wishlist in account gebruiker (11 punten)
13	Data in de database toevoegen en toegankelijk maken	Code	M	X	X					Ontwikkelaars kunnen data in de database toevoegen	Als ontwikkelaar wil ik data in de database kunnen toevoegen opdat deze kan worden getoond op de website	Auto's toevoegen aan de database en mogelijkheid maken om gebruikers en overige data toe te voegen (21 punten)
14	Pipelines instellen	Code	M				X			Developing, testing en productionpipeline in Jenkins maken	Als ontwikkelaar wil ik gebruik maken van continuous integration door middel van een developing, testing en productionpipeline in Jenkins.	Developing, testing en productionpipeline maken in Jenkins(8 punten)



# SCRUM: Product Backlog

ID	Feature	Category	Priority	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Definition of done	User Story	Taken + points
15	Server instellen	Code	M	X						De server kan verbinding maken met zowel de database als het opdienen van een webpagina en het beheren van verkeer hiertussen.	Als ontwikkelaar wil ik gebruik maken van een server, opdat ik verbinding kan maken tussen de database en de website	Verbinden met de server ,NodeJs installeren, Apache installeren ,NodeJs connecten met Neo4j database, Remote acces tussen NodeJs en Neo4j (7 punten)
16	Maak UML diagrammen	Astah	M	X	X	X				Als de UML diagrammen duidelijk zijn en overeenkomen met de producten.	Als ontwikkelaar wil ik use case diagram, class diagram, sequence diagram, deployment diagram en ERD maken opdat ik een beter overzicht op het systeem heb en nieuwe teamleden zich eenvoudiger kunnen inlezen in het systeem.	Use case, sequence, deployment, ERD diagrammen maken aan de hand van de implementaties die gemaakt zijn (34 punten)
17	Maak een database	Cypher/Neo4J	M	X	X					Er is een database die klaar is om autos, orders, gebruikers en overige data te opslaan en ophaalbaar te maken	Als ontwikkelaar wil ik beschikken over een database waar al mijn producten, gebruikers en order instaan, opdat ik deze kan opslaan en kan opvragen.	Neo4J Database maken (5 punten)
18	Usability test	User	C					X		Er is meer dan 1 test gedaan en de feedback is verwerkt in het product.	Als ontwikkelaar wil ik een usability test uitvoeren onder gebruikers,	Usability test onder meerdere gebruikers uitvoeren(8 punten)
19	Samenwerkingscontract	Tutor	M	X						Het samenwerkingscontract is geaccepteerd en ondertekend door alle teamleden die aan het product meewerken.	Als teamlid wil ik een samenwerkingscontract hebben opdat onenigheden in het team snel kunnen worden opgelost aan de hand van het contract.	Samenwerkingscontract opstellen en ondertekenen (0,5 punten)
20	Updaten UML	Astah	M					X		De diagrammen die ontwikkeld zijn in sprint 1 tot en met 3 moeten volledig worden herzien en aangepast worden aan de laatste versie van het eindproduct.	Als ontwikkelaar wil ik alle UML diagrammen geupdated hebben wanneer het eindproduct af is, opdat een nieuwe developer makkelijker het product kan begrijpen.	Alle UML diagrammen updaten uit User Story 16 (18 punten)
21	Bugfixes	Code	S					X		Er zijn geen bugs meer te vinden in het systeem.	Als ontwikkelaar wil ik alle bugs uit het systeem halen, opdat de klant een optimale foutloze gebruikerservaring heeft.	Op zoek gaan naar bugs door alles uit te testen(5 punten)



## Extra Features

Data Crawler

Encryption

Single Page

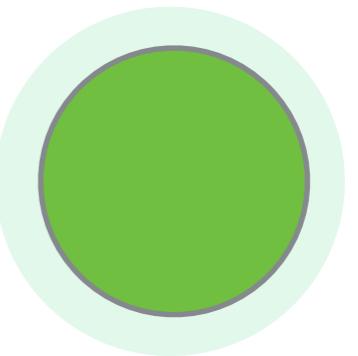
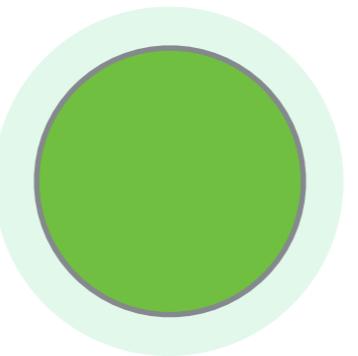
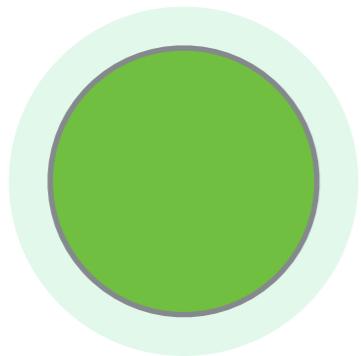
Containerized Deployment

Multiplatform

Concurrency in the Backend



# Demo





# Thank You



Team  
Alpha Q