

ASSIGNMENT COVERSHEET

UTS: ENGINEERING & INFORMATION TECHNOLOGY		
SUBJECT NUMBER & NAME 41026 Advanced Software Development	NAME OF STUDENT(s) (PRINT CLEARLY) Joshua Browne Dev Joisar Gurudeep Singh Dhinjan Lavanya Chandna Keegen Wallis	STUDENT ID(s) 99138950 14083392 24555981 24555979 14272266
STUDENT EMAIL Joshua.Browne@student.uts.edu.au dev.s.joisar@student.uts.edu.au Keegen.j.Wallis@student.uts.edu.au gurudeep.s.dhinjan@student.uts.edu.au lavanya.chandna@student.uts.edu.au	STUDENT CONTACT NUMBER +61 478 560 785 +61 431 937 737 +61 478 560 785 +61 468 378 415	
NAME OF TUTOR Guangsheng Zhang	TUTORIAL GROUP 8	DUE DATE 1st November 2024
ASSESSMENT ITEM NUMBER & TITLE Assessment task 3: Project Release 2		
<p><input type="checkbox"/> x I confirm that I have read, understood and followed the guidelines for assignment submission and presentation on page 2 of this cover sheet.</p> <p><input type="checkbox"/> x I confirm that I have read, understood and followed the advice in the Subject Outline about assessment requirements.</p> <p><input type="checkbox"/> x I understand that if this assignment is submitted after the due date it may incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.</p>		
<p>Declaration of originality: The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s) and has not been previously submitted for assessment. I understand that, should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with University policy and rules. In the statement below, I have indicated the extent to which I have collaborated with others, whom I have named.</p>		
<p>Statement of collaboration:</p> <div style="display: flex; justify-content: space-around; align-items: center;"> Signature of student(s) Dev Joisar    Joshua Browne </div>		



ASSIGNMENT RECEIPT

To be completed by the student if a receipt is required

SUBJECT NUMBER & NAME	NAME OF TUTOR	
SIGNATURE OF TUTOR		RECEIVED DATE

**University of Technology Sydney
(UTS)**

**41026 Advanced Software
Development**

**Assignment 3
Project Release 2**

Workshop 01, Group 08

TABLE OF CONTENTS

1. Individual Contribution.....	4
2. Requirements.....	5
2.1. Requirements Backlog.....	5
2.2. Applying INVEST Criteria to Each User Story.....	14
2.3. Splitting User Stories for INVEST.....	16
2.4. Non-Functional Requirements (NFR).....	18
2.5. Data Requirements.....	22
3. Project Schedule.....	25
4. Cost.....	27
5. Constraints and Risks.....	28
6. Security in the Software.....	29
7. Solution Architecture.....	30
7.1. Data Architecture Model.....	30
7.2. Application Architecture Model.....	31
8. Solution Design.....	34
8.1. Interaction Design Model.....	34
8.2. Backend Database Model.....	39
9. Testing Design.....	40
10. Continuous Deployment Design.....	47
11. Project Timesheet.....	49
11.1. Student 1 - Dev Joisar.....	49
11.2. Student 2 - Joshua Browne.....	50
11.3. Student 3 - Gurudeep Dhinjan.....	51
11.4. Student 4 - Lavanya Chandna.....	52
11.5. Student 5 - Keegen Wallis.....	53
12. Individual Contribution Logbook.....	54
12.1. Student 1 - Dev Joisar.....	54
12.2. Student 2 - Joshua Browne.....	54
12.3. Student 3 - Gurudeep Dhinjan.....	55
12.4. Student 4 - Lavanya Chandna.....	55
12.5. Student 5 - Keegen Wallis.....	56
13. Appendices.....	57
13.1. Github - Discord Integration.....	57
13.2. Azure Build Pipeline Artifact.....	58

1. Individual Contribution

Feature #	Feature	Description	Owner
F101	Account Registration, Login, Logout	Staff and Customer Account Creation, Login and Logout feature	Dev Joisar
F102	Account Management & Logs	Updating Account Details, Delete Accounts, Logs for Register-Login-Logout of Customer, Staff Views Customer Logs	Dev Joisar
F103	Car Catalogue	View Existing Cars Add New Cars Edit Existing Cars Remove Existing Cars	Joshua Browne
F104	Book a Test Drive	View Existing Bookings Book a Test Drive Remove an Existing Booking Adjust booking time to another available slot	Joshua Browne
F105	Order	View Customer Detail Edit Customer Address Remove Customer Address Add New Customer Address View Order History	Gurudeep Singh Dhinjan
F106	Payment	Add Payment Details Delete Payment Detail Read Payment Details Edit Customer Detail	Gurudeep Singh Dhinjan
F107	General Inquiry	Create Inquiry Delete Inquiry Update Inquiry Read Inquiry	Lavanya Chandna
F108	Wishlist	Add Car to Wishlist Read Cars in Wishlist Update which car in the Wishlist Delete Car from Wishlist	Lavanya Chandna
F109	Reporting	Create Sales Report Delete Sales Report Update Sales Report Read Sales Reports Search Sales Reports	Keegen Wallis
F110	Discussion Forum	Create new Message thread Reply to message thread Delete message thread Read message threads Search message threads	Keegen Wallis

2. Requirements

2.1. Requirements Backlog

User Story	As a/an	I want to...	So that..	Est im ate s	Priori ty	Status	Relea se	Own e r	Process/ Service
101	Staff	I want to enter my credentials	I can access the system	3	High	Completed	R0	Dev Joisar	Account Registration & Login
102	Customer	I want to enter my credentials	I can access the system	3	High	Completed	R0	Dev Joisar	Account Registration & Login
103	Customer	I want to create a account	I can access the system using my credentials	3	High	Completed	R0	Dev Joisar	Account Registration & Login
104	Staff	I want to create a account	I can access the system using my credentials	3	High	Completed	R0	Dev Joisar	Account Registration & Login
105	Staff	I want to have a logout button on homepage	I can logout whenever I want to.	3	High	Completed	R0	Dev Joisar	Account Registration & Login
106	Customer	I want to have a logout button on homepage	I can logout whenever I want to.	3	High	Completed	R0	Dev Joisar	Account Registration & Login
107	Staff	If I enter incorrect credentials	I want to be presented with error message.	1	High	Completed	R1	Dev Joisar	Account Registration & Login

108	Customer	If I enter incorrect credentials	I want to be presented with error message.	1	High	Completed	R1	Dev Joisar	Account Registration & Login
109	Customer	I want to update my user profile	I can keep it updated when details change.	3	High	Backlog	R1	Dev Joisar	Account Management & Logs
110	Staff	I want to update my user profile details	I can keep it updated when details change.	3	High	Completed	R1	Dev Joisar	Account Management & Logs
111	Staff	I want to store logs of every register-login-logout of every Customer	so that I can review during support call.	3	High	Completed	R2	Dev Joisar	Account Management & Logs
112	Staff	I want to enter user ID and filter the logs	I can see logs of user I want to	3	High	Completed	R2	Dev Joisar	Account Management & Logs
113	Customer	I want a delete button	I can delete my account when I don't need it anymore .	3	High	Completed	R2	Dev Joisar	Account Management & Logs
114	Staff	I want a delete button	I can delete my account when I don't need it anymore .	3	High	Completed	R2	Dev Joisar	Account Management & Logs

201	Customer	I Want to View the available cars for sale	I can make a choice of what to buy	5	High	Completed	R0	Joshua Browne	Car Catalogue
202	Staff	I want to List a new car for sale	I can sell cars on the platform	5	High	Completed	R0	Joshua Browne	Car Catalogue
203	Staff	I want to edit current listings	I can adjust the information when required	3	Medium	Completed	R0	Joshua Browne	Car Catalogue
204	Staff	I want to Remove a current Listing	I don't sell a car that is already sold or is not longer fit for sale	3	High	Completed	R0	Joshua Browne	Car Catalogue
205	Customer	I want to buy a car that is listed	I can complete my purchase	1	High	Completed	R0	Joshua Browne	Car Catalogue
206	Customer	I want to view all the relevant information of the car including a photo of it	I can make an informed decision regarding my purchase	8	High	Completed	R0	Joshua Browne	Car Catalogue
301	Staff	I want to see who has booked a test drive	I can prepare the cars in time	5	Medium	Completed	R1	Joshua Browne	Book Test Drive

302	Customer	I want to see what slots are available to book a test drive	I can pick a time that suits my schedule	8	High	Completed	R1	Joshua Browne	Book Test Drive
303	Customer	I want to book a test drive in a free slot	I can test drive the car I'm looking at	5	High	Completed	R1	Joshua Browne	Book Test Drive
304	Staff	I want to see what slots are still available to book a test drive	I can speak with customers regarding the available time slots	5	Medium	Completed	R2	Joshua Browne	Book Test Drive
305	Customer	I want to remove my test drive booking	I can advise if the time selected no longer works for me	3	Medium	Completed	R2	Joshua Browne	Book Test Drive
401	Customer	I want to review a summary of my order before submitting it	So that I can verify that all information is correct	4	High	Completed	R0	Gurudeep Singh Dhinjan	Order
402	Customer	I want to edit my shipping address before confirming the order	So that I can update my address if there are any changes before	4	High	Completed	R0	Gurudeep Singh Dhinjan	Order

			proceeding with the purchase.						
403	Customer	I want to place an order for the selected car	So that I can initiate the car buying process and arrange for its delivery	5	High	Completed	R0	Gurudeep Singh Dhinjan	Order
404	Customer	I want to delete my current address and add a new one during checkout	So that I can ensure my delivery address is up to date.	4	Medium	Completed	R0	Gurudeep Singh Dhinjan	Order
405	Customer	I want to review detailed information (such as price, model, and brand) of the car I've selected for ordering	So that I can confirm the car specifications before completing the purchase.	4	Medium	Completed	R1	Gurudeep Singh Dhinjan	Order
501	Customer	I can read a history of my submitted inquiries	So that I can track which cars I've inquired about.	3	Medium	Completed	R0	Lavanya Chandna	General Inquiry
502	Customer	I can create an inquiry by	So that I can request	4	High	Completed	R0	Lavanya	General Inquiry

		submitting a form with my name and contact details	information about a car I'm interested in.					Chandna	
503	Customer	I can update my inquiry details after submission	So that I can correct any mistakes.	3	High	Completed	R0	Lavanya Chandna	General Inquiry
504	Customer	I can delete my inquiry	so that I no longer receive information about a car I'm no longer interested in.	3	High	Completed	R0	Lavanya Chandna	General Inquiry
505	Customer	I want to search through my submitted inquiries by keywords,	So that I can quickly find a specific inquiry without having to manually browse through all my inquiries.	5	Medium	Completed	R1	Lavanya Chandna	General Inquiry
601	Customer	I want to view all previous payments I've made for car purchases	So that I can keep track of my financial records.	4	High	Completed	R1	Gurudeep Singh Dhinjan	Payment
602	Customer	I want to enter my	So that I can	4	High	Completed	R1	Gurudeep	Payment

		payment details (such as credit card)	complete the payment for my car purchase.					Singh Dhinjan	
603	Customer	I want to change my payment method before the transaction is finalised	So that I can switch to a different payment option if needed.	4	High	Completed	R1	Gurudeep Singh Dhinjan	Payment
604	Customer	I want to view all previous payments I've made for car purchases	So that I can keep track of my financial records.		High	Completed	R1	Gurudeep Singh Dhinjan	Payment
605	Customer	I want to remove old or unused payment methods from my account	So that I can keep my payment information secure and up to date.	3	High	Completed	R2	Gurudeep Singh Dhinjan	Payment
701	Customer	I can create a wishlist entry by adding a car	So that I can easily find it later.	4	High	Completed	R1	Lavanya Chandna	Wishlist
702	Customer	I can read all cars in my wishlist	So that I can compare them before making a purchase decision.	4	High	Completed	R1	Lavanya Chandna	Wishlist

703	Customer	I can update the details of a car in my wishlist	So that I can better organise my choices.	4	High	Completed	R1	Lavanya Chandna	Wishlist
704	Customer	I can delete a car from my wishlist	So that I can keep my list up to date.	4	High	Completed	R1	Lavanya Chandna	Wishlist
705	Customer	I can update the order of cars in my wishlist	So that I can prioritize them according to my preferences.	4	Medium	Completed	R2	Lavanya Chandna	Wishlist
801	Admin	I want to be able to create a new sales report	so that I can see a summary of the days sales, stock levels and if there are any pending transactions	4	High	Completed	R0	Keegeen Wallis	Sales Reports
802	Admin	I want to be able to view all current and previous reports	So that I can find sales reports from any given day	3	High	Completed	R0	Keegeen Wallis	Sales Reports
803	Admin	I want to be able to search through reports	So that finding reports will be more efficient	3	Medium	Completed	R1	Keegeen Wallis	Sales Reports

804	Admin	I want to be able to delete a report	so that I can remove old reports that are no longer necessary	2	High	Completed	R0	Keege n Wallis	Sales Report
805	Admin	I want to be able to update a report	So that I can fix errors or update comments as required	4	High	Completed	R0	Keege n Wallis	Sales Report
901	Customer	I want to be able to create a new message thread	so that I can ask other users of AutoMax a question about buying and listing cars.	4	High	Completed	R1	Keege n Wallis	Discussion Board
902	Customer	I want to be able to update a message thread	so that I can respond to other customers or so that I can update my question.	4	High	Completed	R1	Keege n Wallis	Discussion Board
903	User	I want to be able to view previous message threads	so that I can see if there are already answers to a question that I have	3	High	Completed	R1	Keege n Wallis	Discussion Board
904	User	I want to be able to delete a message thread	so that I can remove old or unimportant threads or a thread that I no longer want answered.	2	High	Completed	R1	Keege n Wallis	Discussion Board

905	Customer	I want to be able to search through the discussion board	so that I can find message threads that I'm looking for.	3	Medium	Completed	R2	Keegeen Wallis	Discussion Board
-----	----------	--	--	---	--------	-----------	----	----------------	------------------

2.2. Applying INVEST Criteria to Each User Story

Group Member	Feature	User Story	INVEST Criteria Application
Dev Joisar	Account Management & Logs	As a staff member, I want to store logs of every register-login-logout of every customer so that I can review during support calls.	<p>Independent: Storing logs is separate from other actions.</p> <p>Negotiable: Can adjust the details stored in the logs.</p> <p>Valuable: Important for customer support.</p> <p>Estimatable: Clearly estimatable.</p> <p>Small: Achievable in a sprint.</p> <p>Testable: Test whether logs are stored and retrievable.</p>
Joshua Browne	Car Catalogue	As a customer, I want to view available cars for sale so that I can make a choice of what to buy.	<p>Independent: Viewing cars is standalone.</p> <p>Negotiable: Options for filtering cars.</p> <p>Valuable: Crucial for decision-making.</p> <p>Estimatable: Clear estimation possible.</p> <p>Small: Fits within a sprint.</p> <p>Testable: Ensure the cars are displayed and filter options work.</p>

Gurudeep Singh Dhinjan	Order	<p>As a customer, I want to review a summary of my order before submitting it so that I can verify all information is correct.</p>	<p>Independent: Reviewing an order is a separate task.</p> <p>Negotiable: Different levels of details can be included.</p> <p>Valuable: Critical for confirming order details.</p> <p>Estimatable: Estimatable in terms of time and effort.</p> <p>Small: Can be completed in a sprint.</p> <p>Testable: Test the order summary and submission process.</p>
Lavanya Chandna	Wishlist	<p>As a customer, I want to add a car to my wishlist so that I can easily find it later.</p>	<p>Independent: Adding a car to the wishlist is standalone.</p> <p>Negotiable: Can include priority levels or other options.</p> <p>Valuable: Helps track favorite cars.</p> <p>Estimatable: Clear estimation.</p> <p>Small: Fits in a sprint.</p> <p>Testable: Ensure cars can be added and retrieved from the wishlist.</p>
Keegen Wallis	Discussion Board	<p>As a customer I want to be able to create a new message thread so that I can ask other users of AutoMax a question about buying and listing cars.</p>	<p>Independent: Addresses only the functionality of creating a discussion thread</p> <p>Negotiable: A variety of discussion threads can be created</p> <p>Valuable: Allows customers to make more informed purchases, increasing customer loyalty with AutoMax</p> <p>Estimatable: Easily done through monitoring its overall use and the number of replies a thread can expect to receive</p> <p>Small: Categorises threads by</p>

			<p>title, achievable in one sprint</p> <p>Testable: Test if the thread is created, if the correct data is displayed and if child threads are attached to their parent.</p>
--	--	--	---

2.3. Splitting User Stories for INVEST

Group Member	Feature	Original User Story	Splitting Using INVEST
Dev Joisar	Account Management & Logs	As a staff member, I want to store logs of every register-login-logout of every customer so that I can review during support calls.	<p>Split into two stories:</p> <ol style="list-style-type: none"> As a staff member, I want to store logs of every register, login, and logout. As a staff member, I want to filter logs by user ID for quicker search.
Joshua Browne	Car Catalogue	As a customer, I want to view available cars for sale so that I can make a choice of what to buy.	<p>Split into two stories:</p> <ol style="list-style-type: none"> As a customer, I want to filter available cars by brand and model. As a customer, I want to view details of each car, including price and specifications.
Gurudeep Singh Dhinjan	Order	As a customer, I want to review a summary of my order before submitting it so that I can verify all information is correct.	<p>Split into two stories:</p> <ol style="list-style-type: none"> As a customer, I want to see a detailed order summary including price, item, and shipping details. As a customer, I want the ability to edit my shipping address before submitting the order.
Lavanya Chandna	Wishlist	As a customer, I want to add a car to my wishlist so that I can easily find it later.	<p>Split into two stories:</p> <ol style="list-style-type: none"> As a customer, I want to add cars to my wishlist and prioritize them by preference. As a customer, I want to remove cars from my wishlist once they are no longer needed.

Keegen Wallis	Sales Reports	As an admin, I want to create a new sales report so that I can see a summary of the day's sales.	Split into two stories: 1. As an admin , I want to create a sales report showing total sales and pending transactions. 2. As an admin , I want to filter sales reports by date and generate them accordingly.
---------------	---------------	--	---

- **Updates compared with release 1**
 - No changes as compared to release 1.

2.4. Non-Functional Requirements (NFR)

User Story	As a/an	I want to...	So that...	Estimate	Priority	Release	Owner	Process
Security Requirements								
NFR1001	Staff	Ensure my password is masked when typing in login page	I can securely access the system	1	High	R0	Dev Joisar	Account Registration & Login
NFR1002	Staff	ensure my password is masked durin registration	I can securely access the system	1	High	R0	Dev Joisar	Account Registration & Login
NFR1003	Staff	ensure my session is terminated when logging out	I can prevent session hijacking	1	High	R0	Dev Joisar	Account Registration & Login
NFR1001	Customer	Ensure my password is masked when typing in login page	I can securely access the system	1	High	R0	Dev Joisar	Account Registration & Login
NFR1002	Customer	ensure my password is masked durin registration	I can securely access the system	1	High	R0	Dev Joisar	Account Registration & Login
NFR1003	Customer	ensure my session is terminated when logging out	I can prevent session hijacking	1	High	R0	Dev Joisar	Account Registration & Login
NFR2001	Customer	I want the page to load very quickly so I can browse the available cars	I don't spend a long time waiting for slow pages	1	Medium	R0	Joshua Brown e	Car Catalogue
NFR2002	Staff	I want to ensure only staff can adjust the available cars	the customers don't change the pricing or features of a vehicle	1	High	R0	Joshua Brown e	Car Catalogue

NFR2003	Staff	I want to ensure only staff can list new cars for sale	the customers don't add a car at a very low price	1	High	R0	Joshua Brown e	Car Catalogue
NFR3001	Customer	Only be able to place an order when logged in	My personal and payment details are secure	2	High	R0	Gurudeep Singh Dhinjan	Order
NFR3002	Customer	Save my updated address automatically	The system remembers my latest address for future orders	2	High	R0	Gurudeep Singh Dhinjan	Order
NFR3003	Customer	Ensure address validation follows strict format	The delivery is processed correctly	2	High	R0	Gurudeep Singh Dhinjan	Order
NFR3004	Customer	Be automatically redirected to login if not authenticated	My session is managed securely	1	High	R0	Gurudeep Singh Dhinjan	Order
NFR4001	Customer	I want to be required to log in before submitting an inquiry	so that my inquiry is associated with my account, and I can track its status	1	High	R0	Lavany Chandna	General Inquiry
NFR5001	Admin	I want the user interface to be intuitive	so that I can quickly learn and navigate systems	1	High	R0	Keegan Wallis	Sales Reports
NFR5002	Admin	I want a load time of less than 2 seconds	so that I am able to find or create a report quickly	1	High	R0	Keegan Wallis	Sales Reports

NFR5003	Developer	I want the code to be maintainable with good documentation and code comments	so that it is easy for me to add updates and new features	2	Medium	R0	Keegan Wallis	Sales Reports
NFR6001	Customer	Ensure that the wishlist loads within 2 seconds	I can quickly review and manage my wishlist without delays	3	High	R1	Lavanya Chandna	Wishlist
NFR6002	Customer	Ensure only logged-in users can access and modify the wishlist	Unauthorized users cannot make changes or view my wishlist	1	High	R1	Lavanya Chandna	Wishlist
NFR6003	Customer	The wishlist should update in real-time when changes are made	I can see the changes immediately without refreshing the page	3	Medium	R1	Lavanya Chandna	Wishlist
NFR7001	Customer	Ensure that payment details are encrypted and transmitted securely	My sensitive financial data remains protected	2	High	R1	Gurudeep Singh Dhinjan	Payment
NFR7002	Customer	Ensure that only logged-in users can proceed with a payment	Payment is tied to my user account and can be tracked	1	High	R1	Gurudeep Singh Dhinjan	Payment
NFR7003	Customer	Provide a confirmation and error message during the payment process	I know whether the transaction was successful or not	2	High	R1	Gurudeep Singh Dhinjan	Payment

NFR7004	Customer	Ensure the system logs all payment attempts and outcomes	I can review my payment history for verification and tracking	2	High	R1	Gurudeep Singh Dhinjan	Payment
---------	----------	--	---	---	------	----	------------------------	---------

- **Updates compared with release 1**
 - No changes as compared to release 1.

2.5. Data Requirements

Ref.	Data Element	Format	Length	Description
1. User				
DE-1001	userID	Int	-	Unique ID for registered user.
DE-1002	userEmail	Varchar	30	User's email address
DE-1003	userPassword	Varchar	100	Secured password entered by user
DE-1004	userPhoneNumber	Integer	10	User's phone number
DE-1005	userFirstName	Char	30	User's first name
DE-1006	userLastName	Char	30	User's last name
DE-1007	userStreetNumber	Integer	10	User's Location Street Number
DE-1008	userStreetName	Char	40	User's Location Street Name
DE-1009	userSuburb	Char	40	User's Location's Suburb Name
DE-1010	userCity	Char	40	User's Location's City
DE-1011	userState	Char	40	Users Location's State
DE-1012	userPostcode	Char	4	User' Location's Postcode
2. Staff				
DE-2001	staffID	Int	-	Unique Staff ID Number
DE-2002	staffEmail	Varchar	30	Staff's Email Address
DE-2003	staffPassword	Varchar	100	Staff's Login Password
DE-2004	staffPhoneNumber	Integer	10	Staff's phone number
DE-2005	staffFirstName	Char	30	Staff's first name
DE-2006	staffLastName	Char	30	Staff's last name
3. User Logs				
DE-3001	UserID	Int	-	Unique ID for registered user.
DE-3002	Status	Varchar	15	Type of transaction (Register, Login, or Register)
DE-3003	CurrentTime	Timestamp	-	Time stamp of transaction.
4. Car				
DE-4001	CarID	Int	-	Unique ID for each Car to sell
DE-4002	SellPrice	Double	-	Price of the car for sale
DE-4003	PhotoID	Int	-	ID of the photo that is used for the listed car
DE-4004	Brand	Varchar	255	Brand of the car for sale
DE-4005	Model	Varchar	255	Model of the car for sale
DE-4006	BodyType	Varchar	255	BodyType of the car for sale
DE-4007	Year	Int	-	Year the listed car was built
DE-4008	Kilometres	Int	-	Number of kilometres on the clock of the car for sale
DE-4009	Doors	Int	-	Number of doors on the car for sale
DE-4010	Motor	Varchar	255	Type of Motor in the car for sale
DE-4011	Transmission	Varchar	255	Type of transmission in the car for sale
DE-4012	Driveline	Varchar	255	Type of driveline in the car for sale
DE-4013	SoldRemoved	Int	-	Whether the car is still visible on the site or if it has been sold/removed
5. Order				
DE-5001	OrderID	Int	-	Unique ID for order.
DE-5002	UserID	Int	-	Unique ID for registered user.
DE-5003	CarID	Int	-	Unique ID for each Car to sell
DE-5004	CreatedAt	Timestamp	-	Timestamp of transaction.

DE-5005	PaymentMethodID	Int	-	Unique ID for the payment method used in the order, linked to the Payment table.
DE-5006	OrderAmount	DECIMAL(10,2)	-	Total amount of the order.
6. General Inquiry				
DE-6001	InquiryID	Int	-	Unique identifier for each inquiry, auto-incremented.
DE-6002	UserID	Int	-	Identifier for the user who submitted the inquiry.
DE-6003	InquiryMessage	Text	-	The message or question submitted by the user.
DE-6004	PreferredCallbackTime	Varchar	20	User's preferred time for receiving a callback.
DE-6005	Status	Varchar	20	The current status of the inquiry (e.g., Pending, Resolved).
DE-6006	CreatedAt	Timestamp	-	The timestamp when the inquiry was created, defaults to the current timestamp.
7. Sales Reporting				
DE-7001	ReportID	Int	-	Unique ID for each report, auto-incremented
DE-7002	Title	Varchar	20	The name of the sales report
DE-7003	TotalSales	Int	-	The total number of sales for today
DE-7004	PendingTransactions	Bool	-	If any of these sales still have transactions that are not complete
DE-7005	LastEdited	Timestamp	-	The last time that the sales reported was opened and updated
DE-7006	Comments	Varchar	50	Space for admin to leave additional notes about the sales report
8. Payment				
DE-8001	PaymentMethodID	INT	-	Unique identifier for each payment method.
DE-8002	UserID	INT	-	Unique ID for the user making the payment, linked to the Users table.
DE-8003	CardNumber	VARCHAR(16)	16	Credit card number used for payment (should be securely handled).
DE-8004	CardExpiryDate	VARCHAR(5)	5	Expiry date of the credit card (MM/YY format).
DE-8005	CardCVV	VARCHAR(3)	3	CVV number of the credit card (should be securely handled).
9. Wishlist				
DE-9001	WishlistID	INT	-	Unique identifier for each wishlist entry.
DE-9002	UserID	INT	-	Unique ID for the user creating the wishlist, linked to the Users table.
DE-9003	CarID	INT	-	Unique ID for the car added to the wishlist, linked to the Car table.
DE-9004	DateAdded	TIMESTAMP	-	Timestamp of when the car was added to the wishlist.
DE-9005	PriorityLevel	ENUM	-	Priority level of the wishlist entry (High, Medium, Low).
10. Test Drive Booking				
DE-10001	BookingID	INT	-	Unique identifier for each Booking entry.

DE-10002	UserID	INT	-	Unique ID for the user creating the Booking
DE-10003	CarID	INT	-	Unique ID for the car added to the Booking
DE-10004	BookingDate	Date	-	Date the Booking was made
DE-10005	BookingTime	Time	-	Time the Booking was made
11. Discussion Threads				
DE-11001	ThreadID	INT	-	Unique id for each thread
DE-11002	UserID	INT	-	Foreign key referencing the UserID of the user who created the thread
DE-11003	timeCreated	Date	-	The time that the thread was created
DE-11004	title	VARCHAR	50	The title of the discussion thread
DE-11005	message	VARCHAR	250	The body of the thread, question or contents that the user wants to add
DE-11006	ParentThreadID	INT	-	If the thread is a new thread, this is set to 0. If the thread is a reply to a different thread, this number is set to the ThreadID of its parent.

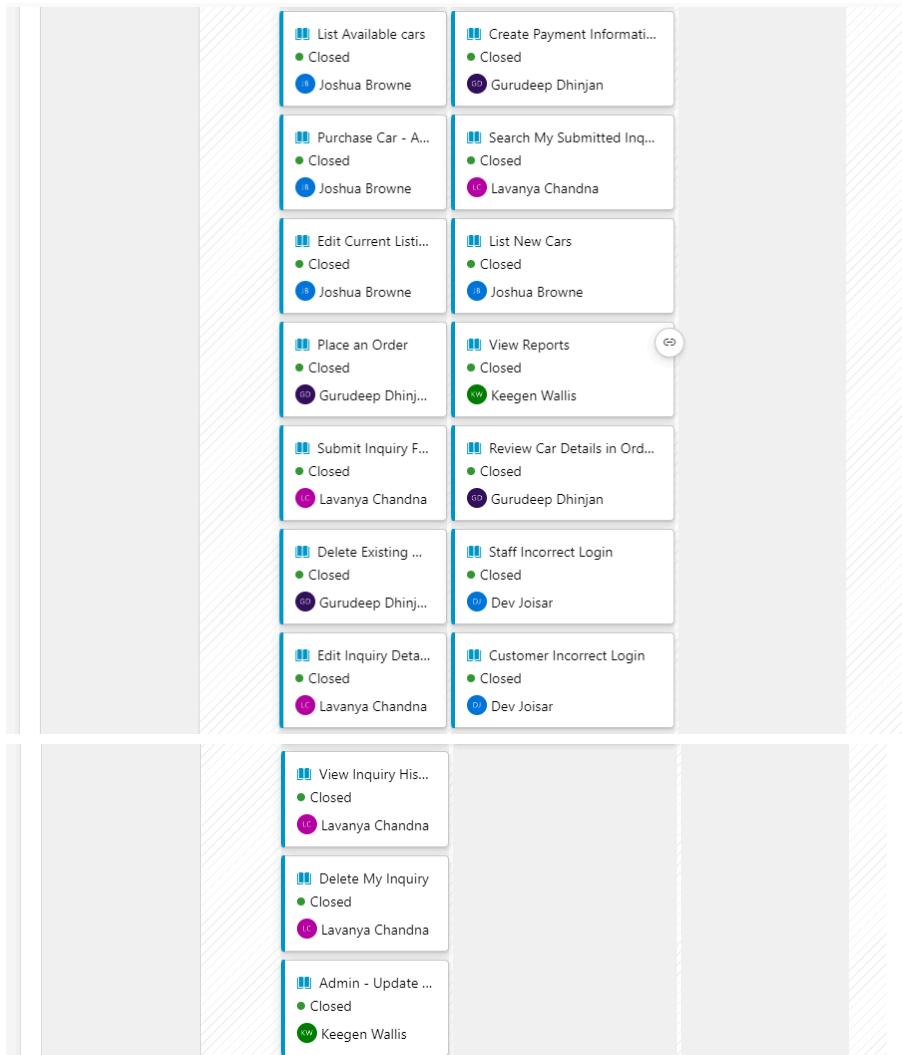
- **Updates compared with release 1**
 - No changes as compared to release 1.

3. Project Schedule

Azure Project Plan Board

ASD Delivery Plan

Teams	August	September	October
ASD Team Stories	Sprint 1 19/08 - 08/09 + Customer Login ● Closed ● Dev Joisar	Sprint 2 09/09 - 06/10 + Staff Delete Account ● Closed ● Dev Joisar	Sprint 3 07/10 - 27/10 + Customer Delete... ● Closed ● Dev Joisar
	 Admin Login ● Closed ● Dev Joisar	 Admin - Update Profile ● Closed ● Dev Joisar	 Filter logs ● Closed ● Dev Joisar
	 Order Summary ... ● Closed ● Gurudeep Dhinj...	 Customer - Update Profile ● Closed ● Dev Joisar	 Logs ● Closed ● Dev Joisar
	 Edit Existing Ad... ● Closed ● Gurudeep Dhinj...	 Add Car to Wishlist ● Closed ● Lavanya Chandna	 Payment History ● Closed ● Gurudeep Dhinj...
	 Customer Registr... ● Closed ● Dev Joisar	 See Available Test Drive B... ● Closed ● Joshua Browne	 See Bookings Ca... ● Closed ● Joshua Browne
	 Customer Logout ● Closed ● Dev Joisar	 View Booked Test Drives ● Closed ● Joshua Browne	 Read Payment H... ● Closed ● Gurudeep Dhinj...
	 Admin Logout ● Closed ● Dev Joisar	 View Wishlist ● Closed ● Lavanya Chandna	 Delete Existing T... ● Closed ● Joshua Browne
	 Admin Registratio... ● Closed ● Dev Joisar	 Book Test Drive - As a Cu... ● Closed ● Joshua Browne	 Reorder Wishlist ● Closed ● Lavanya Chandna
	 Search Reports ● Active ● Keegen Wallis	 Delete Payment Method ● Closed ● Gurudeep Dhinjan	
	 Admin - Delete ... ● Closed ● Keegen Wallis	 Edit Wishlist Item ● Closed ● Lavanya Chandna	
	 Admin - Create ... ● Closed ● Keegen Wallis	 Update Payment Method ● Closed ● Gurudeep Dhinjan	
	 Detailed Info Page ● Closed ● Joshua Browne	 Delete Car from Wishlist ● Closed ● Lavanya Chandna	



- **Updates compared with release 1**
 - No changes as compared to release 1.

4. Cost

Release	Process	Owner	Estimated Hours	Cost @ \$80 per person per hour
R0	Account Registration & Login	Dev Joisar	20	\$1,600
	Car Catalogue	Joshua Browne	25	\$2,000
	Order	Gurudeep Dhinjan	21	\$1,680
	Customer Inquiry Form	Lavanya Chandna	20	\$1,600
	Reporting	Keegen Wallis	20	\$1,600
R1	Account Registration & Login	Dev Joisar	2	\$160
	Car Catalogue	Joshua Browne	4	\$320
	Order	Gurudeep Dhinjan	3	\$240
	Customer Inquiry Form	Lavanya Chandna	2	\$160
	Reporting	Keegen Wallis	3	\$240
	Account Management & Logs	Dev Joisar	20	\$1,600
	Payment	Gurudeep Dhinjan	18	\$1,440
	Book Test Drive	Joshua Browne	20	\$1,600
	Wishlist	Lavanya Chandna	20	\$1,600
	Discussion Forum	Keegen Wallis	15	\$1,200
R2	Account Management & Logs	Dev Joisar	5	\$400
	Payment	Gurudeep Dhinjan	3	\$240
	Book Test Drive	Joshua Browne	5	\$400
	Wishlist	Lavanya Chandna	2	\$160
	Discussion Forum	Keegen Wallis	5	\$400
TOTAL				\$18,640

- **Updates compared with release 1**
 - No changes as compared to release 1.

5. Constraints and Risks

Constraints		
Project Planning	Time Allocation & Due Date	Maintain Budget
No clarity of project plans among team member during each iteration phase	Team member not allocating enough time to the project	Incorrect estimation of project budget
Not meeting the Minimum Requirement during each iteration	Non-allocation of minimum required time for each iteration	Team member submitting timesheet with overtime
No weekly follow-ups for update on assigned task	Members not clear on due date for final submission date	Additional cost due to unknown or unexpected circumstances

Risk		
Human Factor Risk	Data Security Risk	Hardware Risk
Sudden illness of a team member	Theft of Code	Non-compatible hardware of a person
Dropping the project in midway	Breach of remote repository on GitHub Platform	Theft of member's hardware device being used
Non-cooperative team member	Sharing the confidential details to other team members - include code & documents	Misplace of a member's hardware device being used

- **Updates compared with release 1**
 - No changes as compared to release 1.

6. Security in the Software

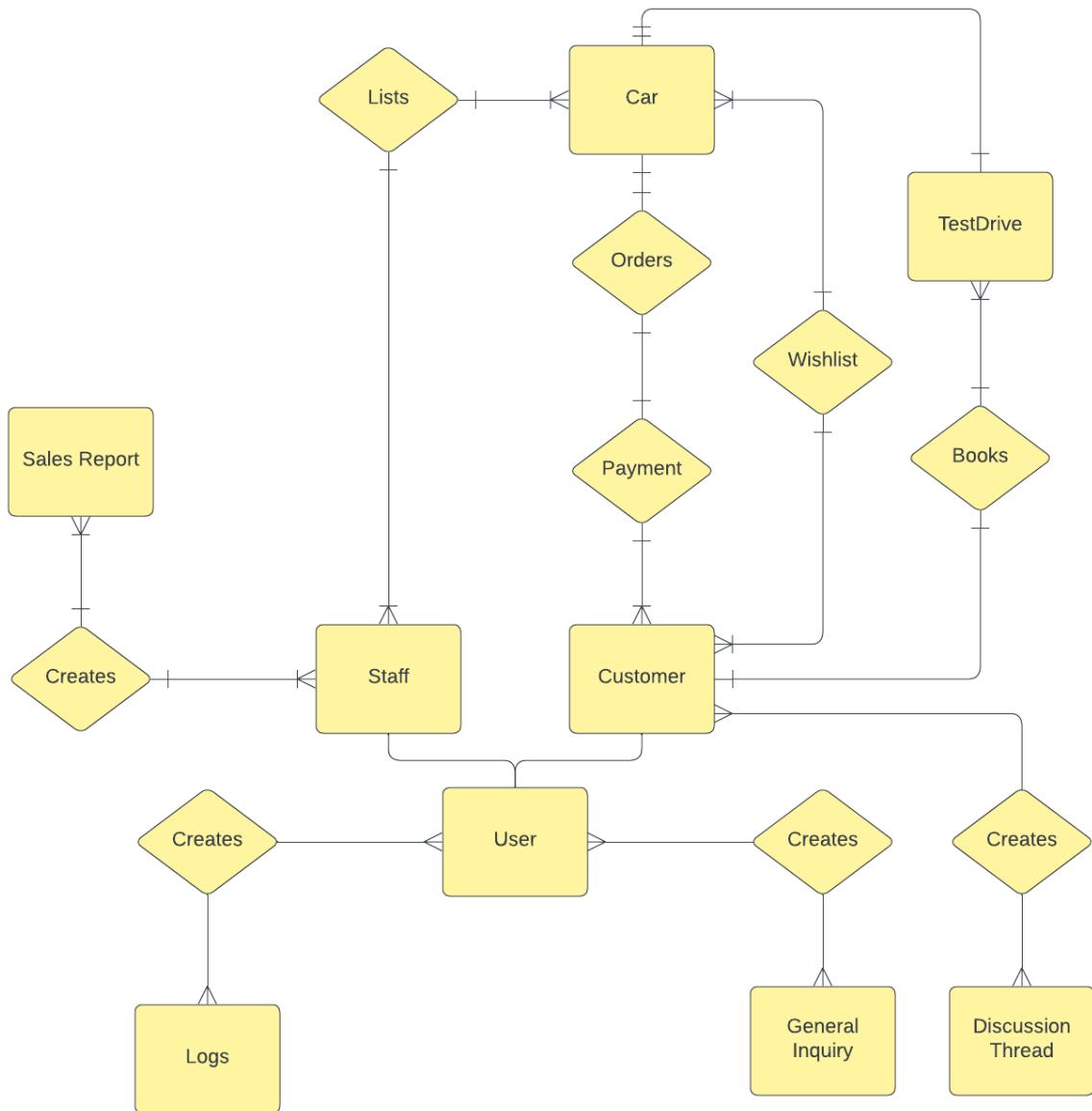
Student Name	Feature	Security Aspects Implemented	Security Aspects Not Implemented
Dev Joisar	Account Registration & Login	<ul style="list-style-type: none"> - Session management with proper logout - Masking passwords during input - SQL Injection prevention with input validation 	<ul style="list-style-type: none"> - Two-factor authentication (2FA) - Password encryption with SHA-256
	Account Management & Logs	<ul style="list-style-type: none"> - Logs for login/register/logout stored securely - Staff-only access for user logs - SQL Injection prevention with input validation 	No anomaly detection for unusual login attempts
Joshua Browne	Car Catalogue	<ul style="list-style-type: none"> - SQL Injection prevention with input validation 	No audit trail for changes in car listings
	Book Test Drive	<ul style="list-style-type: none"> - SQL Injection prevention with input validation - Restricted access so customers can only see their own bookings while admins can view all 	No booking confirmation emails sent to customers
Gurudeep Dhinjan	Order	<ul style="list-style-type: none"> - Address validation - Orders restricted to authenticated users - SQL Injection prevention with input validation 	No fraud detection mechanism
	Payment	<ul style="list-style-type: none"> - Non-null constraints in DB to prevent incomplete data - SQL Injection prevention with input validation 	<ul style="list-style-type: none"> - No payment tokenization or PCI-DSS compliance - Encrypted payment information

			transmission - Payment encryption with SHA-256
Lavanya Chandna	Customer Inquiry Form	- Input validation to block SQL injection attacks - Restricted access based on user authentication	No CAPTCHA for spam prevention
	Wishlist	- Restricted access based on user authentication - SQL Injection prevention with input validation	No notifications or alerts for wishlist changes
Keegen Wallis	Reporting	- Admin-only access - Logging of report access - SQL Injection prevention with input validation	No automatic backup of reports
	Discussion Board	- User authentication required for thread creation - SQL Injection prevention with input validation	No content moderation for spam filtering

7. Solution Architecture

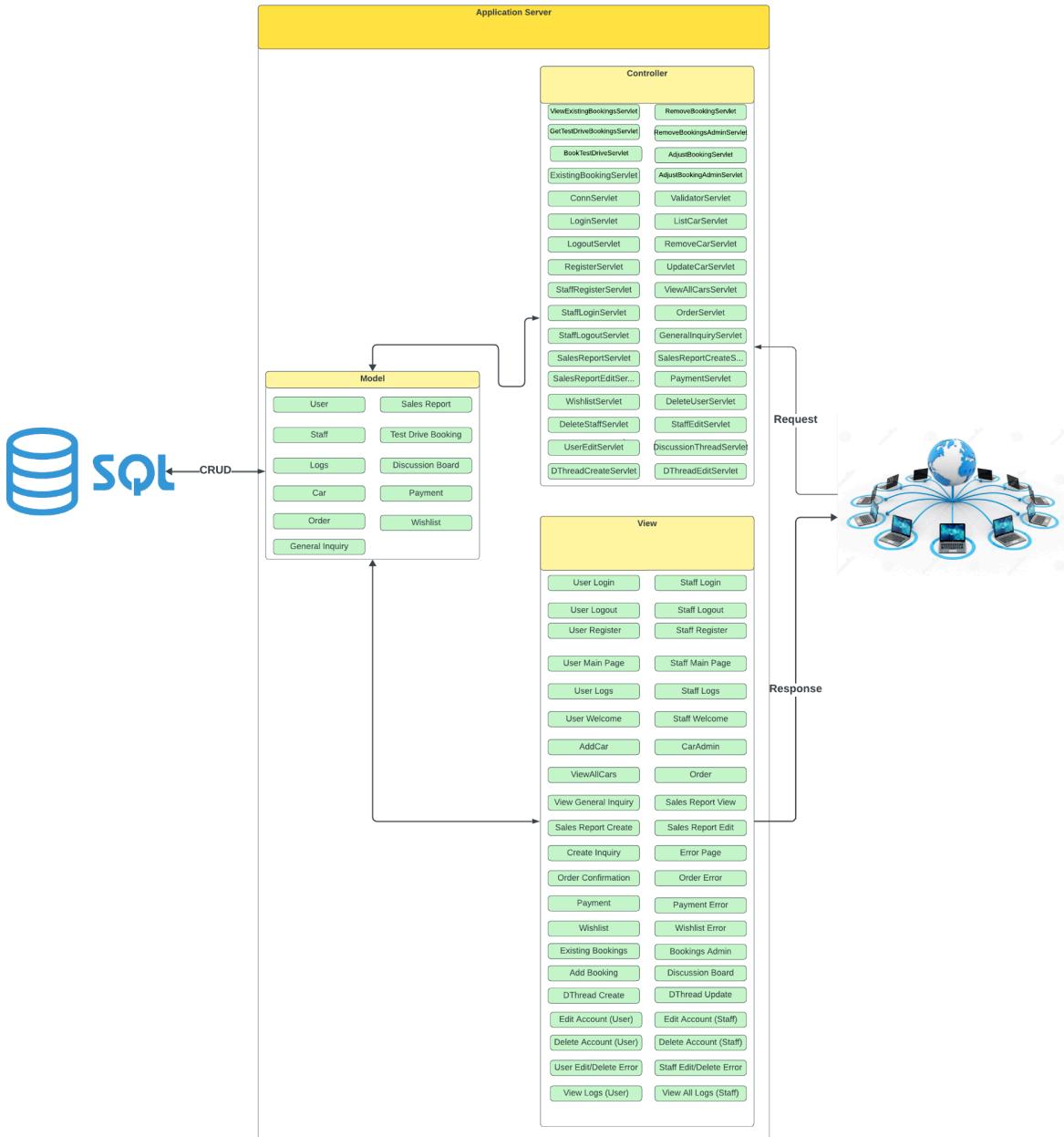
7.1. Data Architecture Model

https://lucid.app/lucidchart/7bdcb3c9-aa83-4f16-83df-fd6d6f256e41/edit?viewport_loc=-127%2C35%2C2303%2C1002%2C0_0&invitationId=inv_8d673a3e-a510-495d-8b3f-13973b013be4



7.2. Application Architecture Model

https://lucid.app/lucidchart/7bdcb3c9-aa83-4f16-83df-fd6d6f256e41/edit?viewport_loc=-37%2C-101%2C2514%2C1280%2C1uZ5fg0Q-CsS&invitationId=inv_8d673a3e-a510-495d-8b3f-13973b013be4



- **Updates compared with release 1**
 - No changes as compared to release 1.

Reflection on the content in "Week 8 workshop - Advanced Software Architecture and Design Patterns":

The current project uses the **Model-View-Controller (MVC)** architecture pattern, which organises the application into three interconnected components: **Model**, **View**, and **Controller**. Below is a detailed reflection on how each component has been applied in this project:

- **Model:**
 - This layer handles all data-related logic. All primary entities involved in the car dealership system—such as User, Staff, Car, Logs, Order, Sales Report, Test Drive Booking, Wishlist, Payment, and General Enquiry—reside in this layer. The models interact directly with the SQL database for **CRUD** (Create, Read, Update, Delete) operations, ensuring data persistence and integrity across the system.
- **View:**
 - The view layer manages the user interface (UI) by rendering data from the model and presenting it to the user. It includes pages for logging in, managing user and staff accounts, handling bookings, payments, and wishlist management. Additionally, the UI now supports dynamic features like activity logs, discussion boards, and error handling for complex interactions such as payments and wishlists.
- **Controller:**
 - This layer acts as an intermediary between the Model and View layers. It contains servlets for handling requests related to all model entities. The controller takes user input, processes it via business logic in the model, and then updates the views to reflect the new or modified data. With the updates in **r1**, the controller now includes servlets to handle operations for new features like test drive bookings, discussion threads, wishlist management, and payments.

Alternative Considerations:

While the **MVC** architecture is robust and well-suited for web applications like the one we're building, our team could have considered the **Layered Architecture Pattern** as an alternative. This pattern organises the application into layers, such as presentation, business logic, and data access. While this pattern is particularly beneficial for large, complex applications, we chose MVC for the following reasons:

- **Clear Separation of Concerns:**
 - MVC provides a clear separation between data (**Model**), UI (**View**), and control logic (**Controller**), making it easier to manage and scale the application over time.
- **Reusability:**
 - Components, including views and controllers, can be reused across different application parts, reducing the need to duplicate code.
- **Modularity:**

- With MVC, changes to one part of the application (e.g., UI) can be made independently of the others (e.g., business logic in the controller), improving flexibility and maintainability.

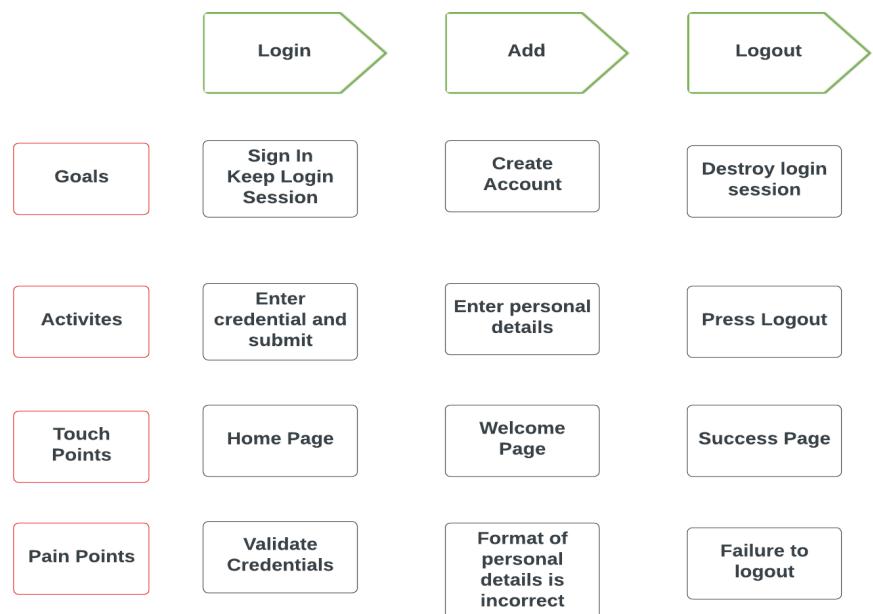
The MVC architecture has proven to be a good fit for our project's needs, offering the necessary structure to accommodate the new features introduced in **r1** while maintaining scalability and ease of maintenance.

8. Solution Design

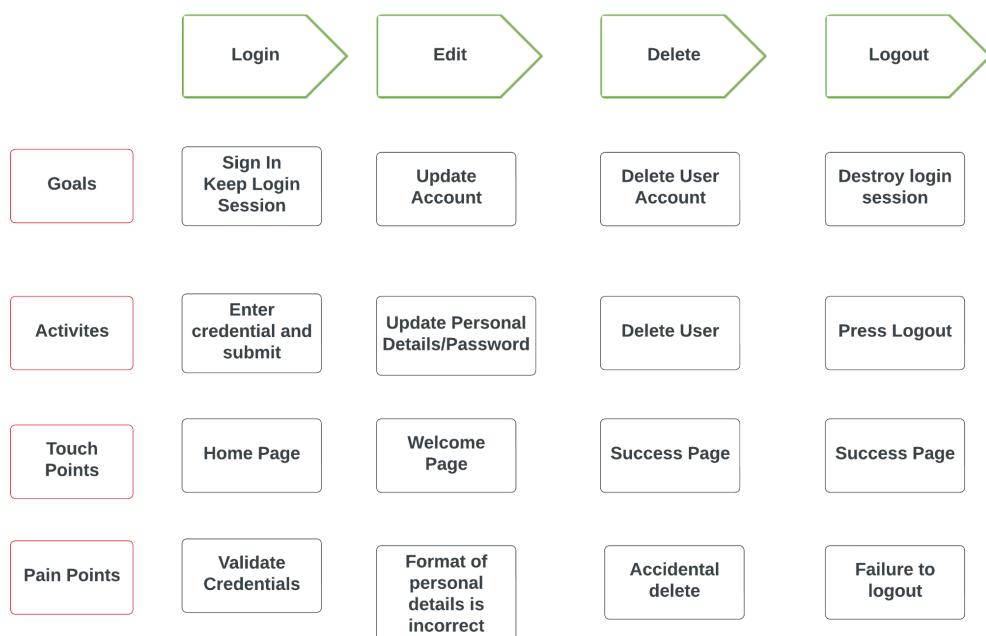
8.1. Interaction Design Model

https://lucid.app/lucidchart/7bdcb3c9-aa83-4f16-83df-fd6d6f256e41/edit?view_items=8ZZ5.z~8PITO&invitationId=inv_8d673a3e-a510-495d-8b3f-13973b013be4

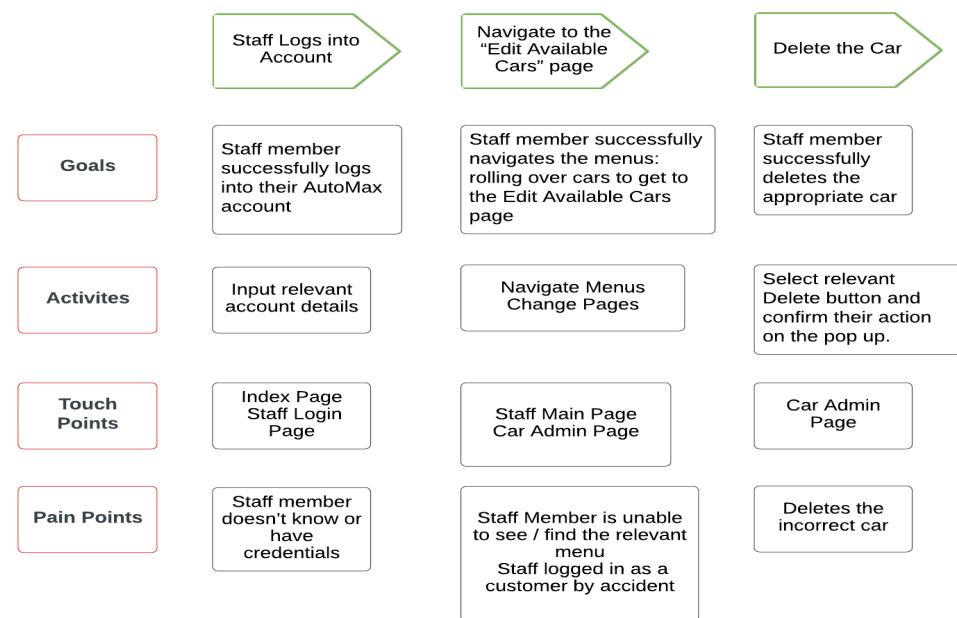
Account Registration, Login, Logout



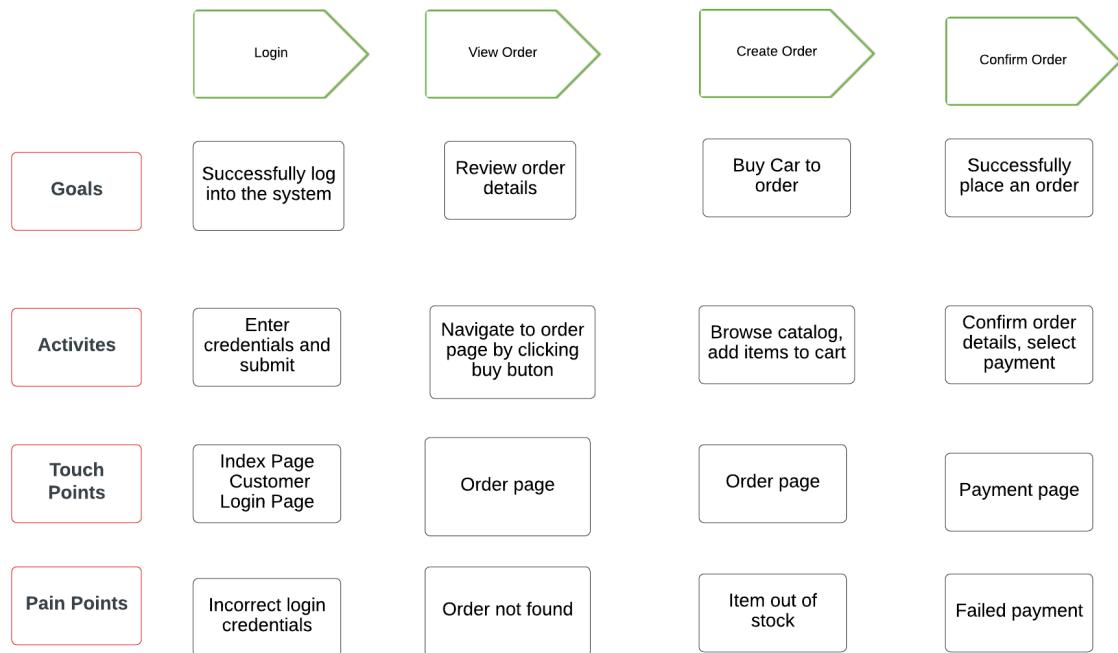
Account Management



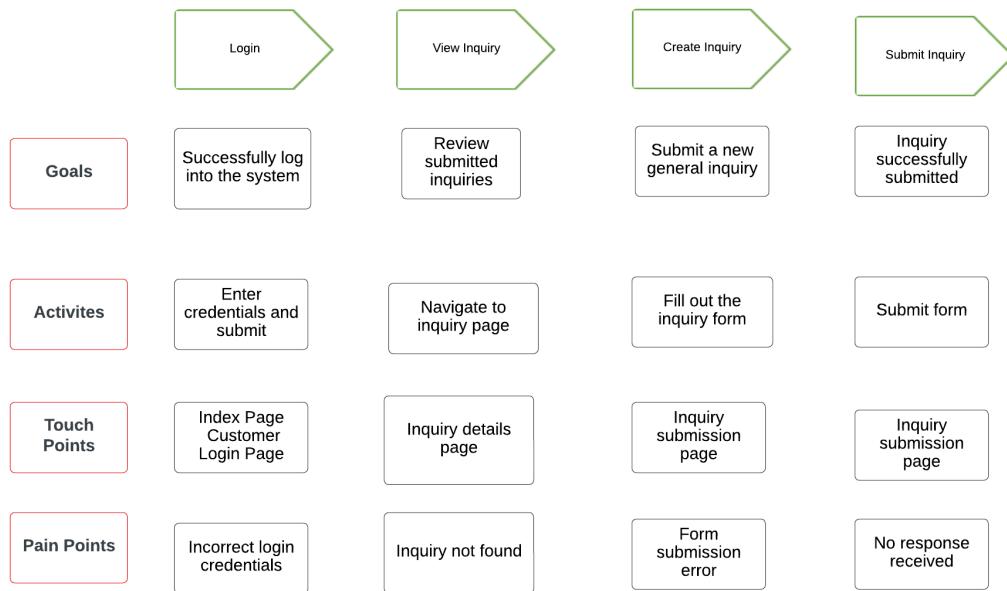
Staff Login to Delete a Car



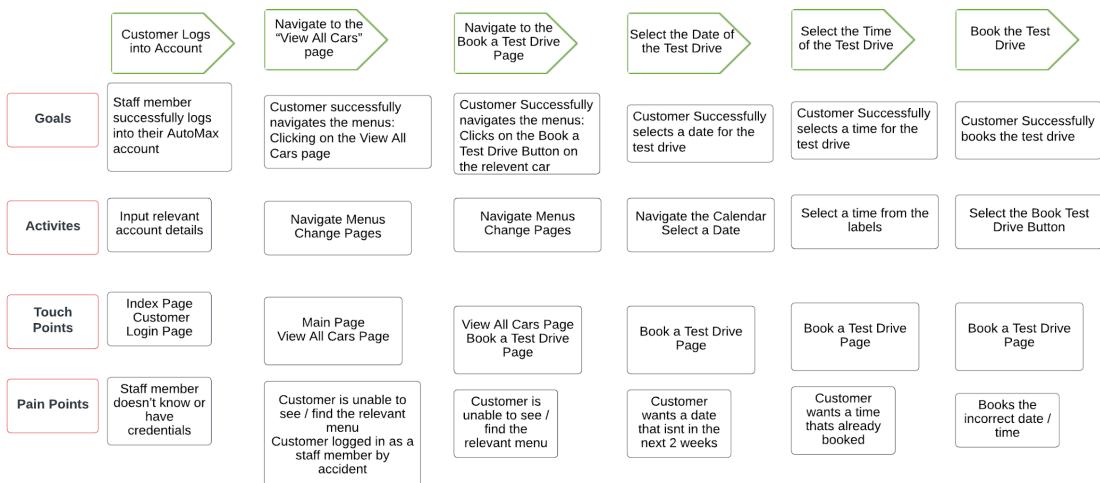
Orders Interaction Design Model



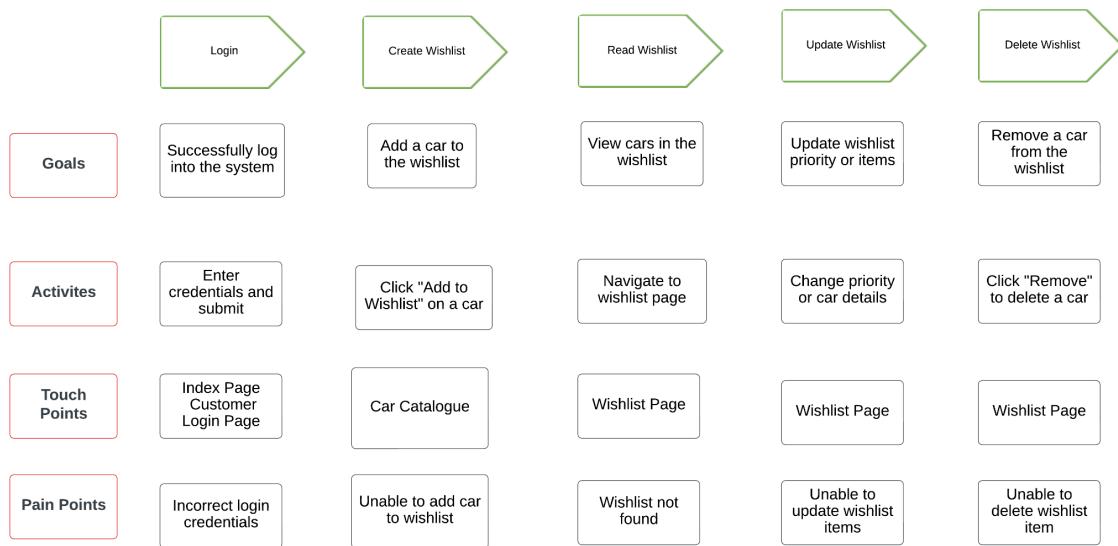
General Inquiry Interaction Design Model



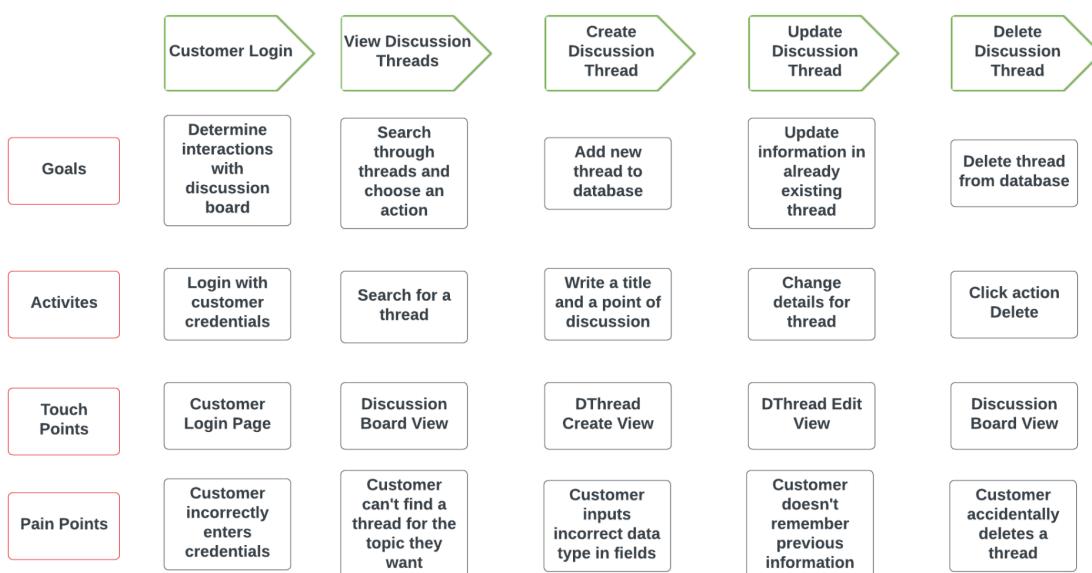
Customer Login to Book a Test Drive



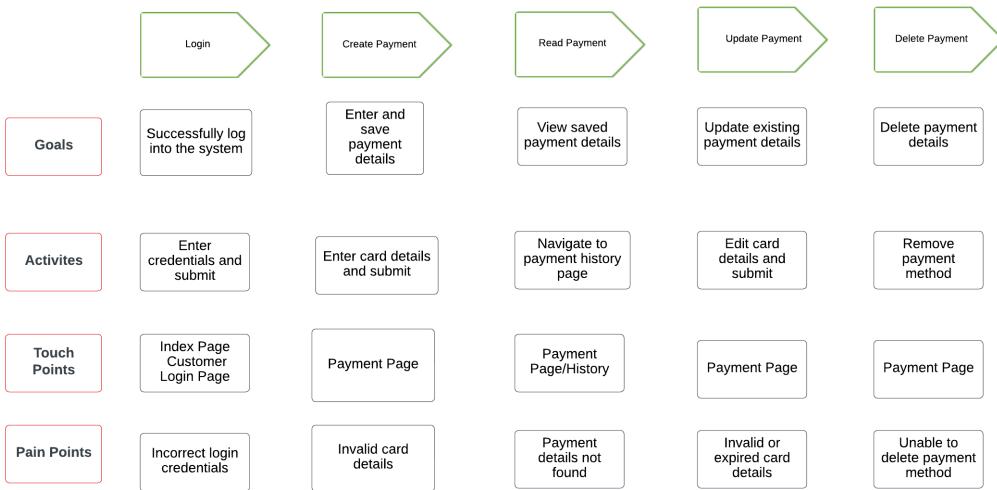
Wishlist Interaction Design Model



Discussion Board Interaction Design Model



Payment Interaction Design Model

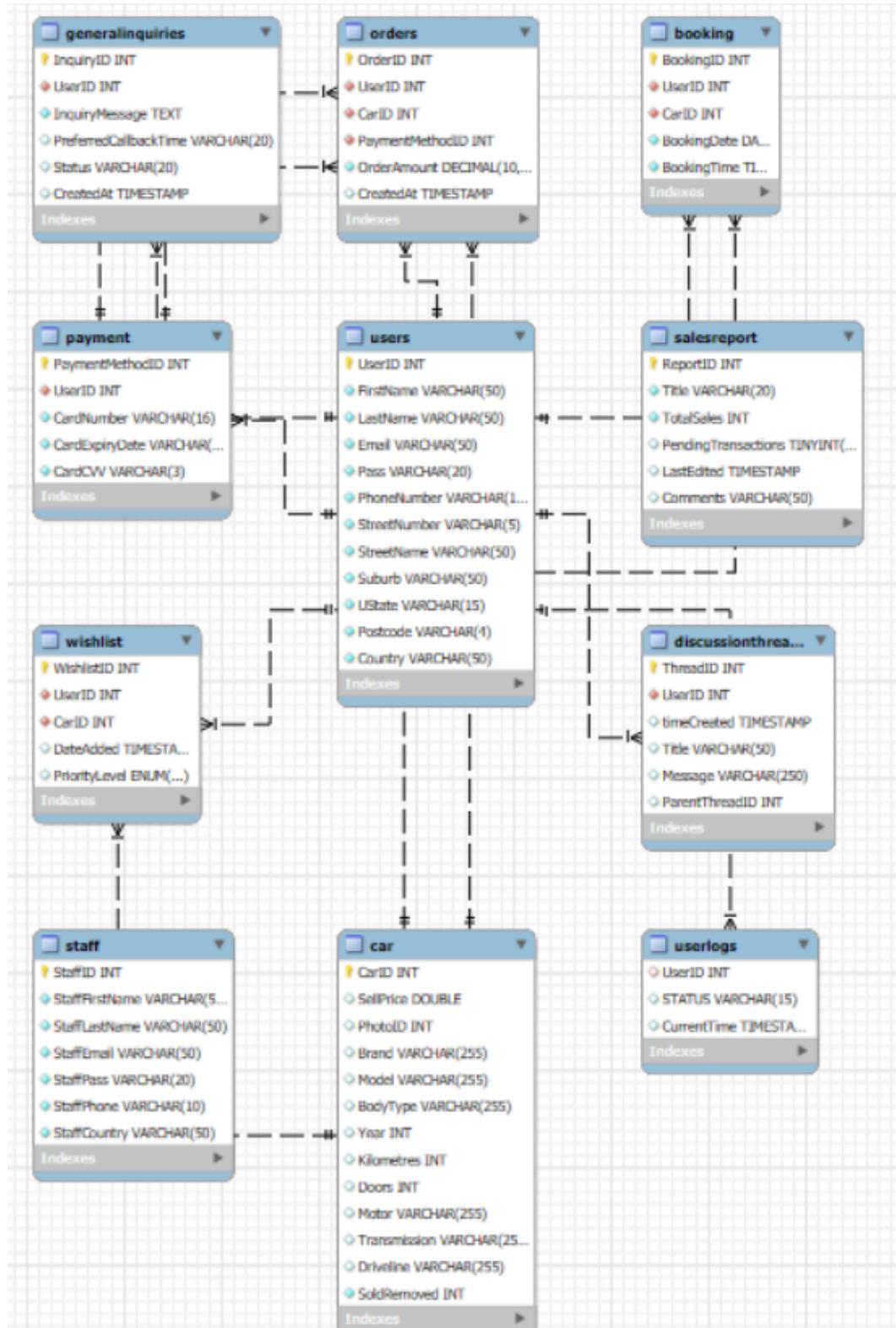


- **Updates compared with release 1**
 - No changes as compared to release 1.

- **Reflection on the content in "Week 8 workshop - Advanced Software Architecture and Design Patterns"**
 - We have used below design patterns in our project
 - Singleton Pattern - A single instance of a database connection manager is being created and shared across the application to handle SQL operations efficiently.
 - Factory Pattern - Primarily being used for creating instances of different models including User, Staff, Order, Payment. This pattern allows us encapsulating the instantiation logic, making it easier to manage changes in object creation.
 - Observer Pattern - For all the models as in factory pattern, the corresponding pages viewed by the user are constantly being updated for every transaction that user performs. This pattern complements the MVC by enhancing our dynamic data handling.

8.2. Backend Database Model

https://lucid.app/lucidchart/7bdcb3c9-aa83-4f16-83df-fd6d6f256e41/edit?view_items=qs05ryNqU~5f&invitationId=inv_8d673a3e-a510-495d-8b3f-13973b013be4



9. Testing Design

This section outlines the software testing results for various DAO (Data Access Object) tests. The testing was conducted to ensure full functionality, data integrity, and system reliability for each DAO class, which covers all user stories of each feature. Each test case is thoroughly documented, including the process, expected outcome, and final result.

DAO Class	Test Method	Process	Expected Outcome	Result
OrderDAOTest	testAddOrder	Add an order, retrieve it, and verify the details.	Order should be successfully added and retrievable with matching details.	Order added successfully, details match the input.
	testGetOrderById	Add an order and retrieve it by OrderID.	Order should be retrievable by OrderID, and details should match the input.	Order retrieved successfully with matching details.
	testUpdateOrder	Add an order, update specific fields like OrderAmount, and verify the update.	Order should be updated, and changes should reflect in the retrieved order.	Order updated successfully, and changes were reflected in the database.
	testDeleteOrder	Add and then delete an order, and attempt to retrieve it afterward.	Order should be deleted and no longer retrievable.	Order deleted successfully, no longer retrievable.

	testGetAllOrders	Add multiple orders and retrieve all using getAllOrders.	All added orders should be retrievable.	All orders retrieved successfully.
	testGetAllOrdersForUser	Add multiple orders for different users and retrieve orders for a specific user by UserID.	Orders for the specified user should be retrieved correctly.	Correct number of orders retrieved for the user.
PaymentDAOTest	testAddPayment	Add a payment method for a user and retrieve it by UserID.	Payment should be successfully added and retrievable by UserID.	Payment added successfully and retrieved.
	testGetPaymentById	Add a payment and retrieve it using PaymentMethodID.	Payment should be retrievable by its ID.	Payment retrieved successfully by ID.
	testUpdatePayment	Add a payment, update card details, and verify the update.	Updated card details should be saved and retrievable.	Payment updated successfully, details reflected correctly.
	testDeletePayment	Add a payment, delete it by PaymentMethodID, and try to retrieve it afterward.	Payment should be deleted and no longer retrievable.	Payment deleted successfully, no longer retrievable.
	test GetAllPaymentsByUserId	Add multiple payments for a user and retrieve them by UserID.	All payments for the user should be retrieved correctly.	All payments retrieved successfully for the user.

	testPaymentExists	Add a payment, then check if the payment exists for a user using paymentExists.	The method should return true if the payment exists for the user.	Payment confirmed to exist for the user.
CarDAOTest	testAddCar	Add a car with fields like SellPrice, Brand, Model, and retrieve the car.	Car should be successfully added and retrievable with matching details.	Car added successfully, details matched.
	testUpdateRemoveCar	Add a car, update fields like Brand, SellPrice, and Model, retrieve the updated car, and delete it.	Car should be updated with new details, and after deletion, it should no longer be retrievable.	Car updated successfully and deleted, no longer retrievable.
DiscussionThreadDAOTest	testAddFindDeleteThread	Add a discussion thread, retrieve it by Title, and then delete it.	Thread should be successfully added, retrievable, and deletable.	Thread added, retrieved, and deleted successfully.
	testUpdateThread	Add a thread, update the Title and Message, and verify the update.	Thread should be updated and retrievable with new details.	Thread updated successfully, details reflected correctly.
GeneralInquiryDAOTest	testCreateInquiry	Add an inquiry with fields like InquiryMessage, PreferredCallbackTime,	Inquiry should be added and retrievable by UserID.	Inquiry added successfully and retrieved by UserID.

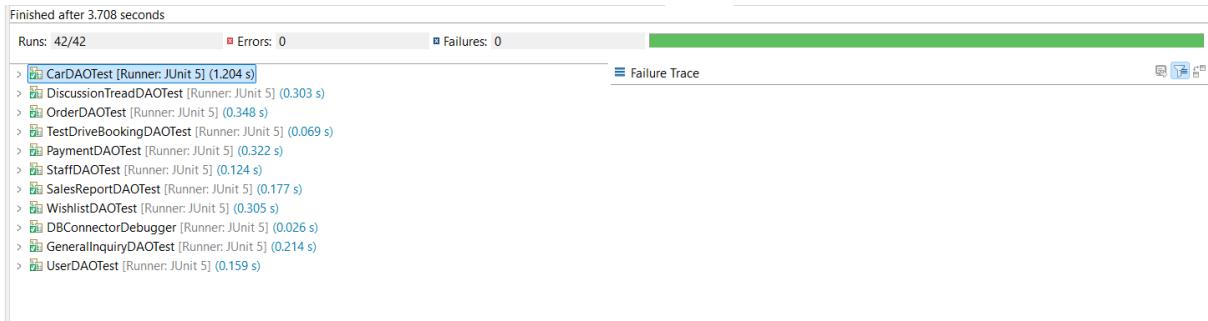
		and UserID, then retrieve it by UserID.		
	testRetrieveInquiryByUserId	Add an inquiry and retrieve it by UserID.	Inquiry should be retrievable with correct details.	Inquiry retrieved successfully.
	testUpdateInquiry	Add an inquiry, update the InquiryMessage, and verify the changes.	Inquiry should be updated with the new message.	Inquiry updated successfully, changes reflected correctly.
	testDeleteInquiry	Add an inquiry, delete it, and attempt to retrieve it afterward.	Inquiry should be deleted and no longer retrievable.	Inquiry deleted successfully, no longer retrievable.
	testSearchInquiries	Add multiple inquiries, search by message or callback time, and verify the results.	Only inquiries matching the search criteria should be retrieved.	Search returned correct inquiries matching criteria.
SalesReportDAOTest	testAddDeleteSalesReport	Add a sales report with fields like Title, TotalSales, Comments, retrieve it, then delete it.	Sales report should be added, retrievable, and deletable.	Sales report added, retrieved, and deleted successfully.
	testUpdateSalesReport	Add a sales report, update fields like Title, TotalSales, and Comments, and verify the update.	Sales report should be updated and retrievable with the new details.	Sales report updated successfully, changes reflected correctly.

	testGetAllSalesReports	Retrieve all sales reports from the system using getAllSalesReports.	All available reports should be retrieved.	All sales reports retrieved successfully.
StaffDAOTest	testAddDeleteStaff	Add a staff member with fields like FirstName, LastName, Email, retrieve the staff member by Email and Password, and delete it.	Staff member should be added, retrievable, and deletable.	Staff member added, retrieved, and deleted successfully.
	testUpdateDeleteStaff	Add a staff member, update fields like PhoneNumber, Password, retrieve the updated details, and then delete the staff member.	Staff member should be updated with new details, retrievable, and deletable.	Staff member updated and deleted successfully.
UserDAOTest	testAddDeleteUser	Add a user with fields like FirstName, LastName, Email, retrieve by Email and Password, and delete it.	User should be added, retrievable, and deletable.	User added, retrieved, and deleted successfully.
	testUpdateDeleteUser	Add a user, update fields like PhoneNumber, Address, retrieve the updated user details, and delete it.	User should be updated with new details, retrievable, and deletable.	User updated and deleted successfully.
WishlistDAOTest	testAddToWishlist	Add a wishlist item for a user, retrieve it, and verify that the details match the added entry.	Wishlist item should be successfully added and retrievable with matching details.	Wishlist item added and retrieved successfully, details matched.

	testGetWishlistByUserId	Add a wishlist item, retrieve the user's wishlist by UserID, and verify the correct number of items.	The correct wishlist items should be retrieved for the user.	Wishlist retrieved successfully.
	testGetWishlistItemById	Add a wishlist item, retrieve it by WishlistID, and verify the details.	Wishlist item should be retrievable by its ID, and the details should match the input.	Wishlist item retrieved successfully by WishlistID.
	testUpdateWishlistItem	Add a wishlist item, update its PriorityLevel, and verify the changes.	Wishlist item should be updated, and the new priority level should be saved.	Wishlist item updated successfully, new priority level saved.
	testDeleteWishlistItem	Add a wishlist item, delete it by WishlistID, and verify that it is no longer retrievable.	Wishlist item should be successfully deleted and no longer retrievable.	Wishlist item deleted successfully, no longer retrievable.
	testSearchWishlistByModelOrPriority	Add multiple wishlist items, search by model or priority, and verify the search results.	Only items matching the search criteria should be retrieved.	Correct wishlist items retrieved based on search criteria.
	testRemoveSoldCarsFromWishlist	Add a wishlist item for a car marked as sold, remove sold cars from the wishlist, and verify that the sold car is removed.	Sold cars should be successfully removed from the wishlist.	Sold car removed from wishlist as expected.

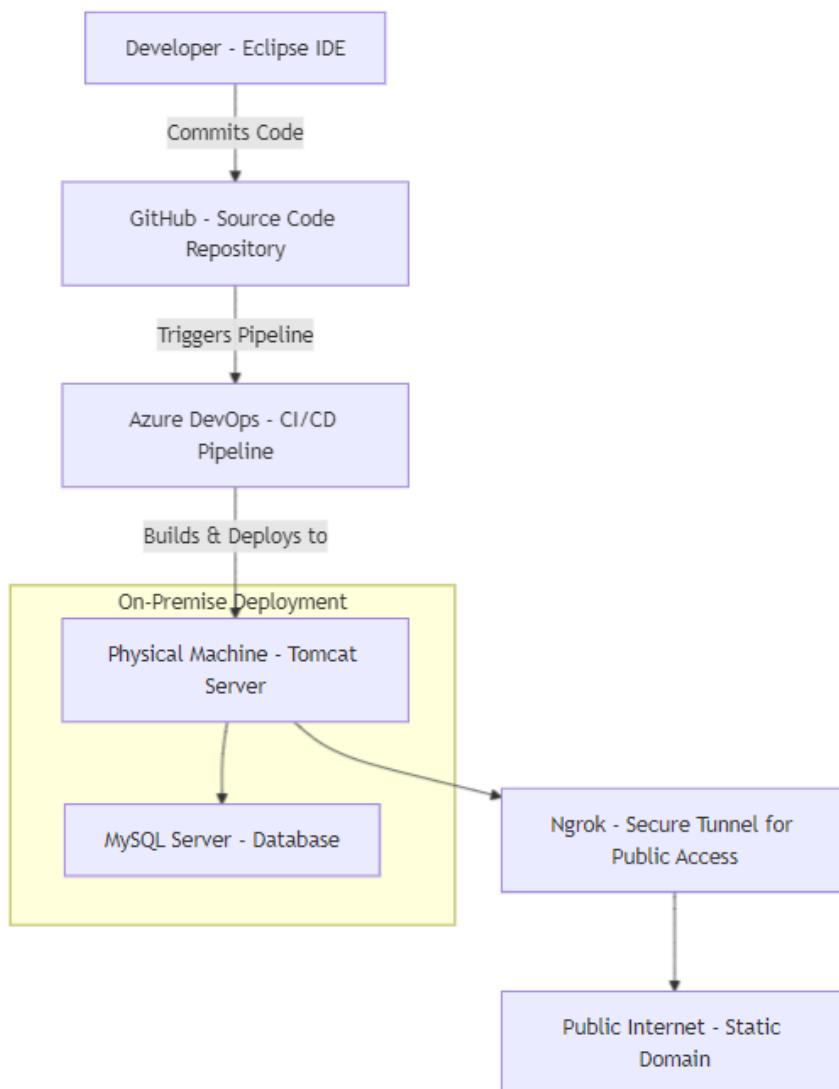
- **JUnit Testing Results**

- As we faced azure issues for testing functionality, the testing was agreed to be presented in eclipse IDE during presentation workshop, and below image showcases the successful completion of our testing phase with no failure trace.



10. Continuous Deployment Design

- **Continuous Deployment (CD) Platform used:**
 - Physical on-premise device
- **Technology used in Continuous Deployment (CD) pipeline includes:**
 - Eclipse IDE
 - Github (<https://github.com/TronKnight/ASD-24>)
 - Azure Devops
 - Tomcat Server
 - MYSQL Server
 - Ngrok (Secure Ingress Platform)
- **CD Workflow**



-
- The Azure Devops pipeline is configured to trigger every time a commit is made on master branch in github and the pipeline involves 2 stages - build and deploy. The build stage using maven results in a “war file” which is deployed to a local machine tomcat server which is exposed to internet via ngrok tunnel and utilizes a static domain for users to access our website.

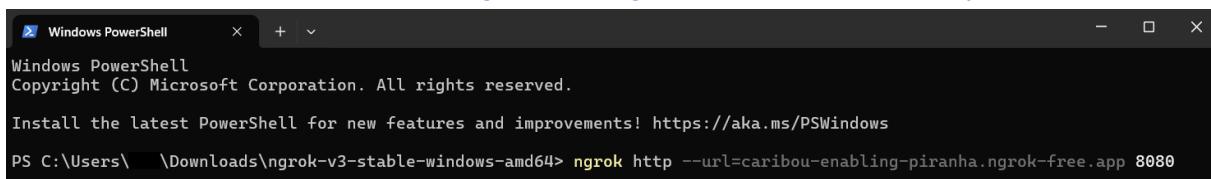
- Screenshot of pipeline configuration for Maven Build Task and Tomcat Deployment Task:

← TronKnight.ASD-24

```
main    ↴ TronKnight/ASD-24 / azure-pipelines.yml

  6  trigger:
  7  --main
  8
  9  pool:
10  --vmImage: ubuntu-latest
11
12  steps:
  Settings
13  - task: Maven@3
14  --inputs:
15  -- mavenPomFile: 'ASD-24/pom.xml'
16  -- mavenOptions: '-Xmx3072m'
17  -- javaHomeOption: 'JDKVersion'
18  -- jdkVersionOption: '1.17'
19  -- jdkArchitectureOption: 'x64'
20  -- publishJUnitResults: true
21  -- testResultsFiles: '**/surefire-reports/TEST-*.xml'
22  -- goals: 'package'
23
  Settings
24  - task: CopyFiles@2
25  --inputs:
26  -- SourceFolder: '$(System.DefaultWorkingDirectory)'
27  -- Contents: '**/target/*.?(war|jar)'
28  -- TargetFolder: '$(Build.ArtifactStagingDirectory)'
29
  Settings
30  - task: PublishBuildArtifacts@1
31  --inputs:
32  -- pathToPublish: '$(Build.ArtifactStagingDirectory)'
33  -- artifactName: drop
34
  Settings
35  - task: TomcatDeployment@1
36  --inputs:
37  -- TomcatUrl: 'https://caribou-enabling-piranha.ngrok-free.app'
38  -- Username: [REDACTED]
39  -- Password: [REDACTED]
40  -- Warfile: '/home/vsts/work/1/s/ASD-24/target/ASD-0.0.1-SNAPSHOT.war'
41  -- Context: '/azure-deploy'
42  -- ServerVersion: '7OrAbove'
```

- Configuring ngrok static domain to point to a on-premise machine's port 8080 opened for the tomcat server where user will access our website using below URL when the server is live. (<https://caribou-enabling-piranha.ngrok-free.app/azure-deploy/>)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ [REDACTED]\Downloads\ngrok-v3-stable-windows-amd64> ngrok http --url=caribou-enabling-piranha.ngrok-free.app 8080
```

11. Project Timesheet

- **Updates compared with release 1:**

- All team members have logged their hours from week 8 to week 12.

11.1. Student 1 - Dev Joisar

Student Name:	Dev Joisar												
Approving Manager Lead:													
Project:	UTS Car Store												
Subject:	ASD												
Day	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Total
Monday	2	1	-	3	-	-	-	-	-	2	-	-	8
Tuesday	-	1	1	-	1	-	-	-	-	2	-	-	5
Wednesday	2	1	-	-	1	-	-	-	-	-	-	-	4
Thursday	-	-	-	-	-	-	-	-	-	-	-	-	-
Friday	-	-	-	-	-	-	-	-	-	-	-	-	-
Saturday	4	3	-	4	3	2	1	-	-	1	-	-	18
Sunday	-	3	2	-	3	2	2	-	-	-	-	-	12
Total	8	9	3	7	8	4	3	-	-	5	-	-	47
Rate	\$80	\$80.00	\$80	\$80	\$80	\$80	\$80	\$80	\$80	\$80	\$80	\$80	\$80
Total	\$640	\$720	\$240	\$560	\$640	\$320	\$240	-	-	\$400	-	-	\$3,760
Team Member Signature:							Date:						
Dev Joisar							1/11/24						
Team Lead/ Manager Signature:							Date:						
Dev Joisar							1/11/24						

11.2. Student 2 - Joshua Browne

Student Name:	Joshua Browne												
Approving Manager Lead:													
Project:	UTS Car Store												
Subject:	ASD												
Day	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Total
Monday	-	-	3	2	-	3	-	-	-	2	-	-	10
Tuesday	4	2	2	-	2	-	-	-	-	2	-	-	12
Wednesday	-	-	-	-	-	-	-	-	-	-	-	-	-
Thursday	2	3	-	-	3	-	-	-	-	-	-	-	8
Friday	-	-	-	2	-	-	-	-	-	-	-	-	2
Saturday	2	3	-	3	-	-	3	-	-	1	-	-	12
Sunday	2	2	-	2	4	-	-	-	-	-	-	-	10
Total	10	10	5	9	9	3	3	-	-	5	-	-	54
Rate	\$80.00	\$80	\$80	\$80	\$80	\$80	\$80.00						
Total	\$800	\$800	\$400	\$720	\$720	\$240	\$240	-	-	\$400	-	-	\$4,320
Team Member Signature:						Date:							
Joshua Browne					1/11/24								
Team Lead/ Manager Signature:					Date:								
Dev Joisar					1/11/24								

11.3. Student 3 - Gurudeep Dhinjan

Student Name:	Gurudeep Dhinjan												
Approving Manager Lead:													
Project:	UTS Car Store												
Subject:	ASD												
Day	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Total
Monday	2	1	-	2	1	-	-	-	-	2	-	-	8
Tuesday	-	2	-	-	2	-	-	-	-	-	-	-	4
Wednesday	-	2	1	-	2	1	1	-	-	-	-	-	7
Thursday	-	-	2	-	-	2	2	-	-	-	-	-	6
Friday	-	2	1	-	2	1	1	-	-	-	-	-	7
Saturday	-	2	2	-	2	2	2	-	-	1	-	-	11
Sunday	1	2	1	1	2	1	1	-	-	-	-	-	9
Total	3	11	7	3	11	7	7	-	-	3	-	-	52
Rate	\$80.00	\$80.00	\$80.00	\$80.00	\$80.00	\$80.00	\$80.00	\$80	\$80	\$80	\$80	\$80	\$80.00
Total	\$240	\$880	\$560	\$240	\$880	\$560	\$320	-	-	\$240	-	-	\$4,160
Team Member Signature:													
Gurudeep Dhinjan													
Team Lead/ Manager Signature:													
Dev Joisar													

11.4. Student 4 - Lavanya Chandna

Student Name:	Lavanya Chandna											
Approving Manager Lead:												
Project:	UTS Car Store											
Subject:	ASD											
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Day	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Monday	3	2	-	3	-	2	-	-	-	2	-	-
Tuesday	-	3	1	-	-	-	-	-	-	-	-	-
Wednesday	-	3	1	-	-	-	-	-	-	-	-	-
Thursday	-	-	-	-	-	-	-	-	-	-	-	-
Friday	-	-	-	-	-	-	1	-	-	-	-	-
Saturday	2	1	1	2	2	3	2	-	-	-	-	-
Sunday	-	2	1	2	3	-	2	-	-	-	-	-
Total	5	11	4	7	5	5	5	-	-	2	-	-
Rate	\$80.00	\$80.00	\$80.00	\$80.00	\$80.00	\$80.00	\$80.00	\$80	\$80	\$80	\$80	\$80.00
Total	\$400	\$880	\$320	\$560	\$400	\$400	\$400	-	-	\$160	-	-
Team Member Signature:					Date:							
Lavanya Chandna					1/11/24							
Team Lead/ Manager Signature:					Date:							
Dev Joisar					1/11/24							

11.5. Student 5 - Keegen Wallis

Student Name:	Keegen Wallis												
Approving Manager Lead:													
Project:	UTS Car Store												
Subject:	ASD												
Day	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Total
Monday	-	3	2	-	-	-	2	-	-	2	-	-	9
Tuesday	-	2	-	-	-	-	-	-	-	2	-	-	4
Wednesday	-	2	-	-	-	-	-	-	-	-	-	-	2
Thursday	1	-	-	-	-	-	-	-	-	-	-	-	1
Friday	-	-	-	3	-	-	-	-	-	-	-	-	3
Saturday	2	-	2	2	3	2	-	-	-	1	-	-	12
Sunday	-	4	2	2	2	2	-	-	-	-	-	-	12
Total	3	11	6	7	5	4	2	-	-	5	-	-	43
Rate	\$80.00	\$80	\$80	\$80	\$80	\$80	\$80.00						
Total	\$240	\$880	\$480	\$560	\$400	\$320	\$160	-	-	\$400	-	-	\$3,440
Team Member Signature:						Date:							
Keegen Wallis					1/11/24								
Team Lead/ Manager Signature:					Date:								
Dev Joisar					1/11/24								

12. Individual Contribution Logbook

12.1. Student 1 - Dev Joisar

Sprint Week	Completed Task
1 (19th August)	Foundation Setup of Azure Account, document template, Github, Discord. Setup project CSS, landing pages (Login, register, and home), database schema for accounts. Build pipeline tested and working.
2 (26th August)	Completed User Register, Login, and logout
3 (2nd September)	Completed Staff Register, Login, and logout Report completed.
4 (9th September)	Completed Staff Account Management
5 (16th September)	Completed User Account Management & Logs
6 (23rd September)	Completed JUnit Testing
7 (30nd September)	Updated report
8 (7th October)	-
9 (14th October)	-
10 (21nd October)	Finalised Deploy pipeline with the team and updated the report for final release.
11 (28th October)	-

12.2. Student 2 - Joshua Browne

Sprint Week	Completed Task
1 (19th August)	Car Catalogue outline drafted
2 (26th August)	Car Database Set up CarDao and Servlets
3 (2nd September)	Car Pages created and linked to main page
4 (9th September)	Completed CarDAO Junit Testing
5 (16th September)	TestDriveBooking Outline drafted TestDriveBooking Database Set up
6 (23nd September)	Test Drive Booking Servlets created and linked Completed Junit Testing for TestDriveBookingDAO
7 (30nd September)	Updated Report
8 (7th October)	-
9 (14th October)	-
10 (21nd October)	Finalised Deploy pipeline with the team and updated the report for final release.
11 (28th October)	-

12.3. Student 3 - Gurudeep Dhinjan

Sprint Week	Completed Task
1 (19th August)	Foundation of order - SQL, Model, DAO, Servlet, JSP
2 (26th August)	Complete order - SQL, Model, DAO, Servlet, JSP All CRUD function were working for order
3 (2nd September)	Fixed the entire project CSS and worked on the report
4 (9th September)	Completed Junit Testing for Order
5 (16th September)	Worked on refactoring, order, payment to make it work in the sql database
6 (23rd September)	Completed CRUD for Paymnt, JUnit for Payment, fix Order Junit and implemented order history
7 (30nd September)	Completed the report for assignment 2
8 (7th October)	-
9 (14th October)	-
10 (21nd October)	Finalised Deploy pipeline with the team and updated the report for final release.
11 (28th October)	-

12.4. Student 4 - Lavanya Chandna

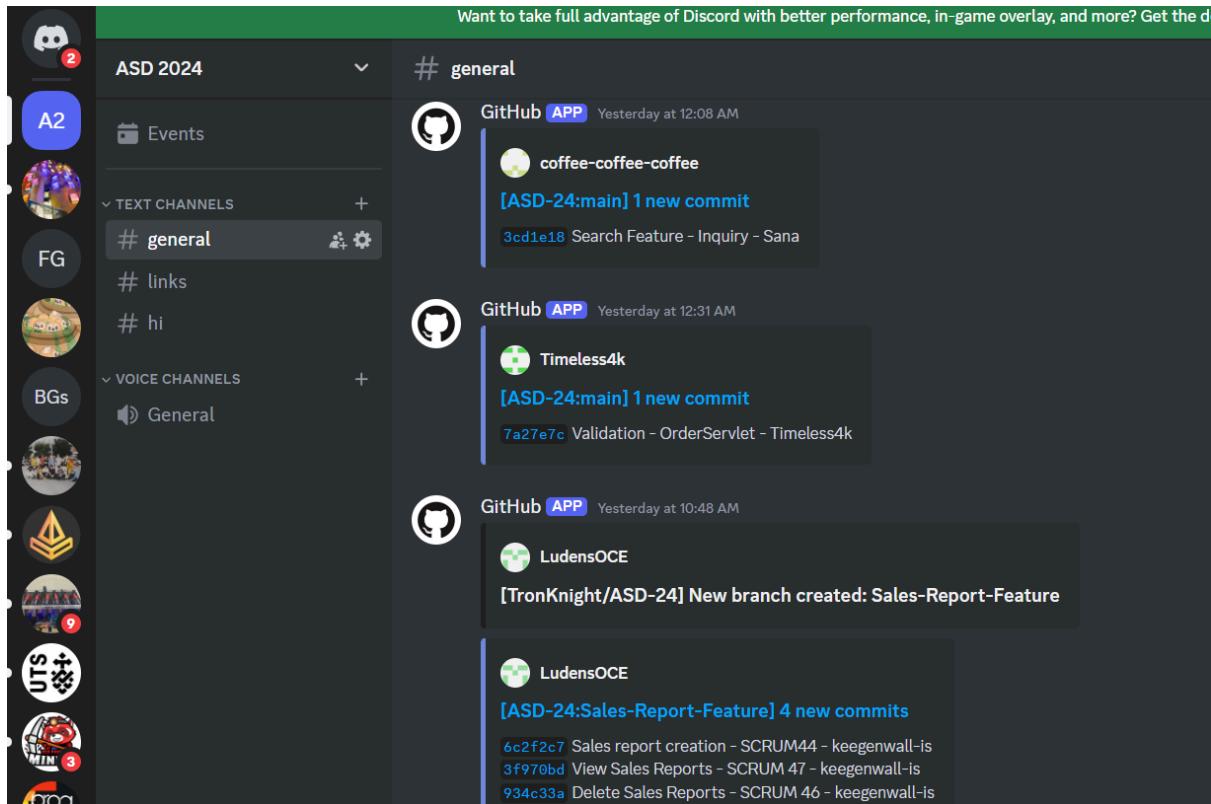
Sprint Week	Completed Task
1 (19th August)	General Inquiry - SQL General Inquiry - Model General Inquiry - DAO (Foundation) General Inquiry - Servlet (Foundation)
2 (26th August)	General Inquiry - JSP (Foundation) Completed - DAO & Servlet
3 (2nd September)	Completed - JSP & Report Implemented - Search function to inquiries
4 (9th September)	Wishlist SQL & Model were set up
5 (16th September)	Junit testing for General Inquiry completed and the DAO and Servlet Foundation were done
6 (23nd September)	Wishlist CRUD was completed and the JSP with the car catalogue updated to work with wishlist
7 (30nd September)	Worked on assignment 2 report and Junit testing for wishlist
8 (7th October)	-
9 (14th October)	-
10 (21nd October)	Finalised Deploy pipeline with the team and updated the report for final release.
11 (28th October)	-

12.5. Student 5 - Keegen Wallis

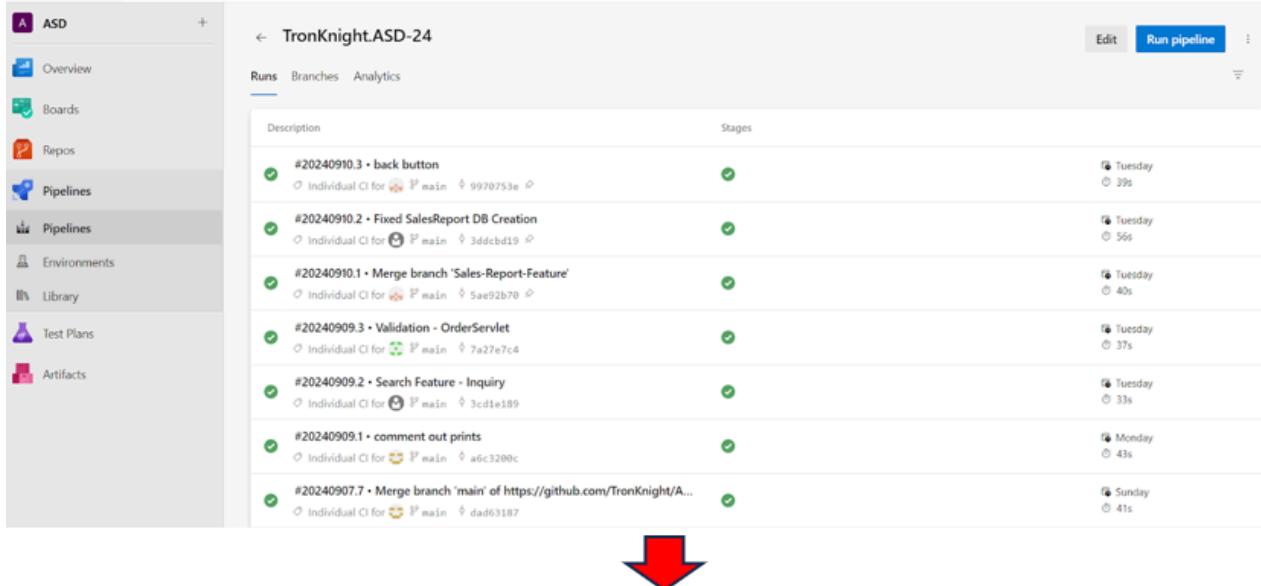
Sprint Week	Completed Task
1 (19th August)	Finished JSP for Sales Reports View Created SQL database for sales reports Started JSP for Create and Update report
2 (26th August)	Finished JSP for Create and Update report Implemented DAO, Servlet and Model
3 (2nd September)	Completed and implemented JSP Completed DAO and Servlet functionality Completed report elements
4 (9th September)	Created JSP for Discussion Board Implemented SQL database for discussion threads
5 (16th September)	Implemented DAO, Servlet and Model Implemented View and Delete functionality
6 (23rd September)	Created JSP for Create and Update Threads Implemented Create and update functionality
7 (30nd September)	Implemented JUnit testing for sales report and discussion thread features Completed report elements
8 (7th October)	-
9 (14th October)	-
10 (21nd October)	Finalised Deploy pipeline with the team and updated the report for final release.
11 (28th October)	-

13. Appendices

13.1. Github - Discord Integration



13.2. Azure Build Pipeline Artifact



The screenshot shows the Azure DevOps Pipelines interface for the project 'ASD'. The left sidebar includes options like Overview, Boards, Repos, Pipelines, Pipelines (selected), Environments, Library, Test Plans, and Artifacts. The main area displays a list of recent pipeline runs under the heading 'TronKnight.ASD-24'. The runs are listed by ID, showing the commit hash, branch, and duration. A red arrow points down to the details of the first run, '#20240910.3 • back button'.

#20240910.3 • back button

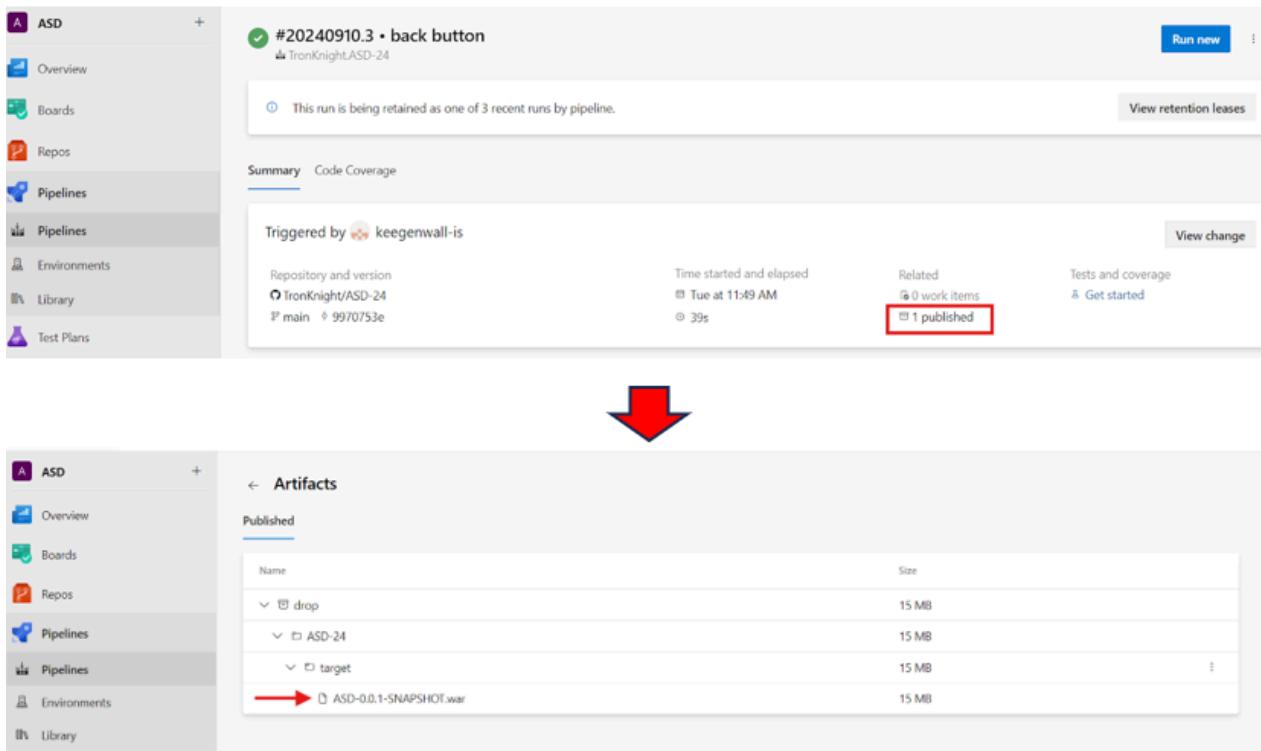
Triggered by keegenwall-is

Repository and version: IronKnight/ASD-24, 1st main 9970753e

Time started and elapsed: Tue at 11:49 AM, 39s

Related: 0 work items, 1 published (boxed)

Tests and coverage: Get started



The screenshot shows the Azure DevOps Artifacts interface for the project 'ASD'. The left sidebar includes options like Overview, Boards, Repos, Pipelines, Pipelines (selected), Environments, Library, Test Plans, and Artifacts. The main area displays a list of published artifacts under the heading 'Artifacts'. A red arrow points down to the 'ASD-0.0.1-SNAPSHOT.war' file.

Name	Size
drop	15 MB
ASD-24	15 MB
target	15 MB
ASD-0.0.1-SNAPSHOT.war	15 MB