

# Proposal: Fleet Transit System

Presented by Night Class

# Contents

Executive Summary.....	13
Goals .....	13
Stakeholders .....	13
Timeline .....	13
Stakeholders: User Stories.....	14
System Access – Administrator .....	15
System Access – Employee .....	16
System Access – Client.....	16
System Access – User .....	17
Employee – Driver .....	19
Employee – Dispatch .....	19
Employee – Fleet-Manager .....	21
Employee – Maintenance .....	22
Employee – Mechanic .....	23
Employee – Parts Person .....	24
Employee – Operations Manager .....	25
Employee – Customer Support.....	26
Riders – Charter with Driver .....	28
Riders – Charter without Driver .....	28
Riders – Ride Requester .....	29
Riders – Guardian.....	29
Outside Contacts .....	31
Stakeholders: Use Cases .....	33
System Access – Administrator .....	34
System Access – User .....	52
System Access – Client.....	84
Employee – Driver .....	85
Employee – Dispatch .....	94
Employee – Fleet-Manager .....	130
Employee – Maintenance .....	143
Employee – Mechanic .....	168
Employee – Parts Person .....	180
Employee – Operations Manager .....	203
Employee – Fleet & Operations .....	228
Employee – Customer Support.....	238

Riders – Charter with Driver .....	246
Riders – Charter without Driver .....	255
Rider –Route Rider .....	262
Riders – Ride Scheduler.....	265
Riders – Ride Requester .....	270
Riders – Passenger.....	275
Riders – Person with Disability .....	278
Riders – Guardian.....	286
Outside Contacts .....	292
Stakeholders: User Interfaces.....	315
UI Drawings for Admin .....	315
Create Profile Desktop View(1) .....	315
Create Profile Desktop View.....	316
System Message - Create - Desktop View.....	317
System Message - Edit - Desktop View .....	318
System Message - Edit_Deactivate - Destkop View.....	319
System Message - List - Desktop View.....	320
System Message - View - Desktop View .....	321
View Employee Profile Desktop View .....	322
View Employee Profile With Edit Destkop View .....	323
View Employee Profile With Edit_Deactivate Desktop View.....	324
View Employee Profile With Reactivate Destkop View .....	325
View Employee Profile With Roles Desktop View.....	326
View List of Users Desktop View(1).....	327
View List of Users Desktop View .....	328
View Profile Desktop View .....	329
View Profile With Edit Button Destkop View .....	330
View Profile With Edit_Deactivate Desktop View .....	331
View Profile With Reactivate Desktop View.....	332
UI Drawings for Charter .....	333
Cancel Reservation Mobile.....	333
Cancel Reservation Web .....	334
Create Reservation mobile .....	335
Create Reservation Web .....	336
Update Reservation Mobile .....	337

Update Reservation Web.....	338
View Reservation Mobile .....	339
View Reservation Web.....	340
UI Drawings for Customer Support Representitive .....	341
Create Support Ticket - Desktop.....	341
Create Support Ticket - web .....	342
Update Support Ticket - Desktop .....	343
Update Support Ticket - Web .....	344
View All Support Tickets - Desktop.....	345
View All Support Tickets - Web .....	346
View Support Ticket Details - Desktop.....	347
View Support Ticket Details - Web .....	348
UI Drawings for Dispatch .....	349
Assign Selected Vehicle - Destkop View .....	349
Assignment View With Deactivate - Desktop View .....	350
Schedule A Driver - Desktop View .....	351
Schedule A Vehidle - Desktop View.....	352
UI Drawings for Driver .....	353
Driver Assigned Vehicle - Mobile.....	353
Driver Assigned Vehicle - Web .....	354
Driver Begin Route - Mobile .....	355
Driver Begin Route - Web .....	356
Driver Dropoff Passenger - Web.....	357
Driver Dropoff Passenger -Mobile.....	358
Driver End Route - Mobile.....	359
Driver End Route - Web .....	360
Driver Passenger List - Mobile .....	361
Driver Passenger List - Web .....	362
Driver Pickup Passenger - Mobile .....	363
Driver Pickup Passenger -Web.....	364
Driver Refuel Log - Mobile .....	365
Driver Refuel Log - Web.....	366
Driver Selecrt Rotues - Mobile.....	367
Driver Select Route - Web .....	368

Driver Select Routes Popups - Web .....	369
Driver Submit Maintenance Report - Mobile .....	370
Driver Submit Maintenance Report - Web .....	371
Driver Submit Safety Report - Mobile .....	372
Driver Submit Safety Report - Web .....	373
Driver Vehicle Checkout - Mobile .....	374
Driver Vehicle Checkout - Web .....	375
Driver Vehicle List - Mobile .....	376
Driver Vehicle List - Web .....	377
Driver Vehicle Return - Mobile .....	378
Driver Vehicle Return - Web .....	379
UI Drawings for Fleet Admin .....	380
Add Rental Vehicle Desktop View .....	380
Add Vehicle Destkop View .....	381
Change Vehicle Info Desktop View .....	382
Deactivate Vehicle Destkop View .....	383
Fuel Receipt List Destkop View .....	384
Lookup Vehicle Desktop Vlew .....	385
Lookup Vehicle Info Desktop View .....	386
Lookup Vehicle Inspection Report Desktop View .....	387
UI Drawings for Fleet Operations Manager .....	388
Add Employee Alt .....	388
Add Employee .....	389
Current Route Tracking .....	390
Employee Detail Alt .....	391
Employee Detail .....	392
Employee Lookup .....	393
Update Deactive Employee Alt .....	394
Update Deactive Employee .....	395
View All Route detail .....	396
View Route Detail .....	397
UI Drawings for Maintenance - Mechanic .....	398
Maintenanace - Create Compatable Part - Desktop .....	398
Maintenance - Complete Vehicle Workorder - Desktop .....	399

Maintenance - Create Vehicle Workorder - Desktop .....	400
Maintenance - Delete Compatable Parts - Desktop .....	401
Maintenance - Edit Vehicle Details - Desktop.....	402
Maintenance - Edit Vehicle Workorder - Desktop .....	403
Maintenance - Pending Services - Desktop .....	404
Maintenance - Pending Vehicle Workorders - Desktop.....	405
Maintenance - Update Compatable Parts - Deskstop .....	406
Maintenance - Vehicle Details - Desktop .....	407
Maintenance - Vehicle Workorder Log - desktop .....	408
Maintenenace - View Compatable Parts - Desktop.....	409
UI_MMTC_04_02 .....	410
UI_MMTC_04_03 .....	411
UI_MMTC_06_01 .....	412
UI_MMTC_07_01 .....	413
UI Drawings for Operatons Manager .....	414
Operations Manager - Create Vehicle - Desktop.....	414
Operations Manager - View Schedule Detail - Desktop .....	415
Operations Manager - View Vehicle - Desktop .....	416
Operations Manager Activate Bus Route - Desktop.....	417
Operations Manager Activate Stop - Desktop .....	418
Operations Manager Add or Edit Driver .....	419
Operations Manager Create New Bus Route - Desktop.....	420
Operations Manager Create Schedule - Desktop.....	421
Operations Manager Create Stop - Desktop .....	422
Operations Manager Deactivate Bus Route - Desktop.....	423
Operations Manager Deactivate Stop - Desktop .....	424
Operations Manager List Bus Routes - Desktop .....	425
Operations Manager Stop List - Desktop .....	426
Operations Manager Update Bus Route - Desktop .....	427
Operations Manager Update Schedule - Desktop .....	428
Operations Manager Update Stop - Desktop.....	429
Operations Manager View Bus Route Details - Desktop.....	430
Operations Manager View or Deactivate Driver.....	431
Operations Manger View Schedule List - Desktop .....	432

UI Drawings for Outside Contact .....	433
Add Service Bid.....	433
Add Special Inspection .....	434
Add Special Service Order .....	435
Add Speical Work Order.....	436
Update or Deactivate Special Service Order .....	437
Update Or Delete Service Bid .....	438
Update or Delete Special Inspection .....	439
Update or Delete Speical Work Order.....	440
View All Service Bid.....	441
View All Special Inspection .....	442
View All Special Service Order .....	443
View All Speical Work Order.....	444
View Service Bid.....	445
View Special Inspection .....	446
View Special Service Order .....	447
View Speical Work Order.....	448
UI Drawings for Parent .....	449
Add Bus Instrucitons - Web .....	449
Add Dependent .....	450
Edit Bus Route Assignment - Web.....	451
Edit Dependent.....	452
Set Bus Route Availability - Web.....	453
View Bus History Report - Web.....	454
View Current Status - Web.....	455
UI Drawings for Parts Person .....	456
Parts Add Part to Inventory - DESKTOP .....	456
Parts Confirm Parts Request - DESKTOP .....	457
Parts Create Change Order - DESKTOP.....	458
Parts Part Inventory Audit - DESKTOP.....	459
Parts Purchase Order Details - DESKTOP.....	460
Parts Reject Part Request - DESKTOP .....	461
Parts View Current PO Details - DESKTOP .....	462
Parts View Park Requests - DESKTOP.....	463

Parts View Part Request Details - DESKTOP .....	464
Parts View Purchase Orders - DESKTOP.....	465
Parts View Special PO Details - DESKTOP.....	466
UI_Part.03.06.Desktop .....	467
UI_Part.03.07.Desktop .....	468
UI_Part.04.01.Desktop .....	469
UI Drawings for Ride Request Later .....	470
Ride Request Later Request Ride MOBILE .....	470
UI Drawings for Ride Request Now.....	471
Details Of Active Services WEB.....	471
Immediate Ride Request WEB .....	472
Next Available Ride Request WEB .....	473
View Active Transit Services WEB .....	474
View Services By Start And End Location WEB .....	475
UI Drawings for Ride Schedule Later .....	476
Time Table WEB.....	476
Location Entry And Active Services WEB.....	477
Schedule Ride WEB .....	478
View Active Transit Services WEB .....	479
View Details Of Active Service WEB .....	480
UI Drawings for Student .....	481
Edit Account Preferences - mobile .....	481
Scheudle Ride - mobile.....	482
View Available Rides - mobile.....	483
UI Drawings for User .....	484
Change Communications Settings Desktop .....	484
Change Communications Settings Mobile .....	485
Change Communications Settings Web.....	486
Change UX Settings Desktop.....	487
Change UX Settings Mobile.....	488
Change UX Settings Web .....	489
Create Account Desktop .....	490
Create Account Mobile .....	491
Create Account Web.....	492
Deactivate Account Mobile .....	493

Deactivate Account Web .....	494
Log In Desktop .....	495
Log In Mobile.....	496
Log In Web .....	497
Log out Desktop .....	498
Log Out Mobile .....	499
Log Out Web .....	500
Modify Payment Info Desktop .....	501
Modify Payment Info Mobile .....	502
Modify Payment Info Web.....	503
Modify Personal Info Desktop.....	504
Modify Personal Info Mobile .....	505
Modify Personal Info Web .....	506
Navigate Application Desktop .....	507
Navigate Application Mobile.....	508
Navigate Application Web .....	509
Reactivate Account Mobile .....	510
Reactivate Account Web .....	511
Receive Notifications Desktop .....	512
Receive Notifications Mobile .....	513
Receive Notifications Web.....	514
Request Help Desktop .....	515
Request Help Mobile.....	516
Request Help Web .....	517
Reset Password Desktop .....	518
Reset Password Mobile .....	519
Reset Password Web.....	520
Retrieve Username Desktop .....	521
Retrieve Username Mobile.....	522
Retrieve Username Web .....	523
UI-User.01.05.DESK.NOTUSED .....	524
UI-User.01.06.DESK.NOTUSED .....	525
UI-User.01.07.DESK.NOTUSED .....	526
UI-User.22.01.MOB.....	527

UI-User.22.02.MOB.....	528
User Main Menu Desktop .....	529
User Main Menu Mobile .....	530
User Main Menu Web.....	531
View Messages Desktop.....	532
View Messages Mobile .....	533
View Messages Web .....	534
View Purchases Desktop.....	535
View Purchases Mobile .....	536
View Purchases Web .....	537
Use Case Context and Subsystem Diagrams .....	539
Charter .....	540
Customer Service.....	541
Driver .....	542
Maintenance .....	543
Notification .....	546
Parts Inventory .....	547
Ride Request and Schedule .....	549
Routing .....	550
Schedule and Assignment .....	551
Database Design.....	552
Entity Relationship Diagrams .....	552
Data Dictionary.....	564
Stored Procedures for Data Operations .....	590
Client Accommodation Stored Procedures .....	592
Bid Related Stored Procedures.....	594
Charter_Accommodation Stored Procedures.....	599
Charter_Assignment Stored Procedures .....	600
Charter_Rider Stored Procedures.....	603
Charter_Stop Stored Procedures.....	605
Charter Stored Procedures.....	607
Client_Client_Role Stored Procedures .....	610
Client Stored Procedures .....	612
Dependent Stored Procedure Definitions .....	613
Dependent Accommodation Stored Procedures .....	614

Driver Maintenance Report Stored Procedures .....	615
DRIVER LICENSE CLASS Stored Procedures.....	618
Driver Unavailable Stored Procedures .....	619
Employee Role Stored Procedures Definitions .....	623
Employee Stored Procedures .....	624
Inspection Report Stored Procedures .....	626
Login Stored Procedures .....	629
Maintenance Schedule Stored Procedures.....	630
Model Compatibility Stored Procedures .....	632
Model_Lookup Stored Procedures.....	633
Packing Slip Line Items Stored Procedure Definitions .....	635
Ride Stored Procedures .....	637
Role Stored Procedures Definitions.....	639
route_stop Stored Procedures.....	639
Service Vehicle Stored Procedures .....	647
Special Inspection-Related Stored Procedures.....	651
Special_Service_detail Related Stored Procedures .....	653
Special Work Order Related Stored Procedures .....	655
Stop Stored Procedures.....	657
UML – Data Domain Models .....	660
Class Interface Diagrams .....	661
UML – Data Access Class Interface Diagrams .....	689
UML - Data Manager Class Interface Diagrams.....	690
Data Flow Diagrams by Subsystem .....	691
Charter Subsystem .....	692
Customer Service Subsystem .....	698
Driver Subsystem .....	699
Maintenance Subsystem .....	707
Notification Subsystem .....	711
Outside Contact Subsystem .....	712
Parts Subsystem .....	713
Route Subsystem .....	716
Activity Diagrams by Subsystem .....	724
Charter Subsystem .....	725
Customer Service Subsystem .....	726

Driver Subsystem .....	727
Maintenance Subsystem .....	735
Parts Subsystem .....	736
Ride Subsystem.....	741
Route Subsystem .....	743
State-Chart Diagrams by Subsystem.....	744
Schedule And Driver Subsystem .....	745
Scheduling.....	745
Charter Customer Trip .....	746
Driver .....	747
Maintenance and Mechanic Subsystem.....	748
Compatible Parts.....	748
Scheduled Maintenance .....	749
Work Orders.....	750
Inventory .....	751
Special Work Orders .....	752
Project Details .....	753
Timeline .....	753
Technology and Tools.....	753
Testing and QA .....	753
Project Management.....	753

# Executive Summary

Contained in this document is a proposal for {Name of Project}, software catered to communities that have a need for a fleet management system. Users can use this application to charter rides, request from transit services and ride on routes. This application can also be used to manage employees, routes, ride services, charter requests, maintenance, and parts inventory. This application will have basic customer support features to allow for efficient handling of client requests.

## Goals

To provide a more rounded product we have outlined goals that will be achieved by project completion:

- Increase operation efficiency.
- Improve vehicle lifespan by implementing proactive maintenance practices.
- Reduce environmental impact of the fleet by optimizing routes.

## Stakeholders

- Local communities
- Local and state governments

## Timeline

The project is estimated to be completed within a 16 week timeframe. Upon project approval development will begin January 2024, and complete May 2024. Due to the educational nature of the project an estimated budget is not applicable.

## Stakeholders: User Stories

The User Stories describe the needs of users and how the application will serve meet those needs. The User Stories are organized by category of users, referred to hereon as the “actor”, and are identified by each need that an actor has for the application to fulfill. The User Stories, when compiled by actor, layout the interface between the application and the actor.

## System Access – Administrator

As a(n) <actor>	I would like to <description>	So that <outcome>
Administrator	Add a new user to the database	People working within the system can have their own login information
Administrator	View a list of users, name, email, phone number, roles, etc	I can see a list of users
Administrator	View a users, name, email, phone number, roles, etc	I can look at what each user has on their profile
Administrator	Update a users, name, email, phone number, roles etc	I can change what information is on their profile
Administrator	Deactivate a users login information	Deactivate a users login
Administrator	Reactivate a users login information	Reactivate a users login
Administrator	Create Employees	I can create employee accounts
Administrator	View Employee list	View a list of employees
Administrator	View Employees	I can see what each employee has for information
Administrator	Edit Employee	I can edit what is within an employee
Administrator	Update Employee Roles	I can give permissions to employees based on what roles they have
Administrator	Deactivate employee accounts	I can deactivate employee accounts when they're no longer active with the company
Administrator	Reactivate employee accounts	I can reactivate an employee account in case they return to the company
Administrator	Be able to put out a system wide prompt, specifying endTime, and a message	Users can know if there is something urgent that has changed/come up
Administrator	View a list of past system messages	see all the messages in the past
Administrator	View past system messages	view the details of a particular system message

Administrator	change the current system message	edit the selected system message
Administrator	deactivate the main system message	deactivate the main system message and move it to the past system messages
Administrator	Create reports on the demographics that utilize the services we provide	Know what areas/people are more in need of services
Administrator	View how any other role in the system sees their screen	To allow me to help with troubleshooting if someone has a question and I need to see what their role does quickly

## System Access – Employee

### System Access – Client

As a(n) <actor>	I would like to <description>	So that <outcome>
Client	Create an profile or account with my unique personal information and password	I can access user-specific features
Client	Sign into the application	I can access user-specific features
Client	Sign out of the application	I can prevent unauthorized access to my account
Client	Use security questions or secondary contact information to access my account	I can access my account if I have forgotten my primary login information (phone, username, etc.)
Client	Use my other account information to reset my password	I can access my account if it has become compromised or I have forgotten my password
Client	Deactivate my account	I can stop using the application
Client	Reactivate my account	I can resume using the application after previously deactivating it
Client	Create a personal information sheet for my account	I can have personal details on my profile

Client	View Personal Information associated with my account	I can have personal details on my profile
Client	Update personal information associated with my account	I can keep important details like my address and phone number up to date
Client	View settings that suit my needs and taste	I can interact with the app in an accessible and comfortable way
Client	Edit settings that suit my needs and taste	I can interact with the app in an accessible and comfortable way
Client	View and select from a list of the features I am likely to use	I can see available ways to use the app and select my desired interaction
Client	Switch from using one feature of the application to another	I can perform a variety of tasks during a single use session.
Client	Receive a notification of any event that affects my use of the service	I know when there is a service disruption, change in operation, etc.
Client	View a history of messages, notifications, notes, and other communication	I can review past information relevant to my interaction with the fleet
Client	Request help with problems and provide feedback	I can solve service, account, and technical problems that arise

## System Access – User

As a(n) <actor>	I would like to <description>	So that <outcome>
User	Sign into the application	I can access user-specific features
User	Sign out of the application	I can prevent unauthorized access to my account
User	Use security questions or secondary contact information to access my account	I can access my account if I have forgotten my primary login information (phone, username, etc.)
User	Use my other account information to reset my password	I can access my account if it has become compromised or I have forgotten my password
User	Deactivate my account	I can stop using the application

User	Reactivate my account	I can resume using the application after previously deactivating it
User	Create a personal information sheet for my account	I can have personal details on my profile
User	View Personal Information associated with my account	I can have personal details on my profile
User	Update personal information associated with my account	I can keep important details like my address and phone number up to date
User	View settings that suit my needs and taste	I can interact with the app in an accessible and comfortable way
User	Edit settings that suit my needs and taste	I can interact with the app in an accessible and comfortable way
User	View and select from a list of the features I am likely to use	I can see available ways to use the app and select my desired interaction
User	Switch from using one feature of the application to another	I can perform a variety of tasks during a single use session.
User	Receive a notification of any event that affects my use of the service	I know when there is a service disruption, change in operation, etc.
User	View a history of messages, notifications, notes, and other communication	I can review past information relevant to my interaction with the fleet
User	Request help with problems and provide feedback	I can solve service, account, and technical problems that arise

## Employee – Driver

As a(n) <actor>	I would like to <description>	So that <outcome>
Driver	See my assigned vehicle	I know which vehicle to check out
Driver	Complete checklist to check out/in vehicle	Find any issues with a vehicle and make sure it is safe to operate
Driver	See my route	I can drive to the stops I am needed to
Driver	Start my route	I can inform dispatch my route has started
Driver	End my route	I can show in the system that I ended my route
Driver	Access vehicle return system	I can inform dispatch that I've returned my vehicle and to check it for any issues so that I can put in a maintenance request
Driver	View passenger(s) per stop and any assistive devices they use/require	I don't miss picking up a passenger and I can be prepared to assist them if needed
Driver	Confirm that passenger(s) picked up	I can document that I have picked up passenger(s)
Driver	Confirm that passenger(s) dropped off	I can confirm that the passenger has reached their destination and is no longer in the vehicle
Driver	Complete a safety report document	I can notify management of a safety concern with a passenger or working condition
Driver	Complete a refuel Log	I can document information when I refuel my vehicle.
Driver	Complete a maintenance report	so that I can inform maintenance and management of an issue with my vehicle

## Employee – Dispatch

As a(n) <actor>	I would like to <description>	So that <outcome>
Dispatcher	View Vehicle Schedules	I can assign vehicles to different services, routes or charters based on availability and needs

Dispatcher	Assign Vehicle to Route Assignment	The vehicle can be set aside for that service during those times
Dispatcher	Assign Vehicle to Charter Request	The vehicle can be set aside for that service during those times
Dispatcher	Assign Vehicle to Ride Service	The vehicle can be set aside for that service during those times
Dispatcher	Assign Vehicle to Maintenance Instance	The vehicle can be set aside for that service during those times
Dispatcher	Assign Vehicle to Other Unavailability	The vehicle can be set aside for that service during those times
Dispatcher	Update Vehicle in Route Assignment	The vehicle in use can be changed under necessary circumstances
Dispatcher	Update Vehicle in Charter Request	The vehicle in use can be changed under necessary circumstances
Dispatcher	Update Vehicle in Ride Service	The vehicle in use can be changed under necessary circumstances
Dispatcher	Deactivate Maintenance Instance	The maintenance instance can be cancelled
Dispatcher	Deactivate Other Unavailability	The vehicle can be made available for those times again.
Dispatcher	Reactivate Maintenance Instance	The maintenance instance can be un-cancelled
Dispatcher	Reactivate Other Unavailability	The vehicle unavailability can be un-cancelled
Dispatcher	Reschedule Maintenance Instance	The time the vehicle is used can be changed
Dispatcher	Reschedule Other Unavailability	The time the vehicle is used can be changed
Dispatcher	View Driver Schedules	I can assign drivers to different services, routes or charters based on availability and needs
Dispatcher	Assign Driver to Route Assignment	The driver can be set aside for that service during those times
Dispatcher	Assign Driver to Charter Request	The driver can be set aside for that service during those times
Dispatcher	Assign Driver to Ride Service	The driver can be set aside for that service during those times
Dispatcher	Assign Schedule Unavailability to Driver	The driver can be marked unavailable during those times (vacation, sick days, etc)

Dispatcher	Update Driver In Route Assignment	The driver can be set aside for that service during those times
Dispatcher	Update Driver in Charter Request	The driver can be set aside for that service during those times
Dispatcher	Update Driver in Ride Service	The driver can be set aside for that service during those times
Dispatcher	Deactivate Schedule Unavailability	The driver can be brought back available if the situation changes
Dispatcher	Reactivate Schedule Unavailability	The driver can be brought back unavailable if the situation changes
Dispatcher	Reschedule Driver Unavailability	The driver's availability can reflect if their situation changes
Dispatcher	Reschedule Route Assignment	The time the vehicle and driver are used can be changed
Dispatcher	Deactivate Vehicle+Driver Assignment	The vehicle and driver are both made available for the time of that instance
Dispatcher	Reactivate Vehicle+Driver Assignment	The vehicle and driver are both made unavailable for the time of that instance

## Employee – Fleet-Manager

As a(n) <actor>	I would like to <description>	So that <outcome>
Fleet Manager	Add a vehicle to inventory	Enter new trucks that have been purchased
Fleet Manager	Deactivate a vehicle	Remove vehicles that have been Wrecked/Sold
Fleet Manager	View a list of vehicles	Can see what vehicles are in inventory and their general status
Fleet Manager	View a specific vehicle	Can see detailed information about a specific vehicle
Fleet Manager	Change vehicle info	Update information about a vehicle that is not correct
Fleet Manager	Track a vehicle on map	Locate a vehicle that is not at the facility
Fleet Manager	Lookup parts inventory	Lookup the parts inventory to see what needs to be ordered.

Fleet Manager	Lookup a parts informaiton	Lookup qty, description, specifications, last order date, last used date, order qty, vendor
Fleet Manager	Create a parts PO	Make a PO for needed parts to send to the vendor
Fleet Manager	Lookup part PO	See PO's for parts that have been ordered
Fleet Manager	Remove a part from inventory	Remove parts that have exceded needed qty and been sold
Fleet Manager	Import a parts list from a vendor	Enter parts that are available for a vehicle
Fleet Manager	View the details of a parts PO	See what is contained in the PO
Fleet Manager	Lookup fuel receipts for vehicles	Lookup fuel qty/\$ entered when vehicle was fueled.
Fleet Manager	Lookup Vehicle Inspection Report	Look at checkout/checkin reports on the status of a vehicle
Fleet Manager	Add Rental Vehicle	Add rental vehicles to cover vehicles in maintenanace

## Employee – Maintenance

As a(n) <actor>	I would like to <description>	So that <outcome>
Maintenance	View a list of vehicles that need to be worked on	I can tell which vehicles need to be fixed
Maintenance	View the details of a vehicle	So I can tell which parts I would need to fix it
Maintenance	Update the vehicle details	I can correct any mistakes, errors, or changes with the vehicles details.
Maintenance	View a list of work that needs completed	I can see what I needs worked on and why
Maintenance	Update existing work orders	So I can explain the work that has been done, even if the issue isn't resolved
Maintenance	Mark work orders as complete	I can signify a task is done
Maintenance	Create work orders	I can report new issues I discover

Maintenance	See a list of previous maintenance logs/work orders	I can look at what work has already been done
Maintenance	See the history of a work order	So I can understand what work has been done over the course of an issue.
Maintenance	Create a compatible part for vehicle	So I can see what parts are compatible with what vehicles
Maintenance	View a list of compatible parts for vehicle	I can see what part goes to what vehicle
Maintenance	Update existing parts compatible for vehicles	Update any parts with new information related to their compatible vehicle
Maintenance	Delete a part that is no longer compatible to a vehicle	I can delete a part that is no longer compatible
Maintenance	Mark inventory items as used	I can report what items I used during maintenance
Maintenance	View list of scheduled maintenance	I know what work needs to be done
Maintenance	Add maintenance to the schedule	I can schedule maintenance
Maintenance	Update a scheduled maintenance	I can change details about the scheduled maintenance
Maintenance	Mark scheduled maintenance as completed	I can signify a task is done
Maintenance	Delete a scheduled maintenance from the system	I can remove unneeded scheduled maintenance orders

## Employee – Mechanic

As a(n) <actor>	I would like to <description>	So that <outcome>
Mechanic	Create parts requests	I can request the parts person to order parts that we don't have in stock
Mechanic	View Parts Requests	See if my request has already been submitted, or update/deactivate requests

Mechanic	Modify parts requests	I can correct wrong requests, or add to my requests
Mechanic	Deactivate a parts request	I can indicate I no longer need this part request
Mechanic	Activate a parts request	I can indicate that I actually -do- need a parts request I thought I didn't

## Employee – Parts Person

As a(n) <actor>	I would like to <description>	So that <outcome>
Parts Person	be able to complete and therefore remove requests from the list of them, and let the mechanic know that the parts have been ordered	allow the mechanic to know that the parts have been ordered, and un-clutter the list of order requests
Parts Person	see a list of parts that mechanics would like to be ordered, with which vehicle and maintenance request that the request is attached to	I can begin deciding the best way to fill each order.
Parts Person	sort the order requests by time, vehicle, etc	narrow down the list of orders to focus on orders from one company, or all similar parts
Parts Person	reject part order requests with a reason why attached	I can remove unnecessary requests, or ask mechanics to revisit what they are ordering/why
Parts Person	see different options for the same part	choose the best option for which part to fill an order with
Parts Person	Add parts to the inventory	let the system tell mechanics that repairs are able to be completed with the updated part inventory
Parts Person	View an inventory of parts, both in stock and ot of stock	I can determine the necessity of ordering parts, and determine that a repair is possible with the current inventory

Parts Person	Sort the view of the parts inventory	to more quickly find the information about the part or parts I am looking for
Parts Person	View past orders for certain parts or vehicles	understand the history of repairs that were done on a certain vehicle
Parts Person	send out parts orders for fulfillment	I can begin the process of getting the parts needed to fix and maintain vehicles
Parts Person	view contact information for vendors	I can contact vendors about problems with ordering or receiving parts
Parts Person	go from part request info to maintenance request info	be able to make an informed decision about which parts to order

## Employee – Operations Manager

As a(n) <actor>	I would like to <description>	So that <outcome>
Operations Manager	Maintain an inventory count and conduct audits	I can ensure a correct count of the current inventory
Operations Manager	Create a Route	Have the route driven by a bus/driver
Operations Manager	Create Drivers/Routes Schedules	I can ensure that our customer's transportation needs are met.
Operations Manager	Run Operations Reports	I can plan for future needs/make adjustments in current resource allocations.
Operations Manager	View info about how vehicle is being operated	I can view driving habits of how customers are using the vehicles.
Operations Manager	Develop a schedule	I can create a schedule of routes and drivers
Operations Manager	Update an existing schedule	I can edit a schedule of routes and drivers
Operations Manager	View a list of schedules	I can see what schedules have been created
Operations Manager	Deactivate a bus route	We no longer use a bus route that is defunct due to obstacles or business rules
Operations Manager	Activate a bus route	We can start using a new route

Operations Manager	Create New Stop	A geographic stop is available in the system to associate with a route.
Operations Manager	List the existing stops in the system.	Stop details can be determined for business purposes.
Operations Manager	Update Existing Stop	The details of an existing stop can be kept up-to-date
Operations Manager	Deactivate an existing, active stop.	The stop is unavailable for use in the system
Operations Manager	Activate an existing but inactive stop	The stop is available for use in the system
Operations Manager	Create New Bus Route	I could define/schedule stops, assign drivers, and assign vehicles
Operations Manager	Search/List the existing bus routes in the system.	Perform business functions (assign drivers or routes; add stops)
Operations Manager	View the details of a bus route.	To gain information to make decisions.
Operations Manager	Update the details of a bus route.	Store current bus route information in the system.
Operations Manager	Deactivate an existing, active bus route	The bus route is unavailable for use in the system
Operations Manager	Activate an existing but inactive bus route	The bus route is available for use in the system

## Employee – Customer Support

As a(n) <actor>	I would like to <description>	So that <outcome>
Customer Support Representative	View the list of customer support tickets	I can effectively manage and respond to customer inquiries and issues
Customer Support Representative	View the details of an individual customer support ticket	I can understand the specific issue and provide assistance to the customer

Customer Support Representative	Update a customer support ticket's information	I can keep a record of actions and progress on the ticket
Customer Support Representative	Mark a customer support ticket as resolved	I can accurately track the progress of the ticket
Customer Support Representative	Create a new ticket that is related to an existing ticket	I can manage and address each issue separately

## Riders – Charter with Driver

As a(n) <actor>	I would like to <description>	So that <outcome>
Charter	reserve a vehicle with a driver	my employees can get to where they need to go without anyone needing a license with a passenger endorsement
Charter	view the details of my reservation for a vehicle with a driver	I can confirm the details
Charter	update details on my reservation for a vehicle with a driver	I can schedule it for a different time
Charter	cancel my reservation	the vehicle and driver are made available again for other users
Charter	verify reservation start	I can inform the system that the reservation has begun
Charter	verify reservation has ended	I can inform the system that the reservation has ended
Charter	reserve a vehicle without a driver	I can rent a vehicle for a trip and drive it myself

## Riders – Charter without Driver

As a(n) <actor>	I would like to <description>	So that <outcome>
Charter w/o Driver	Rent some vehicles for my company	my employees can get to where they need to go
Charter w/o Driver	View vehicles that are reserved	I can look at vehicles I have reserved and for when
Charter w/o Driver	Modify a reservation	I can change details pertaining to my vehicle reservation
Charter w/o Driver	Remove vehicles that are reserved	I can cancel a scheduled vehicle reservation
Charter w/o Driver	return a rented vehicle	the vehicle I borrowed can no longer be in my possession

## Riders – Ride Requester

As a(n) <actor>	I would like to <description>	So that <outcome>
Ride Requester	View All Running Transit Services	I can select a transit service
Ride Requester	View Details of Running Transit Service	I can choose whether I want to use this transit service
Ride Requester	Enter a Start and End Location and Receive Applicable Running Services	I can see which transit services serve my location needs
Ride Requester	Request a Ride	I can receive a time for an available ride
Ride Requester	Request Info on Requested Ride	I can know what time my ride will pick me up at
Ride Scheduler	View All Active Transit Services	I can select a transit service
Ride Scheduler	View Details of Active Transit Service	I can choose whether I want to use this transit service
Ride Scheduler	Enter a Start and End Location and Receive Applicable Active Services	I can see which transit services serve my location needs
Ride Scheduler	Schedule a Ride	I can have a ride scheduled in the system
Ride Scheduler	Request Info on Scheduled Ride	I can know what time my ride will pick me up at

## Riders – Guardian

As a(n) <actor>	I would like to <description>	So that <outcome>
Guardian	Create a profile for my dependent	I can create, read, update, and delete my dependent's ride requests

Guardian	View a list of my dependents	I can create, read, update, and delete my dependent's ride requests or update dependent profiles
Guardian	Update a dependent profile	I can accurately represent my dependent's features or abilities
Guardian	Deactivate my relationship with my dependent	I no longer see the dependent on my account
Guardian	Create a Ride request for my dependent	They can get a vehicle ride
Guardian	View my dependent's existing ride requests	I can update or deactivate the requests if necessary
Guardian	Update ride requests for my dependent	I can change a request if necessary
Guardian	Deactivate my dependent's ride requests	I can indicate the dependent no longer needs this ride
Guardian	Check my dependent's current status (on ride or not, ride location)	I can make sure my dependent is safe

## Outside Contacts

As a(n) <actor>	I would like to <description>	So that <outcome>
Contact Point for the company	Add a Special Service order	We begin work on fixing a majorly damaged item
Contact Point for the company	Update a Special Service order	I can fix errors on what I put in
Contact Point for the company	View the details of a Special Service order	Get up to date on the status of a vehicle
Contact Point for the company	View all Special Service order	Have an overview of our outstanding work
Contact Point for the company	deactivate a special service order	mark a special order as inactive
Contact Point for the company	activate a special service order	reactivate a apseical order
Contact Point for the company	Add a record of a bid we received	Keep a record of this document
Contact Point for the company	update a bid we received	I can fix errors on what I put in
Contact Point for the company	view bids for this particular repair job	Get up to date on the status of a vehicle
Contact Point for the company	view all bids that we have receeived	Work on a plan. Either we need to get more bids, or we can pick a bid to proceed with
Contact Point for the company	deactivate a bad	mark a bid as inacrive
Contact Point for the company	activate a bad	mark a bid as acive
Contact Point for the company	Add an appraisal/inspection of a vehicle before or after major repair work	Record the status of our item
Contact Point for the company	update an appraisal/inspection of a vehicle to fix any errors	I can fix errors on what I put in
Contact Point for the company	View appraisals of a specific vehicle	Get up to date on the status of a vehicle
Contact Point for the company	View all logged appraisals	Have an overview of our outstanding work

Contact Point for the company	deactivate a inspection	mark an inspection as inactive
Contact Point for the company	activate an inspection	mark an inspection as active
Contact Point for the company	Add a work order to a vehicle	Progress the vehicle through the process
Contact Point for the company	Update a work order on a vehicle	I can fix errors on what I put in
Contact Point for the company	View the work order for a vehicle	Get up to date on the status of a vehicle
Contact Point for the company	View all work orders.	Have an overview of our outstanding work
Contact Point for the company	deactivate a work order	mark a work order as inactive
Contact Point for the company	activate a work order	mark a work order as active
Contact Point for the company	Approve a bid	So that work may begin on our asset

# Stakeholders: Use Cases

The Use Cases detail the interfaces between an actor and the application that will be provided to satisfy User Stories. Each User Story will have an attached Use Case. The Use Case will be detailed by its attached User Story, the relevant actor, a description of the Use Case, pre-requisite conditions for the Use Case to occur, the event that triggers the Use Case, the resultant postconditions of the Use Case, the frequency of use of the Use Case, and the assumptions necessary for the Use Case to be applicable.

The normal flow of the Use Case describes the “normal” or regular set of steps involved in the interfacing between the actor and the application. The result of the normal flow should always be successful satisfaction of the attached User Story. Steps in the normal flow may also describe changes in the backend of the system such as in the database.

The alternative flow of the Use Case describes fewer common sets of steps or other deviations from the normal flow. The alternative flow should still result in satisfaction of the User Story.

Exceptions describe contingency circumstances and postconditions for situations in which the actor, the application, and/or the system fail to satisfy the attached User Story. Exceptions can occur due to system failures or due to invalid inputs from the actor.

# System Access – Administrator

## Add Login Profile

<b>Use Case ID:</b>	UC-1.1		
<b>Use Case Name:</b>	Add Login Profile		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/8/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Allows admin to add a profile to the system		
<b>Trigger:</b>	Admin clicks a button labeled “Add Login Profile”		
<b>Preconditions:</b>	1. Admin is logged into the admin panel		
<b>Postconditions:</b>	1. A successfully created user profile		
<b>Normal Flow:</b>	1. Admin logs in 2. Admin clicks on Add Login Info 3. Screen prompts to fill out a profile for a potential employee/user 4. Information is entered 5. Click Submit 6. The profile is created 7. Returns to admin panel		
<b>Alternative Flows:</b>	2a, 3a, 4a Admin decides to cancel  1. Resume normal flow of use case at step 7 of normal flow		
<b>Exceptions:</b>	4a. Admin enters the wrong datatype  1. System prompts to enter the correct one 2. Use case resumes on step 4 of normal flow		
<b>Includes:</b>			
<b>Frequency of Use:</b>	Anywhere from daily to yearly		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin knows English  Admin understands how the data is stored  Admin knows basic info of the person they are entering		
<b>Notes and Issues:</b>			

## View Login Profile

<b>Use Case ID:</b>	ADM.1.2		
<b>Use Case Name:</b>	View Login Profile		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	View the profile of any user		
<b>Trigger:</b>	Desire to see information on a user		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin has an active administrator account</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin can see the profile of the selected user</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Admin logs in</li> <li>2. Admin sees a message on screen showing credentials have been taken.</li> <li>3. Admin gains access to admin panel.</li> <li>4. Admin clicks User Profiles, navigates to drop down View profiles</li> <li>5. Admin can sort through the list of profiles</li> <li>6. Click on a specific profile</li> <li>7.</li> </ol>		
<b>Alternative Flows:</b>	5a. In step 5 of normal flow if the sort criteria doesn't return a profile		
<b>[Alternative Flow 1 - Profile not found]</b>	<ol style="list-style-type: none"> <li>1. No users found please refine search</li> <li>2. Use case resumes on step 5</li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Admin understands English</p> <p>Admin knows what criteria they are searching for</p>		
<b>Notes and Issues:</b>			

## Edit Login Profile

<b>Use Case ID:</b>	ADM.1.3		
<b>Use Case Name:</b>	Edit Login Profile		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Edit the profile of any user		
<b>Trigger:</b>	Desire to edit information on a user		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin has an active administrator account</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin can see the profile of the selected user</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Admin logs in</li> <li>2. Admin sees a message on screen showing credentials have been taken.</li> <li>3. Admin gains access to admin panel.</li> <li>4. Admin clicks User Profiles, navigates to drop down view profiles</li> <li>5. Admin can sort through the list of profiles</li> <li>6. Click on a specific profile</li> <li>7. Click edit profile</li> <li>8. Change whatever needs to be changed</li> <li>9. Click save</li> <li>10. Goes back to the initial list of profiles</li> </ol>		
<b>Alternative Flows:</b>	5a. In step 5 of normal flow if the sort criteria doesn't return a profile		
[Alternative Flow 1 - Profile not found]	<ol style="list-style-type: none"> <li>1. No users found please refine search</li> <li>2. Use case resumes on step 5</li> </ol>		
<b>Exceptions:</b>	8a. In step 8 of normal flow if the new information entered does not fit the datatype. <ol style="list-style-type: none"> <li>1. Displays invalid entry and prevents it from being modified</li> <li>2. Resume on step 8</li> </ol>		
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Admin understands English.</p> <p>Admin knows what criteria they are searching for.</p> <p>Admin knows what information to edit</p>		

<b>Notes and Issues:</b>
--------------------------

## Deactivate Login Profile

<b>Use Case ID:</b>	ADM.1.4		
<b>Use Case Name:</b>	Deactivate Login Profile		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Deactivate the profile of any user		
<b>Trigger:</b>	Desire to deactivate a user		
<b>Preconditions:</b>	<p>1. Admin has an active administrator account</p>		
<b>Postconditions:</b>	<p>1. Admin deactivates the profile of the selected user</p>		
<b>Normal Flow:</b>	<p>1. Admin logs in          2. Admin sees a message on screen showing credentials have been taken.          3. Admin gains access to admin panel.          4. Admin clicks User Profiles, navigates to drop down view profiles          5. Admin can sort through the list of profiles          6. Click on a specific profile          7. Click edit profile          8. Click Deactivate          9. Click save          10. Goes back to the initial list of profiles</p>		
<b>Alternative Flows:</b>	5a. In step 5 of normal flow if the sort criteria doesn't return a profile		
[Alternative Flow 1 - Profile not found]	<p>1. No users found please refine search          2. Use case resumes on step 5</p>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Admin understands English.          Admin knows what criteria they are searching for.          Admin knows what information to edit</p>		
<b>Notes and Issues:</b>			

## Reactivate Login Profile

<b>Use Case ID:</b>	ADM.1.5		
<b>Use Case Name:</b>	Reactivate Login Profile		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Reactivate the profile of any user		
<b>Trigger:</b>	Desire to deactivate a user		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Admin has an active administrator account</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Admin reactivates the profile of the selected user</li> </ul>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Admin logs in</li> <li>2. Admin sees a message on screen showing credentials have been taken.</li> <li>3. Admin gains access to admin panel.</li> <li>4. Admin clicks User Profiles, navigates to drop down view profiles</li> <li>5. Admin can sort through the list of profiles</li> <li>6. Click on a specific profile</li> <li>7. Click edit profile</li> <li>8. Click Reactivate</li> <li>9. Click save</li> <li>10. Goes back to the initial list of profiles</li> </ol>		
<b>Alternative Flows:</b>	5a. In step 5 of normal flow if the sort criteria doesn't return a profile		
<b>[Alternative Flow 1 - Profile not found]</b>	<ol style="list-style-type: none"> <li>1. No users found please refine search</li> <li>2. Use case resumes on step 5</li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Admin understands English.</p> <p>Admin knows what criteria they are searching for.</p> <p>Admin knows what information to edit</p>		
<b>Notes and Issues:</b>			

## Create Employee Profile

<b>Use Case ID:</b>	ADM.2.1		
<b>Use Case Name:</b>	Create Employee Profile		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Create an employee profile		
<b>Trigger:</b>	Desire to create information on a employee		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Admin has an active administrator account</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Admin can see the profile of the selected user</li> </ul>		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Admin logs in</li> <li>2. Admin sees a message on screen showing credentials have been taken.</li> <li>3. Admin gains access to admin panel.</li> <li>4. Admin clicks User Profiles, navigates to drop down View profiles</li> <li>5. Admin can sort through the list of profiles</li> <li>6. Click on a specific profile</li> <li>7.</li> </ul>		
<b>Alternative Flows:</b> [Alternative Flow 1 - Profile not found]	<p>5a. In step 5 of normal flow if the sort criteria doesn't return a profile</p> <ul style="list-style-type: none"> <li>1. No users found please refine search</li> <li>2. Use case resumes on step 5</li> </ul>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Admin understands English</p> <p>Admin knows what criteria they are searching for</p>		
<b>Notes and Issues:</b>			

## View Employee Profile

<b>Use Case ID:</b>	ADM.2.2		
<b>Use Case Name:</b>	View Employee Profile		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	View the profile of any Employee		
<b>Trigger:</b>	Desire to see information on a employee		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin has an active administrator account</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin can see the profile of the selected employee</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Admin logs in</li> <li>2. Admin sees a message on screen showing credentials have been taken.</li> <li>3. Admin gains access to admin panel.</li> <li>4. Admin clicks Employee Profiles, navigates to drop down View profiles</li> <li>5. Admin can sort through the list of profiles</li> <li>6. Click on a specific profile</li> <li>7.</li> </ol>		
<b>Alternative Flows:</b>	5a. In step 5 of normal flow if the sort criteria doesn't return a profile		
[Alternative Flow 1 - Profile not found]	<ol style="list-style-type: none"> <li>1. No users found please refine search</li> <li>2. Use case resumes on step 5</li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Admin understands English</p> <p>Admin knows what criteria they are searching for</p>		
<b>Notes and Issues:</b>			

## Edit Employee Profile

<b>Use Case ID:</b>	ADM.2.3		
<b>Use Case Name:</b>	Edit Employee Profile		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Edit the profile of any employee		
<b>Trigger:</b>	Desire to edit information on an employee		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin has an active administrator account</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin can see the profile of the selected user</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Admin logs in</li> <li>2. Admin sees a message on screen showing credentials have been taken.</li> <li>3. Admin gains access to admin panel.</li> <li>4. Admin clicks Employee Profiles, navigates to drop down view profiles</li> <li>5. Admin can sort through the list of profiles</li> <li>6. Click on a specific profile</li> <li>7. Click edit profile</li> <li>8. Change whatever needs to be changed</li> <li>9. Click save</li> <li>10. Goes back to the initial list of profiles</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 - Profile not found]	<p>5a. In step 5 of normal flow if the sort criteria doesn't return a profile</p> <ol style="list-style-type: none"> <li>1. No users found please refine search</li> <li>2. Use case resumes on step 5</li> </ol>		
<b>Exceptions:</b>	<p>8a. In step 8 of normal flow if the new information entered does not fit the datatype.</p> <ol style="list-style-type: none"> <li>1. Displays invalid entry and prevents it from being modified</li> <li>2. Resume on step 8</li> </ol>		
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Admin understands English.</p> <p>Admin knows what criteria they are searching for.</p> <p>Admin knows what information to edit</p>		

<b>Notes and Issues:</b>	

## Add/Remove Roles

<b>Use Case ID:</b>	ADM-2.4		
<b>Use Case Name:</b>	Add/Remove Roles		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	9/8/2023		<b>Last Revision Date:</b> 9/10/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Allows admin to edit user roles		
<b>Trigger:</b>	Admin clicks a button labeled edit users		
<b>Preconditions:</b>	1. Admin is logged into the admin panel		
<b>Postconditions:</b>	1. A successfully created user profile		
<b>Normal Flow:</b>	1. Admin logs in 2. Admin clicks edit users 3. Admin searches for the user they want to edit 4. Clicks edit permissions for selected user 5. Dropdown list of all the roles with a searchable bar appears 6. Select role wanted to add 7. Click apply role 8. User now has those permissions 9. Return to edit users screen 10. Return to admin panel		
<b>Alternative Flows:</b> [Alternative Flow 1 - Admin wants to cancel]	2a, 3a, 4a, 5a, 6a, Admin decides to cancel  1. Resume normal flow of use case at step 9 of normal flow		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	Anywhere from daily to yearly		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin knows English  Admin understands how the data is stored  Admin knows what role they need to give the person		
<b>Notes and Issues:</b>	<b>Overlapping permissions may cause issues</b>		

## Deactivate Employee Account

<b>Use Case ID:</b>	ADM.2.5		
<b>Use Case Name:</b>	Deactivate Employee Account		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Deactivate the profile of any Employee		
<b>Trigger:</b>	Desire to deactivate an employees account		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Admin has an active administrator account</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Admin deactivates the account of the selected employee</li> </ul>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Admin logs in</li> <li>2. Admin sees a message on screen showing credentials have been taken.</li> <li>3. Admin gains access to admin panel.</li> <li>4. Admin clicks Employee, navigates to drop down view profiles</li> <li>5. Admin can sort through the list of profiles</li> <li>6. Click on a specific profile</li> <li>7. Click edit profile</li> <li>8. Click Deactivate</li> <li>9. Click save</li> <li>10. Goes back to the initial list of profiles</li> </ol>		
<b>Alternative Flows:</b>	5a. In step 5 of normal flow if the sort criteria doesn't return a profile <ul style="list-style-type: none"> <li>1. No users found please refine search</li> <li>2. Use case resumes on step 5</li> </ul>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Admin understands English.</p> <p>Admin knows what criteria they are searching for.</p>		
<b>Notes and Issues:</b>			

## Reactivate Employee Account

<b>Use Case ID:</b>	ADM.2.6		
<b>Use Case Name:</b>	Reactivate Employee Account		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Reactivate the profile of any Employee		
<b>Trigger:</b>	An employee has been rehired and needs to be reactivated		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin has an active administrator account</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Admin reactivates the account of the selected employee</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Admin logs in</li> <li>2. Admin sees a message on screen showing credentials have been taken.</li> <li>3. Admin gains access to admin panel.</li> <li>4. Admin clicks Employee, navigates to drop down view profiles</li> <li>5. Admin can sort through the list of profiles (show deactivated)</li> <li>6. Click on a specific profile</li> <li>7. Click edit profile</li> <li>8. Click Reactivate</li> <li>9. Click save</li> <li>10. Goes back to the initial list of profiles</li> </ol>		
<b>Alternative Flows:</b>	5a. In step 5 of normal flow if the sort criteria doesn't return a profile <ol style="list-style-type: none"> <li>1. No users found please refine search</li> <li>2. Use case resumes on step 5</li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin understands English. Admin knows what criteria they are searching for.		
<b>Notes and Issues:</b>			

## Create System Message

<b>Use Case ID:</b>	ADM.3.1		
<b>Use Case Name:</b>	Create System Message		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/10/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Allows admin to create a system wide message		
<b>Trigger:</b>	Admin clicks a button labeled system message		
<b>Preconditions:</b>	1. Admin is logged into the admin panel		
<b>Postconditions:</b>	1. The message is displayed when anyone logs in at the top of the screen in a clickable banner		
<b>Normal Flow:</b>	1. Admin logs in 2. Admin clicks Create System message 3. Admin types in the message they want 4. Clicks on send 5. Message appears on everyone's screen at login 6. Admin returns to admin panel		
<b>Alternative Flows:</b>	2a, 3a,, Admin decides to cancel		
<b>[Alternative Flow 1 - Admin wants to cancel]</b>	1. Resume normal flow of use case at step 6 of normal flow		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	Anywhere from daily to yearly		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin knows English Admin understands how the data is stored		
<b>Notes and Issues:</b>	Might need a way to put it as a banner or something so it doesn't annoy the user consistently		

## View Past System Message

<b>Use Case ID:</b>	ADM.3.2		
<b>Use Case Name:</b>	View Past System Message		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Allows admin to create a system wide message		
<b>Trigger:</b>	Admin clicks a button labeled system message		
<b>Preconditions:</b>	1. Admin is logged into the admin panel		
<b>Postconditions:</b>	1. The message is displayed when anyone logs in at the top of the screen in a clickable banner		
<b>Normal Flow:</b>	1. Admin logs in 2. Admin clicks view past System messages 3. Admin can filter the date of the system message 4. Admin selects from a list 5. Admin Clicks on the past message 6. Has the option to click send again as well 7. Admin returns to admin panel		
<b>Alternative Flows:</b> [Alternative Flow 1 - Admin wants to cancel]	2a, 3a,, Admin decides to cancel  1. Resume normal flow of use case at step 6 of normal flow		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	Anywhere from daily to yearly		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin knows English  Admin understands that everyone will see the message they put out		
<b>Notes and Issues:</b>	Might need a way to put it as a banner or something so it doesn't annoy the user consistently		

## Change System Message

<b>Use Case ID:</b>	ADM.3.3		
<b>Use Case Name:</b>	Change System Message		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Allows admin to create a system wide message		
<b>Trigger:</b>	Admin clicks a button labeled system message		
<b>Preconditions:</b>	1. Admin is logged into the admin panel		
<b>Postconditions:</b>	1. The message is displayed when anyone logs in at the top of the screen in a clickable banner		
<b>Normal Flow:</b>	1. Admin logs in 2. Admin clicks view past System messages 3. Admin can filter the date of the system message 4. Admin selects from a list 5. Admin Clicks on the past message 6. Click edit message 7. Click confirm 8. Admin returns to admin panel		
<b>Alternative Flows:</b>	2a, 3a,, Admin decides to cancel		
[Alternative Flow 1 - Admin wants to cancel]	1. Resume normal flow of use case at step 6 of normal flow		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	Anywhere from daily to yearly		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin knows English  Admin understands that everyone will see the message they put out		
<b>Notes and Issues:</b>	<b>Might need a way to put it as a banner or something so it doesn't annoy the user consistently</b>		

## View Past System Message

<b>Use Case ID:</b>	ADM.3.4		
<b>Use Case Name:</b>	Change System Message		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/1/2023	<b>Last Revision Date:</b>	10/2/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Allows admin to create a system wide message		
<b>Trigger:</b>	Admin clicks a button labeled system message		
<b>Preconditions:</b>	1. Admin is logged into the admin panel		
<b>Postconditions:</b>	1. The message is displayed when anyone logs in at the top of the screen in a clickable banner		
<b>Normal Flow:</b>	1. Admin logs in 2. Admin clicks view past System messages 3. Admin can filter the date of the system message 4. Admin selects from a list 5. Admin Clicks on the past message 6. Click edit message 7. Click confirm 8. Admin returns to admin panel		
<b>Alternative Flows:</b> [Alternative Flow 1 - Admin wants to cancel]	2a, 3a,, Admin decides to cancel 1. Resume normal flow of use case at step 6 of normal flow		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	Anywhere from daily to yearly		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin knows English Admin understands that everyone will see the message they put out		
<b>Notes and Issues:</b>	<b>Might need a way to put it as a banner or something so it doesn't annoy the user consistently</b>		

## View as Other Roles

<b>Use Case ID:</b>	UC-6.1		
<b>Use Case Name:</b>	View as other roles		
<b>Created By:</b>	Jacob Rohr	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/10/2023
<b>Actors:</b>	Admin		
<b>Description:</b>	Allows admin to view the system as another role would similar to a teacher viewing talon in student mode		
<b>Trigger:</b>	Admin clicks a button labeled change view		
<b>Preconditions:</b>	1. Admin is logged into the admin panel		
<b>Postconditions:</b>	1. Admin can see other roles		
<b>Normal Flow:</b>	1. Admin logs in 2. Admin clicks change view 3. Admin scrolls through to find the role they want to see viewed 4. Click on the desired role 5. Do whatever you need to as the other role 6. Close out of role selector 7. Return to admin panel		
<b>Alternative Flows:</b> [Alternative Flow 1 - Admin wants to cancel]	2a-6a Admin decides to cancel  1. Resume normal flow of use case at step 7 of normal flow		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	daily to yearly		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin knows English  Admin understands how to write emails		
<b>Notes and Issues:</b>	Would need to make sure any changes made are registered as being made by the admin instead.		

# System Access – User

## Account Creation

<b>Use Case ID:</b>	U-USER.01.01		
<b>Use Case Name:</b>	Create Account		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-07	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	User		
<b>Description:</b>	The user creates an account.		
<b>Trigger:</b>	The user accesses the account creation page.		
<b>Preconditions:</b>	1. No account is logged into the local client of the application. 2. User is viewing the account creation page 3. Local client has network access to the application		
<b>Postconditions:</b>	Success 1. System contains a record of user account information 2. User is logged into the system 3. System displays the main menu corresponding to the user's type Failures 4. System displays an error message with the reason for failure		
<b>Normal Flow:</b>	1. User enters data into required account information fields 2. System prompts user to confirm results 3. System creates a record of the new account 4. System logs user into account 5. System displays main menu		
<b>Alternative Flows:</b>	<b>Exceptions:</b> 1a. In step 1 of the normal flow, if the user enters credentials corresponding with an existing account 1. System displays an account exists error message and link to login form 2. User amends affected fields 3. System proceeds to normal flow step 2  1b. In step 1 of the normal flow, if the user enters unique credentials corresponding with an existing account 1. System displays an account exists error message and link to login form 2. User navigates to login form 3. System proceeds to use case UC-22.2.0, "Log In"  1c. In step 1 of the normal flow, if the user enters invalid credentials in one or more of the fields 1. System displays a corresponding error message 2. System highlights the invalid field(s) on the form  3a. In step 3 of the normal flow, the system is unable to create a record for the new account 3. System displays a relevant system error message		
<b>Includes:</b>	1. Account Login 2. View Main Menu		
<b>Frequency of Use:</b>	[On Demand.]		

<b>Special Requirements:</b>	
<b>Assumptions:</b>	Users requiring extended permissions (primarily employees) will have their accounts created or permissions adjusted by an administrator.
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Should non-required fields be treated the same as required fields?</li> <li>2. What if any measures should be taken to prevent a banned or suspended user from creating a new account?</li> </ol>

## Account Login

<b>Use Case ID:</b>	UC-USER.01.02		
<b>Use Case Name:</b>	Log In		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-07	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	User		
<b>Description:</b>	The user enters a username and password to log into the application.		
<b>Trigger:</b>	User opens the application's log-in page or expands the log-in form menu item		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User must be viewing the login form of the application</li> <li>2. User has a preexisting account</li> <li>3. User account is currently activated</li> <li>4. No account is logged into the local client of the application</li> <li>5. Local client has network access to the application</li> </ol>		
<b>Postconditions:</b>	<p>Success</p> <ol style="list-style-type: none"> <li>1. Instances of login form have been replaced with log out buttons or account settings links</li> <li>2. Application displays a menu or other splash page</li> <li>3. Other accounts are blocked from logging into the local client of the application</li> <li>4. System contains a record documenting the log in</li> <p>Failures</p> <li>5. Login form displays an account reactivation prompt</li> <li>6. Login form displays an incorrect password error</li> <li>7. Use of an incorrect password with a valid username is recorded and added to a count</li> <li>8. Login form displays a password reset prompt and link</li> <li>9. Account is locked</li> <li>10. Login form displays an invalid username error</li> <li>11. Login form displays an account recovery prompt and link</li> <li>12. Login form displays a network connection error message</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User enters username</li> <li>2. User enters password</li> <li>3. User submits login form</li> <li>4. System refreshes view to relevant user menu</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Deactivated Account]	<p>3a. In step 3 of the normal flow, if the user enters credentials for a deactivated account</p> <ol style="list-style-type: none"> <li>1. System will prompt user to reactivate account</li> <li>2. User accepts</li> <li>3. System displays reactivation form</li> </ol> <p>3b. In step 3 of the normal flow, if the user enters credentials for a deactivated account</p> <ol style="list-style-type: none"> <li>1. System will prompt user to reactivate account</li> <li>2. User declines</li> <li>3. System displays main login form</li> </ol>		
<b>Exceptions:</b>	<p>1a. In step 1 of the normal flow, if the user enters an invalid username</p> <ol style="list-style-type: none"> <li>1. System displays an error message and account recovery or creation prompt on the login menu</li> <li>2. User enters a valid username and resubmits</li> <li>3. System refreshes to relevant user menu</li> </ol>		

	<p>1b. In step 1 of the normal flow, if the user enters an invalid username</p> <ol style="list-style-type: none"> <li>1. System displays an error message and account recovery or creation prompt on the login menu</li> <li>2. User enters an invalid username and resubmits</li> <li>3. System redisplays an error message and account recovery prompt</li> </ol> <p>1c. In step 1 of the normal flow, if the user enters an invalid username</p> <ol style="list-style-type: none"> <li>1. System displays an error message and account recovery or creation prompt on the login menu</li> <li>2. User selects the account creation link</li> <li>3. System displays the account creation page</li> </ol> <p>1d. In step 1 of the normal flow, if the user enters an invalid username</p> <ol style="list-style-type: none"> <li>1. System displays an error message and account recovery or creation prompt on the login menu</li> <li>2. User selects the account recovery link</li> <li>3. System displays the account recovery page</li> </ol> <p>2a. In step 2 of the normal flow, if the user enters an invalid password</p> <ol style="list-style-type: none"> <li>1. System displays an error message and password reset prompt on the login menu</li> <li>2. User enters a valid password and resubmits</li> <li>3. System refreshes to relevant user menu</li> </ol> <p>2a. In step 2 of the normal flow, if the user enters an invalid password</p> <ol style="list-style-type: none"> <li>1. System displays an error message and password reset prompt on the login menu</li> <li>2. User repeatedly enters an invalid password</li> <li>3. System flags account and prevents logging in to the associated username</li> <li>4. System redirects to home page</li> </ol> <p>3a. In step 3 of the normal flow, if the system is unable to contact the server to validate login information</p> <ol style="list-style-type: none"> <li>1. System displays a network error message on the login menu</li> </ol>
<b>Includes:</b>	Account Logout Create Account Retrieve Account Reactive Account Reset Password View Main Menu
<b>Frequency of Use:</b>	[On Demand.]
<b>Special Requirements:</b>	

<b>Assumptions:</b>	<p>Account registration and login will be necessary to use most features of the application.</p> <p>The same overall login process should be used by all types of users, with user experiences diverging only after login.</p>
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. What are the format requirements for the login credentials?</li> <li>2. Should formal validation of credentials be used as a filter prior to account validation?</li> <li>3. How many attempts are allowed before an invalid password locks an account?</li> <li>4. What, if any, automatic log out procedures are in place? How will this affect the frequency of use?</li> </ol>

## Account Logout

<b>Use Case ID:</b>	UC-USER.01.04		
<b>Use Case Name:</b>	Reset Password		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	Michael Springer
<b>Date Created:</b>	2023-09-08	<b>Last Revision Date:</b>	2023-09-24
<b>Actors:</b>	User		
<b>Description:</b>	The user resets their account password		
<b>Trigger:</b>	User activates the reset password button from the login page		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. User must be viewing a password reset page or pop up</li> <li>2. User has a preexisting account.</li> <li>3. User has entered a valid username</li> </ul>		
<b>Postconditions:</b>	<p>Success</p> <ul style="list-style-type: none"> <li>1. System holds record of the new password as user's active password</li> <li>2. System has flagged or moved the user's previous password as inactive</li> <li>3. User is logged in</li> <li>4. System displays user settings page</li> </ul> <p>Failure</p> <ul style="list-style-type: none"> <li>5. Password reset feature is locked for the user account</li> <li>6. User has message(text or email) regarding failed reset effort</li> <li>7. System has log of failed attempt</li> <li>8. System displays an error message</li> </ul>		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. User selects "reset password"</li> <li>2. System prompts user for security method (text, email, recovery question)</li> <li>3. User selects their default method (e.g., text)</li> <li>4. System sends a confirmation link to user's phone</li> <li>5. User activates link</li> <li>6. System displays new password form</li> <li>7. User enters password twice to assure spelling</li> <li>8. System validates password for match and formal requirements</li> <li>9. User submits new password</li> <li>10. System logs user in and displays user settings page</li> </ul>		
<b>Alternative Flows:</b> [Alternative Flow 1 – recovery via email]	<p>3a. In step 3 of the normal flow, if the user selects email reset</p> <ul style="list-style-type: none"> <li>1. System sends a confirmation link to user's email</li> <li>2. User opens email and selects link</li> <li>3. Return to step 6 of normal flow: system displays a new password form</li> </ul>		
<b>Alternative Flows:</b> [Alternative Flow 2 – recovery via security questions]	<p>3b. In step 3 of the normal flow, the user selects security questions</p> <ul style="list-style-type: none"> <li>1. System prompts user to reenter username</li> <li>2. User enters username and submits</li> <li>3. System displays a form with 2 of the user's security question</li> <li>4. User answers security questions correctly</li> <li>5. Return to step 6 of normal flow: system displays a new password form</li> </ul>		
<b>Exceptions:</b>	<p>3b.a. In step 4 of alternative flow 3b, the user enters invalid answers</p> <ul style="list-style-type: none"> <li>1. System displays error message and prompts for an additional try</li> </ul>		

	<ol style="list-style-type: none"> <li>2. Repeat step 1</li> <li>3. User repeatedly answers incorrectly (3 tries?)</li> <li>4. System displays error message with link to help</li> <li>5. System locks password recovery for the account</li> <li>6. System sends warning email to email associated with account</li> <li>7. System adds an in-application notification about attempted logins to the account</li> <li>6a. In step six of normal flow, the password reset link has expired           <ol style="list-style-type: none"> <li>1. System displays error message and prompts user to return to login page</li> </ol> </li> </ol>
<b>Includes:</b>	Account Login User Preferences
<b>Frequency of Use:</b>	[On Demand.]
<b>Special Requirements:</b>	
<b>Assumptions:</b>	User knows their username or has already gone through the Account Retrieval use case
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. What help page or other support should be provided once a user is locked out of the password reset process?</li> <li>2. How many attempts can be made to answer security questions?</li> <li>3. Is it necessary to have three methods (questions, email, text)?</li> <li>4. Should there be a time-based limit on number of password resets?</li> <li>5. Are users allowed to enter a previous password as a new password?</li> </ol>

## Account Retrieval

<b>Use Case ID:</b>	UC-USER.01.05		
<b>Use Case Name:</b>	Retrieve Username		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-09	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	User		
<b>Description:</b>	User retrieves their username		
<b>Trigger:</b>	The user navigates to the application's account retrieval form		
<b>Preconditions:</b>	1. An account with the user's contact and retrieval information exists 2. No one is logged into the local instance of the application 3. User is viewing the account retrieval form		
<b>Postconditions:</b>	Success 1. A message has been sent (via email or text) with the user's username 2. System displays a confirmation page Failure 3. System displays an error message with links to other recovery method and account creation page		
<b>Normal Flow:</b>	1. User activates account retrieval link 2. System prompts user to input an email address or phone number associated with the account 3. User submits their email account 4. System sends a message documenting the recovery attempt and including the username to the email account 5. System displays a message stating the recovery email has been sent		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Text Retrieval]</b>	3a. In step 3 of the normal flow, if the user selects recovery via text 1. System sends a message documenting the recovery attempt and including the username to the phone number 2. System displays a message stating the recovery text has been sent		
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the customer enters an invalid email address 1. System displays an error message with phone recovery option and account creation prompt  3b. In step 3 of the normal flow, if the customer enters an invalid phone number 1. System displays an error message with email recovery option and account creation prompt		
<b>Includes:</b>	Create Account		
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	User is required to enter data for recovery methods during the account creation process		
<b>Notes and Issues:</b>	1. Messages for errors or confirming a recovery message has been sent could be handled in several ways. They could remain in a static view until the user navigates away, or (as a popup item) they could automatically redirect to a login screen when the user dismisses them. 2. Should there be any limit on account recovery attempts? 3. Should the system log such attempts? If so, should it send a notification to the user about these attempts?		

## Account Deactivation

<b>Use Case ID:</b>	UC-USER.01.06		
<b>Use Case Name:</b>	Deactivate Account		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-09	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	User		
<b>Description:</b>	The user sets their account to deactivated status		
<b>Trigger:</b>	User activates the "Deactivate Account" link in the User Preferences menu		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User must have an existing account</li> <li>2. user account must be activated</li> <li>3. User must be logged in</li> <li>4. User is viewing the account deactivation form</li> <li>5. User has system permission to deactivate account</li> <li>6. Local client has network access to the application.</li> </ol>		
<b>Postconditions:</b>	<p>Success</p> <ol style="list-style-type: none"> <li>1. The system has flagged the account as "Deactivated"</li> <li>2. The user is logged out</li> <li>3. The system displays the home page</li> </ol> <p>Failure</p> <ol style="list-style-type: none"> <li>4. The system displays an error message</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User activates the deactivation button</li> <li>2. System displays a notice explaining deactivation, reactivation, and prompting the user to enter their password and confirm their choice</li> <li>3. User enters password</li> <li>4. User confirms deactivation</li> <li>5. System flags account as deactivated</li> <li>6. System displays a message explaining the reactivation process</li> <li>7. User closes message</li> <li>8. System logs user out</li> <li>9. System displays home page</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Decline Deactivation]	<ol style="list-style-type: none"> <li>3a. In step 3 of the normal flow, if the user declines to deactivate their account             <ol style="list-style-type: none"> <li>1. System displays user preferences page</li> </ol> </li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>4a. In step 4 of normal flow, user enters the incorrect password             <ol style="list-style-type: none"> <li>1. System displays an error message</li> <li>2. User closes message</li> <li>3. System displays user preferences page</li> </ol> </li> <li>5b. In step 5 of normal flow, network or system errors prevent changing the status of the account             <ol style="list-style-type: none"> <li>1. System displays an error message</li> </ol> </li> </ol>		
<b>Includes:</b>	Account Logout User Preferences		
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Some users (like employees) will not have system permission to deactivate their own account. In these cases, the "Deactivate Account" menu item will be deactivated or removed.</p> <p>The program incorporates account deactivation rather than deletion.</p>		
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Will the system include any automatic triggers for deactivation?</li> </ol>		

	<ol style="list-style-type: none"><li>2. Will some users, other than admin (like managers) have the ability to manually deactivate other user accounts?</li><li>3. Development team will need to decide what features are deactivated with the account e.g., will user continue to receive email updates?</li><li>4. Should long-term deactivation lead to an automated deletion?</li><li>5. Are there other systematic conditions for permanent deletion?</li><li>6. Should long periods of inactivity result in account being deactivated?</li></ol>
--	--

## Account Reactivation

<b>Use Case ID:</b>	UC-USER.01.07		
<b>Use Case Name:</b>	Reactivate Account		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-09	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	User		
<b>Description:</b>	The user reactivates their account		
<b>Trigger:</b>	User completes the login process for a deactivated account		
<b>Preconditions:</b>	1. Account must exist 2. Account must be flagged as deactivated 3. User must have logged in via the account login feature 4. Local client has network access to the application.		
<b>Postconditions:</b>	Success 1. Account is flagged as active 2. System displays main menu Failure 3. System displays an error message		
<b>Normal Flow:</b>	1. User logs into a deactivated account 2. System displays a prompt for account reactivation or logout 3. User confirms reactivation 4. System flags user account as active 5. System displays notification of reactivation 6. User closes notification 7. System displays main menu		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – User Declines Reactivation]</b>	3a. In step 3 of the normal flow, if the user selects logout rather than reactivation 1. System logs user out		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, the account has been flagged as locked 1. System displays an error message with details of account recovery or ban appeal  4a. In step 4 of the normal flow, a network or system error prevents updating the account status 1. System displays a relevant error message  4b. In step 4 of normal flow, user credentials are no longer valid and must be updated 1. System prompts user to update credentials with a locked-screen form 2. User updates credentials with valid values 3. Return to step 4 of Normal flow		
<b>Includes:</b>	Account Logout Update Account Main Menu		
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	User has successfully completed the account login process		
<b>Notes and Issues:</b>	1. If a user (like an employee) formerly had extended privileges, and their account was deactivated when they left that role, should they be able to		

	activate a basic account with their existing credentials or should they have to create a new account? Assuming a mechanic could also be a rider and use one account for both roles, it would create discontinuity to force them to create an entirely new account.
--	--

## User Preferences

<b>Use Case ID:</b>	UC-User.01.08		
<b>Use Case Name:</b>	Modify Personal Info		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-09	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	User		
<b>Description:</b>	The user updates account and personal information		
<b>Trigger:</b>	User navigates to or is directed to the account update form		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User account exists</li> <li>2. User is logged in</li> <li>3. User account is active <i>or</i> user is directed from account reactivation feature</li> <li>4. User has system permissions to modify account information</li> </ol>		
<b>Postconditions:</b>	<p>Success</p> <ol style="list-style-type: none"> <li>1. System contains an updated record of user's account credentials and personal information</li> <li>2. System no longer uses previous information for business processes—login, default pickup address, etc.</li> <li>3. System displays the account information form</li> </ol> <p>Failures</p> <ol style="list-style-type: none"> <li>4. System displays an error message</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User updates desired fields in the account information form</li> <li>2. User selects "save changes"</li> <li>3. System prompts user to confirm changes, and to reenter existing password if changing password</li> <li>4. User confirms changes</li> <li>5. System stores new field values, associating them with account functions</li> <li>6. System returns to the account information form</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – User reverts fields]	<ol style="list-style-type: none"> <li>1a. In step 1 of the normal flow, the user clicks a revert button after modifying fields</li> <li>1. System displays refreshes the account information form populated with previous values</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 2 – User declines changes]	<ol style="list-style-type: none"> <li>4a. In step 4 of the normal flow, if the user declines the changes</li> <li>1. System displays account information form with values the user had amended</li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1a. In step 1 of the normal flow, if the customer enters an invalid value for a field             <ol style="list-style-type: none"> <li>1. System highlights the invalid field and deactivates the "save changes" button until a valid value is entered</li> </ol> </li> <li>4a. In step 4 of the normal flow, if the customer enters an invalid password             <ol style="list-style-type: none"> <li>1. System displays an error message and prompts user to reenter both the new and the old password</li> </ol> </li> <li>5a. In step 5 of the normal flow, if a network or system error prevents updating the records             <ol style="list-style-type: none"> <li>1. System displays an error message</li> <li>2. System returns to the form with the values the user had amended</li> </ol> </li> </ol>		
<b>Includes:</b>	Reset password		
<b>Frequency of Use:</b>	[On Demand.]		

<b>Special Requirements:</b>	
<b>Assumptions:</b>	<p>Different user roles will have different levels of permission for changing account information e.g., an employee unable to change their “position” field.</p> <p>Admin will have the ability to customize the fields used by the system as well as default permission levels.</p> <p>The section of User Preferences combines account credentials and personal information, but is distinct from sections related to system, payment, and communications settings.</p>
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. See 22.1.1, “Account Creation” and 22.4.1, “Reset Password” for considerations related to selecting and modifying a password</li> <li>2. What fields should be required/not required?</li> <li>3. How should we handle bad data?</li> </ol>

## Change UX Settings

<b>Use Case ID:</b>	UC-USER.01.09		
<b>Use Case Name:</b>	Change UX Settings		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-09	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	User		
<b>Description:</b>	The user views and edits UX features		
<b>Trigger:</b>	User navigates to the system settings form		
<b>Preconditions:</b>	1. User account exists 2. User is logged in 3. User account is active <i>or</i> user is directed from account reactivation feature 4. User has system permissions to modify system settings information		
<b>Postconditions:</b>	Success 1. System contains an updated record of user's selected settings 2. System no longer uses previous settings for presentation processes—theme, accessibility features, etc. 3. System displays the system settings form Failures 4. System displays an error message		
<b>Normal Flow:</b>	1. User updates desired fields in the system settings form 2. User selects "save changes" 3. System prompts user to confirm changes 4. User confirms changes 5. System stores new field values, associating them with account functions 6. System returns to the system settings form		
<b>Alternative Flows:</b> [Alternative Flow 1 – User reverts fields]	1a. In step 1 of the normal flow, the user clicks a discard changes button after modifying fields 1. System displays refreshes the system settings form populated with previous values		
<b>Alternative Flows:</b> [Alternative Flow 2 – User declines changes]	4a. In step 4 of the normal flow, if the user declines the changes 1. System displays system settings form with values the user had altered		
<b>Exceptions:</b>	5a. In step 5 of the normal flow, if a network or system error prevents updating the records 3. System displays an error message 4. System returns to the form with the values the user had amended		
<b>Includes:</b>			
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Admin will have the ability to customize the fields used by the system as well as default permission levels.  The system settings form will use selection and menu inputs rather than text inputs. Any invalid or forbidden choices will be deactivated by the system so that users can only select valid settings.  The UX settings will vary depending on the hardware version of the app: a phone app will have some different options than a desktop app, etc.		

<b>Notes and Issues:</b>	<ol style="list-style-type: none"><li>1. Development team will need to determine the granularity of these features—what level of choice is necessary to assure accessibility to the widest possible range of users?</li><li>2. Designing and clearly labeling access to accessibility features may require some amount of legal research as well as best practice research.</li></ol>
--------------------------	---

## Change Communications Settings

<b>Use Case ID:</b>	UC-USER.01.10		
<b>Use Case Name:</b>	Change Communications Settings		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-09	<b>Last Revision Date:</b>	2023-9-11
<b>Actors:</b>	User		
<b>Description:</b>	The user updates communication preferences		
<b>Trigger:</b>	User navigates to or is directed to the communication preferences form		
<b>Preconditions:</b>	1. User account exists 2. User is logged in 3. User account is active 4. User has system permissions to modify communication preferences		
<b>Postconditions:</b>	<b>Success</b> 1. System contains an updated record of user's communication preferences 2. System no longer uses previous information for communication—email preferences, text notifications, etc. 3. System displays the communication preferences form <b>Failures</b> 4. System displays an error message		
<b>Normal Flow:</b>	1. User updates desired fields in the communication preferences form 2. User selects "save changes" 3. System prompts user to confirm changes 4. User confirms changes 5. System stores new field values, associating them with account functions 6. System displays a confirmation pop up 7. System returns to the communication preferences form		
<b>Alternative Flows:</b> [Alternative Flow 1 – User reverts fields]	1a. In step 1 of the normal flow, the user clicks a revert button after modifying fields 1. System displays refreshes the communication preferences form populated with previous values		
<b>Alternative Flows:</b> [Alternative Flow 2 – User declines changes]	4a. In step 4 of the normal flow, if the user declines the changes 1. System displays communications preferences form with values the user had amended		
<b>Exceptions:</b>	5a. In step 5 of the normal flow, if a network or system error prevents updating the records 1. System displays an error message 2. System returns to the form with the values the user had amended		
<b>Includes:</b>	Reset password		
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>	<b>Assumptions:</b> Different user roles will have different levels of permission for changing communication settings, e.g., an employee unable to change some notification settings.  Admin will be able to customize default settings and other options.  The communications settings form will use selection and menu inputs rather than text inputs. Any invalid or forbidden choices will be deactivated by the system so that users can only select valid settings.		

<b>Notes and Issues:</b>	<ol style="list-style-type: none"><li>1. Determining the communication options available to users may include legal research to meet FCC and other requirements</li><li>2. Can this be merged into another use case? Like updating an account?</li></ol>
--------------------------	--

## Modify Payment Information

<b>Use Case ID:</b>	UC-22.01.11		
<b>Use Case Name:</b>	Modify Payment Info		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-09	<b>Last Revision Date:</b>	2023-09-13
<b>Actors:</b>	User		
<b>Description:</b>	The user updates payment information for the account		
<b>Trigger:</b>	User navigates to or is directed to the account update form		
<b>Preconditions:</b>	1. User account exists 2. User is logged in 3. User account is active		
<b>Postconditions:</b>	Success 1. System contains an updated record of user's payment information 2. System no longer uses previous information for business processes 3. System displays the payment options page Failures 4. System displays an error message		
<b>Normal Flow:</b>	1. User updates desired fields in the payment information form 2. User selects "save changes" 3. System prompts user to confirm changes 4. User confirms changes 5. System validates financial information 6. System stores new field values, associating them with account functions 7. System displays a confirmation message 8. System returns to the payment information form		
<b>Alternative Flows:</b> [Alternative Flow 1 – User declines changes]	4a. In step 4 of the normal flow, if the user declines the changes 1. System displays payment information form with values the user had amended		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, if the customer enters an invalid value for a field 1. System highlights the invalid field and deactivates the "save changes" button until a valid value is entered  5b. In step 5 of normal flow, the system detects invalid payment data 1. System displays an error message 2. System displays the amended state of the form with invalid fields highlighted  6a. In step 6 of the normal flow, if a network or system error prevents updating the records 1. System displays an error message 2. System displays the form with the values the user had amended		
<b>Includes:</b>			
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Best practice includes two steps of validation: basic formal validation of fields and a following validation of financial info (routing numbers etc.)		
<b>Notes and Issues:</b>	1. Transmission and storage of payment information should be encrypted and must follow necessary laws and best practices		

- |  |  |
|--|--|
|  | <ol style="list-style-type: none"><li>2. Should the application require a password check when updating payment information?</li><li>3. What processes need to be created to handle automatic events like expiration of credit cards, etc.?</li></ol> |
|--|--|

## Main Menu Access

<b>Use Case ID:</b>	UC-USER.01.12		
<b>Use Case Name:</b>	User Main Menu		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-10	<b>Last Revision Date:</b>	2023-09-13
<b>Actors:</b>	User		
<b>Description:</b>	The user views the main menu and navigates to application features		
<b>Trigger:</b>	User logs into application or navigates to main menu		
<b>Preconditions:</b>	1. User is logged into the application. 2. User has a preexisting account. 3. User account is currently activated. 4. User is viewing main menu 5. Local client has network access to the application.		
<b>Postconditions:</b>	Success 1. User is viewing desired page or form of the application Failure 2. System displays an error message		
<b>Normal Flow:</b>	1. User views a GUI listing system features 2. System updates GUI elements to reflect current status and unviewed changes (new message icon, etc.) 3. User activates link to desired application function 4. System displays corresponding page, form, etc.		
<b>Alternative Flows:</b> [Alternative Flow 1 – User Closes Application]	1a. In step 1 of the normal flow, if the user closes the application 1. System logs closing of the local application instance		
<b>Exceptions:</b>	2a. In step 2 of the normal flow, network or system errors result in a failure to update display 1. System displays default main menu  4a. In step 4 of the normal flow, network or system errors result in a failure to access the desired feature 1. System displays an error message		
<b>Includes:</b>	Navigation Notification System		
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>Main Menu (as a page) and the navigation bar are distinct from one another. The main menu serves as a landing page when logging into the application, while the navigation bar is persistent across many features of the application.</p> <p>The main menu and navigation bar contain some redundant information, but some application services might only be available from one or the other (e.g. "User Preferences" is accessed from the navigation bar, but not from the main menu landing page.)</p>		
<b>Notes and Issues:</b>	1. Should the main menu be the default landing place? Could the default landing place depend on the user's role(s)? 2. Should admin or users be able to adjust the default features on the main menu (e.g. decide if a to-do list is displayed on load)		

	3. The main menu is a primary design element of the application, and some design decisions will likely affect the specific flow involved in use cases.
--	--

## Navigation Bar

<b>Use Case ID:</b>	UC-USER.01.13		
<b>Use Case Name:</b>	Navigate Application		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-10	<b>Last Revision Date:</b>	2023-09-13
<b>Actors:</b>	User		
<b>Description:</b>	The user views a menu of site features and navigates to them		
<b>Trigger:</b>	User views a page or form that contains the navigation bar		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User is logged into the application.</li> <li>2. User has a preexisting account.</li> <li>3. User account is currently activated.</li> <li>4. User is viewing a page or form that includes the navigation bar.</li> <li>5. Local client has network access to the application.</li> </ol>		
<b>Postconditions:</b>	<p>Success</p> <ol style="list-style-type: none"> <li>1. User is viewing desired page or form of the application</li> </ol> <p>Failure</p> <ol style="list-style-type: none"> <li>2. System displays an error message</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. System periodically refreshes graphics relating to notifications (e.g. messages in inbox)</li> <li>2. User expands elements of the navigation bar by clicking, tapping, etc.</li> <li>3. System displays any necessary drop-down menus or animations</li> <li>4. User activates link to desired application function</li> <li>5. System displays corresponding page, form, etc.</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – User Closes Application]	<p>4a. In step 4 of the normal flow, if the user removes focus from the activated element</p> <ol style="list-style-type: none"> <li>1. System collapses menus and otherwise returns to default appearance</li> </ol>		
<b>Exceptions:</b>	<p>1a. In step 1 of the normal flow, network or system errors result in a failure to update display</p> <ol style="list-style-type: none"> <li>1. System continues to display the current version of the navigation bar</li> </ol> <p>5a. In step 5 of the normal flow, network or system errors result in a failure to access the desired feature</p> <ol style="list-style-type: none"> <li>1. System displays an error message</li> </ol>		
<b>Includes:</b>	<p>Notification System</p> <p>**Many other features can be called via a link/button in the navigation bar: main menu, logout, etc.</p>		
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	<p>The navigation bar will be a primary mode of navigating throughout the application.</p> <p>Most pages/forms within the app will contain a version of the navigation bar.</p> <p>The structure of the navigation bar may vary depending on format (phone, desktop, etc.).</p>		

	The structure of the navigation bar may vary depending on the current page being displayed by system, e.g., a simplified navigation bar on some pages.
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Are there more practical navigation systems than a web-style nav bar?</li> <li>2. What consistent criteria can be used to determine if a particular function on the app does not feature the navigation bar?</li> <li>3. The navigation bar is a primary design element of the application, and some design decisions will likely affect the specific flow involved in use cases.</li> </ol>

## Message Log

<b>Use Case ID:</b>	UC-USER.01.14		
<b>Use Case Name:</b>	View Messages		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-07	<b>Last Revision Date:</b>	2023-09-13
<b>Actors:</b>	User		
<b>Description:</b>	View messages within the application		
<b>Trigger:</b>	User navigates to the application's messages page		
<b>Preconditions:</b>	1. User is logged in. 2. User account is currently activated. 3. User is viewing the messages page 4. Local client has network access to the application.		
<b>Postconditions:</b>	<b>Successes</b> 1. System displays the body of an individual message 2. System displays a list of messages <b>Failures</b> 3. The system displays a message load error message 4. The system displays an empty list of messages and error message		
<b>Normal Flow:</b>	1. User views a list of notifications/messages 2. User selects an item from the list 3. System displays single message 4. System logs a record of access to the message 5. User reads and closes single message 6. If the item contained a new/unread flag, system removes flag 7. System displays refreshed list of message		
<b>Alternative Flows:</b> [NA]			
<b>Exceptions:</b>	1a. In step 1 of the normal flow, if system cannot load the list of messages 1. System displays an error message and empty list  3a. In step 3 of the normal flow, if the system cannot load the body of an individual message 1. System displays an error message 2. User closes error message 3. System displays list of messages  7a. In step 7 of the normal flow, if system cannot load an updated list of messages 1. System displays the last loaded list of messages		
<b>Includes:</b>			
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	As described, this feature assumes a basic log of messages, rather than a more elaborate message management system that includes common features such as sorting, deleting, etc.  A single message system will handle all categories of messages: system notifications, direct messages, memos, etc.		

<b>Notes and Issues:</b>	<ol style="list-style-type: none"><li>1. The amount of functionality built into the message system will need to be revised as the project develops and more use cases arise. Flagging, sorting, searching, and deleting messages are likely necessary features.</li><li>2. What kind of users should have the power to send out messages of any sort? Will all employees need this functionality, or is it solely a managerial/admin use case?</li><li>3. If the system is implemented as a simple log, what are the criteria for archiving or deleting old messages? What system can be used to keep the list manageable?</li></ol>
--------------------------	--

## Notification System

<b>Use Case ID:</b>	UC-USER.01.16		
<b>Use Case Name:</b>	Receive Notifications		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-10	<b>Last Revision Date:</b>	2023-09-13
<b>Actors:</b>	User		
<b>Description:</b>	The user receives a notification from the system or a sending party		
<b>Trigger:</b>	A new message is sent to the user		
<b>Preconditions:</b>	1. User is currently logged into local client 2. User account is currently activated. 3. Local client has network access to the application.		
<b>Postconditions:</b>	Success 1. User views new message 2. System logs that message has been viewed 3. Graphical flags and other system displays accurately represent the current state of notifications, e.g. "3 unread" Failure 4. Graphical flags and other system displays represent the previous state of notifications, e.g. "4 unread"		
<b>Normal Flow:</b>	1. System displays a new message dialog box 2. System updates visual flag of the "notifications" item in the navigation bar 3. User activates the message dialog 4. System displays the new message		
<b>Alternative Flows:</b> [Alternative Flow 1 – Use Navigation Bar]	3a. In step 3 of the normal flow, if the user activates the navigation bar rather than the dialog box 1. System calls the Message Log feature, where new messages are listed at the top of the log (unless otherwise sorted)		
<b>Alternative Flows:</b> [Alternative Flow 1 – Dismiss Notification]	3b. In step 3 of the normal flow, if the user dismisses the dialog box 1. System closes dialog box		
<b>Exceptions:</b>	4a. In step 4 of the normal flow, a network or system error prevents retrieving the message 1. System displays the message log page 2. System displays an error message		
<b>Includes:</b>	Message Log		
<b>Frequency of Use:</b>	[On Demand.]		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	All types of messages(schedule, technical, user-sent, etc.) can and will be handled by the same notification system.  A simple pop-up dialog box combined with a visual flag in the navigation bar is the best choice for handling a notification.		
<b>Notes and Issues:</b>	1. It may be difficult or poor UX practice to handle every kind of communication through the same notification system, depending on the use cases associated with specific roles 2. Should we reinforce the notification with sounds, vibration, etc.? 3. Should notification settings be adjusted under the system settings or under the communications settings forms (or both)?		

- |  |  |
|--|--|
|  | 4. We should consider the best UX practices for notifications, should the dialog box auto-disappear? Should it be disabled in certain contexts, etc. |
|--|--|

## Help System

<b>Use Case ID:</b>	UC-USER.01.17		
<b>Use Case Name:</b>	Request Help		
<b>Created By:</b>	Michael Springer	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-07	<b>Last Revision Date:</b>	2023-09-13
<b>Actors:</b>	User		
<b>Description:</b>	User troubleshoots problems via system and external resources (FAQ, support request, etc.)		
<b>Trigger:</b>	User opens the application's "help" page or activates a help link from the login screen		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>User must be viewing a client instance of the application.</li> <li>User has a preexisting account (internal help page)</li> <li>User account is currently activated (internal help page)</li> <li>Local client has network access to the application.</li> </ol>		
<b>Postconditions:</b>	<p>Successes</p> <ol style="list-style-type: none"> <li>The user views a FAQ or other resource explaining a solution to the problem</li> <li>Customer Support receives an automated ticket regarding the user's request</li> <li>Customer Support receives a manual message regarding the user's request</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>User selects help menu page from navigation menu</li> <li>System displays page tips and/or troubleshooting links to frequent issues and a link to a customer service contact form</li> <li>User activates link to issue</li> <li>User views an internal tip or external resource, e.g., FAQ hosted on organization's web page</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Help Request Message]	<p>3a. In step 3 of the normal flow, the user activates the link to a service request form</p> <ol style="list-style-type: none"> <li>System displays a confirmation screen</li> <li>User accepts</li> <li>System redirects user to the form with submission locked</li> <li>User completes required fields</li> <li>When all fields are validated, system unlocks submission</li> <li>User submits</li> <li>System logs the help request event</li> <li>System sends a message to customer service</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Request from Login]	<p>1a. A user selects the help item from the login page</p> <ol style="list-style-type: none"> <li>System displays a static site with tips and troubleshooting links as well as a link to a customer service form</li> <li>User follows link to support article</li> <li>System displays support article or redirects to an external page</li> </ol> <p>1b. A user selects the help item from the login page</p> <ol style="list-style-type: none"> <li>System displays a static site with tips and troubleshooting links as well as a link to a customer service form</li> <li>User follows link to help request</li> <li>System displays a modified version of the form used in step Alternative Flow 3a.3 (contains additional required fields)</li> <li>User completes required fields</li> <li>When all fields are validated, system unlocks submission</li> <li>User submits</li> </ol>		

	7. System logs the help request event for customer service
<b>Exceptions:</b>	<p>1a. In step 3a.7 of the alternative flow, if the system cannot complete logging the support ticket</p> <p>1. System displays an error message with customer service contact information</p>
<b>Includes:</b>	
<b>Frequency of Use:</b>	[On Demand.]
<b>Special Requirements:</b>	The level of support available necessarily depends on the infrastructure of the organization using the application. Features will need to be enabled and disabled by Admin to accommodate the needs of an org.
<b>Assumptions:</b>	<p>Account registration and login will be necessary to use most features of the application.</p> <p>The same overall login process should be used by all types of users, with user experiences diverging only after login.</p>
<b>Notes and Issues:</b>	

## GPS Navigation

<b>Use Case ID:</b>	UC-22.01		
<b>Use Case Name:</b>	GPS Navigation		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Everett DeVaux
<b>Date Created:</b>	9/5/2023	<b>Last Revision Date:</b>	09/17/2023
<b>Actors:</b>	<ul style="list-style-type: none"> <li>• Student</li> </ul>		
<b>Description:</b>	The customer will be able to see where the driver and them are always.		
<b>Trigger:</b>	The customer will know their location, where they are going, and when they will arrive at their destination.		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>• Customer must have an account.</li> <li>• Customer and Driver have cellular signal or stable Wi-Fi signal.</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>• Customer and Driver can see where they are and destination arrival.</li> </ul>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Customer utilizes driver reservation.</li> <li>2. Customer selects [GPS]</li> <li>3. The system will gather the location of the customer and driver.</li> <li>4. The system will show the location on the screen.</li> <li>5. The system will show the arrival time.</li> <li>6. The customer can close the GPS and able to return to other options in the app.</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<p>2a. In step 2 of the normal flow if the customer does not have an internet connection.</p> <ol style="list-style-type: none"> <li>1. The system will prompt the customer to connect to a Wi-Fi or data plan.</li> <li>2. The system will navigate the customer back to the main menu.</li> <li>3. Use Case resumes in step 6.</li> </ol>		
<b>Exceptions:</b>	<p>3a. In step 2 of the normal flow if the customer does not have internet but the connection is restored.</p> <ol style="list-style-type: none"> <li>1. The GPS will be accessible.</li> <li>2. Customer can check their GPS information.</li> </ol>		
<b>Includes:</b>	<ul style="list-style-type: none"> <li>• Customer database</li> <li>• GPS database</li> <li>• Driver database</li> </ul>		
<b>Frequency of Use:</b>	On-demand		
<b>Special Requirements:</b>	<ul style="list-style-type: none"> <li>• Needs to be able to connect to GPS system to find customers.</li> <li>• Access to Customer database.</li> </ul>		
<b>Assumptions:</b>	<ul style="list-style-type: none"> <li>• The customer has an internet or data plan.</li> </ul>		
<b>Notes and Issues:</b>	<ul style="list-style-type: none"> <li>• How do we implement GPS for the customer?</li> <li>• How do we get their location into our GPS?</li> </ul>		

## Passes for Qualifying Individuals

<b>Use Case ID:</b>	UC-19.4		
<b>Use Case Name:</b>	Passes for Qualifying Individuals		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Everett DeVaux
<b>Date Created:</b>	09/06/2023	<b>Last Revision Date:</b>	09/17/2023
<b>Actors:</b>	<ul style="list-style-type: none"> <li>• Student</li> </ul>		
<b>Description:</b>	Be able to get an all-day pass for a cheaper amount than making multiple rides separately.		
<b>Trigger:</b>	Students can make multiple stops in a day with the ease of not having to make multiple payments at a high rate.		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>• Customer has an account.</li> <li>• Customer has multiple stops to make.</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>• Customer receives a day pass.</li> <li>• Customer can make multiple stops on a day pass.</li> </ul>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Customer logs into account</li> <li>2. The customer goes to [Schedule Ride]</li> <li>3. The system will have an option for [Day Pass]</li> <li>4. Once chosen the system will prompt a message explaining how a day pass works.</li> <li>5. Customer accepts.</li> <li>6. Customer makes payment.</li> <li>7. Customers can get a ride at any point in the day.</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]			
<b>Exceptions:</b>	<p>6a. In step 6 of the normal flow if the customer enters an invalid payment info.</p> <ol style="list-style-type: none"> <li>1. Transactions are disapproved.</li> <li>2. Message to customer to re-enter information.</li> <li>3. Customer enters correct information.</li> <li>4. Use Case resumes on step 7 of normal flow.</li> </ol>		
<b>Includes:</b>	<ul style="list-style-type: none"> <li>• Customer database</li> </ul>		
<b>Frequency of Use:</b>	On-demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## System Access – Client

# Employee – Driver

## Vehicle Checkout

<b>Use Case ID:</b>	DRV.01.01		
<b>Use Case Name:</b>	Vehicle Check Out		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-01	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	A driver checks out a vehicle for use		
<b>Trigger:</b>	Driver selects to check out a vehicle		
<b>Preconditions:</b>	1. Vehicle available for checkout 2. Driver logged into system		
<b>Postconditions:</b>	1. Vehicle assigned to driver 2. Vehicle marked as in use		
<b>Normal Flow:</b>	1. Available vehicles are displayed 2. Driver selects vehicle 3. Vehicle safety check list displayed 4. Driver completes checklist with no concerns 5. Driver confirms check out 6. System displays checkout success message		
<b>Alternative Flows:</b>	2a-5a. In step 2, 3, 4, or 5 of normal flow, Driver cancels request 1. Use case resumes on step 1 of normal flow 4b. In step 4 of normal flow, Driver finds issues with the vehicle requiring maintenance. 1. System prompts Driver that vehicle is no longer available 2. Use case resumes on step 1 of normal flow		
<b>Exceptions:</b>	2a. In step 2 of normal flow, the driver selects a vehicle for checkout that becomes unavailable. 1. System displays message that vehicle is no longer available 2. Use case resumes on step 1 of normal flow 4a. In step 4 of normal flow, the driver doesn't complete the checklist. 1. System prompts Driver that safety checklist needs to be completed 2. Use case resumes on step 4 of normal flow.		
<b>Includes:</b>	Checkout Checklist		
<b>Frequency of Use:</b>	Daily		
<b>Special Requirements:</b>	1. Vehicle database		
<b>Assumptions:</b>	1. Driver knows the vehicle they need to check out. 2. Driver trained to identify issues with a vehicle. 3. Driver knows how to navigate to available vehicles		
<b>Notes and Issues:</b>			

## Vehicle Checkout/Return Checklist

<b>Use Case ID:</b>	DRV.01.02		
<b>Use Case Name:</b>	Vehicle Checkout/Return Checklist		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-01	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	A Driver completes a checklist to ensure a vehicle is safe for operation.		
<b>Trigger:</b>	Driver is checking out a vehicle for operation.		
<b>Preconditions:</b>	1a. A driver is checking out a vehicle. 1b. A driver is in the process of returning a vehicle from operation.		
<b>Postconditions:</b>	1. No issues found, vehicle checkout can resume 2. Issues found: 1. Checklist report sent to maintenance 2. Vehicle flagged as unavailable and needing maintenance		
<b>Normal Flow:</b>	1. Driver completing vehicle checkout or return. 2. Checklist displayed to Driver 3. Driver marks items as checked 4. Driver completes safety checklist		
<b>Alternative Flows:</b>	3a. In step 3 of normal flow, the driver finds an issue with the vehicle while completing the checklist. 1. Checklist item flagged as needing attention 2. Driver documents issues by filing a maintenance report. 3. Driver submits the checklist. 4. Vehicle removed from service and flagged for repair		
<b>Exceptions:</b>	2a – 4a. In step 2, 3, or 4 of normal flow, Driver exits the checklist before completion. 1. Driver returned to the vehicle checkout. 4b. In step 3 of normal flow, the driver doesn't check off item(s) and attempts to submit a checklist in step 4 of normal flow. 1. Driver prompted to complete checklist item(s) 2. Use case resumes on step 2 of normal flow.		
<b>Includes:</b>	1. Vehicle Check Out 2. Maintenance Reporting		
<b>Frequency of Use:</b>	Twice Daily/On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	1. Driver trained to identify issues with a vehicle.		
<b>Notes and Issues:</b>	1. What should be on the checklist?		

## View Routes

<b>Use Case ID:</b>	DRV.01.03		
<b>Use Case Name:</b>	Select Routes		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-01	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	Allow the driver to view assigned routes and details such as distance, destination, and estimated time of arrival.		
<b>Trigger:</b>	A route is assigned to the driver.		
<b>Preconditions:</b>	1. Preplanned route created by the manager 2. Added on as needed by dispatch or the manager 3. Route created by the driver		
<b>Postconditions:</b>	1. Driver can see destination(s), distance, and estimated time of arrival.		
<b>Normal Flow:</b>	1. Driver opens the route feature and is given a list of routes. 2. Driver selects an assigned route 3. Driver starts the route and follows directions given 4. Driver ends the route when finished.		
<b>Alternative Flows:</b>	2a. The driver creates a route. 1. Driver enters the destination(s) for the route 2. Flow continues at step 3 of normal flow. 2b. Driver modifies assigned route. 1. Driver presented with the destinations of the assigned route 2. Driver modifies any destinations as needed. 3. Continue at step 3 of normal flow.		
<b>Exceptions:</b>	2a, 2b. Driver doesn't have permissions to enter or modify the route. 1. Error prompt explaining that this feature is not accessible 2. Continue at step 1 of normal flow.		
<b>Includes:</b>	1. List of routes		
<b>Frequency of Use:</b>	On Demand/Frequently		
<b>Special Requirements:</b>	1. Mapping system		
<b>Assumptions:</b>	1. Driver has access permissions to create a route		
<b>Notes and Issues:</b>	1. What mapping system will be used?		

## Start Route

<b>Use Case ID:</b>	DRV.01.04		
<b>Use Case Name:</b>	Start Route		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-06	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	Confirm in the system that a route has started.		
<b>Trigger:</b>	Driver selects and begins a route.		
<b>Preconditions:</b>	1. Route is assigned to a driver. 2. Driver has a vehicle assigned to them and checked out. 3. Driver is in vehicle, ready to start their route		
<b>Postconditions:</b>	1. The driver and vehicle are assigned to the route. 2. Dispatch is notified that the driver has initiated the route.		
<b>Normal Flow:</b>	1. Driver selects their assigned route. 2. Driver selects that they are beginning their route. 3. Route is shown as currently active with the assigned driver.		
<b>Alternative Flows:</b>	1a. Driver not assigned to route selected. 1. The driver is notified that this is not their assigned route. 2a. The driver confirms that they understand and that they are assigning themselves to this route. 3. Continue to step 3 of normal flow.		
<b>Exceptions:</b>	1a. Route currently active with another driver and vehicle. 1. The driver is notified that the route is already currently active. 2. Return to step 1 of the normal flow.		
<b>Includes:</b>	1. Route list 2. Vehicle Checkout		
<b>Frequency of Use:</b>	Daily/On Demand		
<b>Special Requirements:</b>	1. GPS		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1. What should happen if the route is already active with another driver? 2. What should happen if the driver is not preassigned to the route?		

## Vehicle Refueling Log

<b>Use Case ID:</b>	DRV.01.05		
<b>Use Case Name:</b>	Vehicle Refueling Log		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-01	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	A feature that allows a driver to log when a vehicle has been refueled during their route.		
<b>Trigger:</b>	Driver needs to refuel the vehicle.		
<b>Preconditions:</b>	1. Vehicle requires fuel. 2. Approved refueling station nearby		
<b>Postconditions:</b>	1. Mileage between refueling logged 2. Amount of fuel added logged 3. Payment receipt logged 4. Fueling report sent to manager		
<b>Normal Flow:</b>	1. Driver selects the option to start refueling. 2. Vehicle information displayed (fuel type) 3. Mileage logged 4. Driver initiates refueling 5. Price per gallon, total price, and total amount in gallons logged. 6. Driver ends refueling		
<b>Alternative Flows:</b>	1a, 2a. Driver cancels refueling. 1. Exit refueling mode		
<b>Exceptions:</b>	3a, 5a. Driver doesn't log data. 1. Error prompt reminding driver to enter data. 2. Return to step 5 of normal flow.		
<b>Includes:</b>	1. Vehicle Information		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	1. Vehicle sensors for fuel level and mileage. 2. Driver issued a credit card for refueling. 3. Available Gas stations		
<b>Assumptions:</b>	1. Driver has the money to pay for gas		
<b>Notes and Issues:</b>	1. Should a report be generated that shows information about time/distance between refueling?		

## End Route

<b>Use Case ID:</b>	DRV.01.06		
<b>Use Case Name:</b>	End Route		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-06	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	Driver ends their route, unassigning themselves from it.		
<b>Trigger:</b>	1. Driver reaches the end of their route. 2. Driver chooses to end the route manually.		
<b>Preconditions:</b>	1. Driver is currently assigned and active on the route.		
<b>Postconditions:</b>	1. The driver and vehicle are unassigned from the route. 2. Route flagged as inactive. 3. The vehicle is flagged as needing returned. 4. Dispatch notified that route has ended.		
<b>Normal Flow:</b>	1. Driver selects their route. 2. Driver selects the option to end the route. 3. Prompt asks the driver to confirm action. 4. Action confirmed. 5. The driver is notified that they have ended the route.		
<b>Alternative Flows:</b>	3a. Driver does not confirm. 1. Exits normal flow of this feature.		
<b>Exceptions:</b>	2a. Passengers still on vehicle 1. Driver prompted to finish current rides for passengers 2. Exit normal flow and return driver to active route  2b. Driver has not reached the final destination of the route. 1. The driver is presented with a notification that the route is still in progress. 2. The driver can choose to end the route early or continue.		
<b>Includes:</b>	1. Active routes 2. Vehicle Checkout Information 3. Passenger List		
<b>Frequency of Use:</b>	After Driver reaches destination		
<b>Special Requirements:</b>	1. GPS		
<b>Assumptions:</b>	1. Driver is ending the route at a predetermined time or destination.		
<b>Notes and Issues:</b>			

## Vehicle Return

<b>Use Case ID:</b>	DRV.01.07		
<b>Use Case Name:</b>	Vehicle Return		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-01	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	Driver returns a vehicle after use.		
<b>Trigger:</b>	Driver selects the option to return a vehicle.		
<b>Preconditions:</b>	1. Vehicle is currently checked out to the driver attempting to return it.		
<b>Postconditions:</b>	1. Vehicle passes return checklist: Vehicle returned to service and is available for checkout. 2. Vehicle fails return checklist: Vehicle is flagged for needing maintenance and remains unavailable for checkout.		
<b>Normal Flow:</b>	1. Driver views list of vehicles checked out under their account. 2. Driver selects vehicle to return. 3. Return checklist displayed to Driver. 4. Driver completes the checklist and submits it. 5. Prompt verifying vehicle return shown. 6. Prompt asking if Driver has another vehicle to return. 7. Driver selects no other vehicle to return.		
<b>Alternative Flows:</b>	1a. If there is no vehicle to return, the driver is shown "No vehicles to return" and given a link to the vehicle checkout feature. 1b. Driver cancels return process and exits feature. 4a, 5a. Driver cancels 1. Return Driver to step 1 of normal flow. 6. If the Driver selects that they have another vehicle to return, repeat steps from normal flow step 1.		
<b>Exceptions:</b>	4a. Driver doesn't fully complete the checklist. 1. Driver prompted that the checklist is not complete. 2. Return to step 4 of normal flow. 7a. Driver selects that they have another vehicle to return. 1. Return to step 1 of normal flow.		
<b>Includes:</b>	1. Vehicle Checkout/Return Checklist		
<b>Frequency of Use:</b>	After the driver finishes their ride		
<b>Special Requirements:</b>	1. Vehicle Database		
<b>Assumptions:</b>	1. Driver trained to identify issues with a vehicle. 2. Driver made it to vehicle drop off safely		
<b>Notes and Issues:</b>	Where would the vehicles be dropped off at?		

## View Available Vehicles

<b>Use Case ID:</b>	DRV.01.14		
<b>Use Case Name:</b>	View Available Vehicles		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-09	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	The driver can view a list of vehicles that are available for use.		
<b>Trigger:</b>	The driver opts to view the available vehicle list.		
<b>Preconditions:</b>	1. There is a list of available vehicles. 2. The driver is assigned a route or passengers so they know which vehicle they will need.		
<b>Postconditions:</b>	1. The driver sees a list of vehicles that are available.		
<b>Normal Flow:</b>	1. Driver selects the option to view available vehicles. 2. Available vehicles and their information is displayed. 3. The driver can select a vehicle and initiate vehicle checkout.		
<b>Alternative Flows:</b>	3. The driver can exit without initiating vehicle checkout.		
<b>Exceptions:</b>			
<b>Includes:</b>	1. Vehicle Checkout 2. Available Vehicle List		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	1. Database of vehicles		
<b>Assumptions:</b>	1. The drivers assign themselves vehicles, not a manager or dispatch.		
<b>Notes and Issues:</b>			

## View Passenger List

<b>Use Case ID:</b>	DRV.02.01		
<b>Use Case Name:</b>	View Passenger List		
<b>Created By:</b>	James Williams	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	2023-09-04	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Driver		
<b>Description:</b>	Drivers can view a list of passengers to be picked up, any assistance they may require, assistive devices they use, and their destinations.		
<b>Trigger:</b>	1. When the driver selects a destination point on their route. 2. The driver opts to view upcoming passengers.		
<b>Preconditions:</b>	1. Passengers are pre registered for rides. 2. Passengers are assigned to a driver's route.		
<b>Postconditions:</b>	1. The driver can view information about passengers on their route.		
<b>Normal Flow:</b>	1. Driver selects their assigned passengers list 2. Passenger list populates showing passenger name, pickup spot, assistive device used by passenger, and any additional requirements. 3. Driver exits passenger list and returns to their route view		
<b>Alternative Flows:</b> Driver reaches stop on route.	1a. Driver reaches stop on route. 2a. List of expected passengers and their information is shown to the driver.		
<b>Alternative Flows</b> Driver reaches passenger's destination	1b. Driver reaches the passenger's destination. 2b. Departing passenger information displayed to the driver.		
<b>Exceptions:</b>			
<b>Includes:</b>	1. Route information		
<b>Frequency of Use:</b>	Every ride		
<b>Special Requirements:</b>	1. GPS enabled device.		
<b>Assumptions:</b>	1. Passengers are assigned to the appropriate vehicle and driver that can meet their physical requirements. Like a van with a wheelchair ramp for example.		
<b>Notes and Issues:</b>			

# Employee – Dispatch

## View Vehicle Schedules

<b>Use Case ID:</b>	DISP.1.1		
<b>Use Case Name:</b>	View Vehicle Schedules		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/23
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch views a list of all vehicles and their schedules		
<b>Trigger:</b>	<ul style="list-style-type: none"> <li>1. Dispatch needs to assign, update or deactivate/reactivate Route Assignments</li> <li>2. Dispatch needs to assign, update or deactivate/reactivate Vehicle Assignments to Ride Services</li> <li>3. Dispatch needs to assign, update or deactivate/reactivate Vehicle Assignments to Charters</li> <li>4. Dispatch needs to assign, update or deactivate/reactivate Vehicle Maintenance Instances</li> <li>5. Dispatch needs to assign, update or deactivate/reactivate Other vehicle Unavailability blocks</li> </ul>		
<b>Preconditions:</b>	1. Logged in employee has dispatch role		
<b>Postconditions:</b>	1. Dispatch sees a list of all vehicles and their schedules		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Dispatch rep presses a button from the main screen, the view individual Route screen, the view Charter Request screen, the view Ride Service screen, or the view Maintenance Instances screen</li> <li>2. Dispatch rep is presented with a list of Vehicles and their schedules</li> </ul>		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	<p>1a. In step 1 of the normal flow, The databases containing the data may be unavailable</p> <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed and the user is brought back to the prior screen</li> </ul>		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Create Vehicle Charter Assignment

<b>Use Case ID:</b>	DISP.1.2		
<b>Use Case Name:</b>	Create Vehicle Charter Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch assigns a vehicle to a Charter Request		
<b>Trigger:</b>	1. Dispatch needs to assign a vehicle to a Charter Request		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. A charter has been requested		
<b>Postconditions:</b>	1. Vehicle is assigned to the charter for a set time window.		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Route window 2. Dispatch is brought to the View Vehicle Schedules Screen 3. Dispatch may use a set of filters to narrow down vehicle choice, including the time frame of the route instance. 4. Dispatch selects one or more vehicles that meet the requirements, then presses an action button. 5. The vehicle(s) are then attached to the charter request.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	This will typically flow into UC-DISP.2.2 Create Driver Charter Assignment, but not always.		

## Create Vehicle Maintenance Assignment

<b>Use Case ID:</b>	DISP.1.2		
<b>Use Case Name:</b>	Create Vehicle Maintenance Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch assigns a vehicle to a maintenance instance		
<b>Trigger:</b>	1. Dispatch needs to send a vehicle in for maintenance, and needs to make that shown		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. A maintenance order needs to be handled for a vehicle		
<b>Postconditions:</b>	1. Vehicle is assigned to maintenance for a set time window.		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Maintenance window 2. Dispatch is brought to the View Vehicle Schedules Screen 3. Dispatch may use a set of filters to narrow down vehicle choice, including the time frame of the route instance. 4. Dispatch selects one or more vehicles that meet the requirements, then presses an action button. 5. Dispatch is then presented with a question of what timeframe the maintenance needs to have the vehicle, and which days of the week. 6. Once the dispatch rep has selected those options, they then press one final action button to submit the assignment. 7. The vehicle(s) are then attached to the maintenance.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-6 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View maintenance window		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Create Vehicle Other Unavailability Assignment

<b>Use Case ID:</b>	DISP.1.2		
<b>Use Case Name:</b>	Create Vehicle other unavailability Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch marks a vehicle as unavailable		
<b>Trigger:</b>	1. Dispatch needs to mark a vehicle as unavailable for an abnormal reason		
<b>Preconditions:</b>	1. Logged in employee has dispatch role		
<b>Postconditions:</b>	1. Vehicle is marked unavailable for the set time window.		
<b>Normal Flow:</b>	1. Dispatch presses a button from the view vehicle schedules screen associated with a specific vehicle. 2. Dispatch is then presented with a question of what timeframe the unavailability needs to have the vehicle, and which days of the week 3. Dispatch enters a description in a text field to describe the reason the vehicle is unavailable. 4. Once the dispatch rep has selected those options, they then press one final action button to submit the assignment. 5. The vehicle(s) are then attached to the other unavailability.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View other unavailability window		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Create Vehicle Ride Service Assignment

<b>Use Case ID:</b>	DISP.1.2		
<b>Use Case Name:</b>	Create Vehicle Ride Service Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch assigns a vehicle to a Ride Service		
<b>Trigger:</b>	1. Dispatch needs to assign a vehicle to a Ride Service		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. A ride service has been created		
<b>Postconditions:</b>	1. Vehicle is assigned to the Ride Service for a set time window.		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Ride Service window 2. Dispatch is brought to the View Vehicle Schedules Screen 3. Dispatch may use a set of filters to narrow down vehicle choice, including the time frame of the route instance. 4. Dispatch selects one or more vehicles that meet the requirements, then presses an action button. 5. Dispatch is then presented with a question of what timeframe the ride service needs to have the vehicle, and which days of the week. 6. Once the dispatch rep has selected those options, they then press one final action button to submit the assignment. 7. The vehicle(s) are then attached to the ride service.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-6 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Ride Service window		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	This will typically flow into UC-DISP.2.2 Create Driver Ride Service Assignment.,		

## Create Vehicle Route Assignment

<b>Use Case ID:</b>	DISP.1.2		
<b>Use Case Name:</b>	Create Vehicle Route Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch assigns a vehicle to a Route Assignment		
<b>Trigger:</b>	1. Dispatch needs to assign a vehicle to a Route Assignment		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Route and stops have already been created		
<b>Postconditions:</b>	1. Vehicle is assigned to route for a set time window.		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Route window 2. Dispatch is brought to the View Vehicle Schedules Screen 3. Dispatch may use a set of filters to narrow down vehicle choice, including the time frame of the Route Assignment. 4. Dispatch selects one or more vehicles that meets the requirements, then presses an action button. 5. Dispatch is then presented with a question of what timeframe the Route Assignment needs to have the vehicle. 6. Once the dispatch rep has selected those options, they then press one final action button to submit the assignment. 7. The vehicle(s) are then attached to the route.		
<b>Alternative Flows:</b>	1a. In step 6 of the normal flow, the dispatch rep may select hours that the vehicle is not available. 7a. The system will display an error explaining that the vehicle is unavailable at those times due to scheduling conflicts 8a. The screen will return to the timeframe the route assignment needs the vehicle, step 5 of the application..		
<b>Exceptions:</b>	1a. In step 2-6 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	This will typically flow straight into DISP.2.2 Create Driver Route Assignment		

## Update Vehicle Charter Assignment

<b>Use Case ID:</b>	DISP.1.3		
<b>Use Case Name:</b>	Update Vehicle Charter Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reassigns a different vehicle to a charter request that already had a vehicle		
<b>Trigger:</b>	1. Dispatch needs to reassign a vehicle to a charter request		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Route and stops have already been created 3. A vehicle has already been attached to charter request		
<b>Postconditions:</b>	1. A different vehicle is assigned to that route for that set time window. 2. The original assigned vehicle is freed from that route and time window.		
<b>Normal Flow:</b>	1. Dispatch selects a button from the view charter request window. 2. Dispatch is brought to the View Vehicle Schedules screen, with the filters for time already narrowed down to match the charter request. 3. Dispatch selects one vehicle that matches the requirements from this screen and presses an action button. 4. The vehicle is updated in the charter request, and the dispatch rep is brought back to the view charter request screen.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 1-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View charter request window		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Update Vehicle Ride Service Assignment

<b>Use Case ID:</b>	DISP.1.3		
<b>Use Case Name:</b>	Update Vehicle Ride Service Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reassigns a different vehicle to a Ride Service that already had a vehicle		
<b>Trigger:</b>	1. Dispatch needs to reassign a vehicle to a Ride Service		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Route and stops have already been created 3. A vehicle has already been attached to Ride Service		
<b>Postconditions:</b>	1. A different vehicle is assigned to that route for that set time window. 2. The original assigned vehicle is freed from that route and time window.		
<b>Normal Flow:</b>	1. Dispatch selects a button from the view Ride Service window. 2. Dispatch is brought to the View Vehicle Schedules screen, with the filters for time already narrowed down to match the Ride Service . 3. Dispatch selects one vehicle that matches the requirements from this screen and presses an action button. 4. The vehicle is updated in the Ride Service, and the dispatch rep is brought back to the view Ride Service screen.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 1-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Ride Service window		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Update Vehicle Route Assignment

<b>Use Case ID:</b>	DISP.1.2		
<b>Use Case Name:</b>	Update Vehicle Route Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reassigns a different vehicle to a route instance that already had a vehicle		
<b>Trigger:</b>	1. Dispatch needs to reassign a vehicle to a route instance		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Route and stops have already been created 3. A vehicle has already been attached to Route Instance		
<b>Postconditions:</b>	1. A different vehicle is assigned to that route for that set time window. 2. The original assigned vehicle is freed from that route and time window.		
<b>Normal Flow:</b>	1. Dispatch selects a button from the view Route Instance window. 2. Dispatch is brought to the View Vehicle Schedules screen, with the filters for time already narrowed down to match the route instance. 3. Dispatch selects one vehicle that matches the requirements from this screen and presses an action button. 4. The vehicle is updated in the route instance, and the dispatch rep is brought back to the view route instance screen.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 1-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route Instance window		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Deactivate Maintenance Instance

<b>Use Case ID:</b>	DISP.1.4		
<b>Use Case Name:</b>	Deactivate Maintenance Instance		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch cancels a scheduled maintenance		
<b>Trigger:</b>	1. Dispatch needs to cancel maintenance for a number of reasons		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Maintenance Instance has already been scheduled		
<b>Postconditions:</b>	1. Maintenance is canceled		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Vehicle Schedules Screen 2. Dispatch is asked to confirm if the scheduled maintenance should be canceled, dispatch selects yes 3. The maintenance is canceled and the dispatch rep is brought back to the View Vehicle Schedules screen.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 3 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	10 times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Deactivate Other Vehicle Unavailability

<b>Use Case ID:</b>	DISP.1.4		
<b>Use Case Name:</b>	Deactivate Other Vehicle Unavailability Instance		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch cancels a scheduled Other Unavailability		
<b>Trigger:</b>	1. Dispatch needs to cancel Other Unavailability for a number of reasons		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Other Unavailability Instance has already been scheduled		
<b>Postconditions:</b>	1. Other Unavailability is canceled		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Vehicle Schedules Screen 2. Dispatch is asked to confirm if the scheduled Other Unavailability should be canceled, dispatch selects yes 3. The Other Unavailability is canceled and the dispatch rep is brought back to the View Vehicle Schedules screen.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 3 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	10 times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reactive Maintenance Instance

<b>Use Case ID:</b>	DISP.1.5		
<b>Use Case Name:</b>	Reactivate Maintenance Instance Instance		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch un-cancels a scheduled Maintenance Instance		
<b>Trigger:</b>	1. Dispatch needs to bring back canceled Maintenance Instance		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Maintenance Instance Instance has already been scheduled and deactivated		
<b>Postconditions:</b>	1. Maintenance Instance is reactivated and blocks out that section of time		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Vehicle Schedules Screen to show deactivated records 2. Dispatch presses an action button on the view vehicle schedules screen 3. Dispatch is asked to confirm if the scheduled Maintenance Instance should be reactivated, dispatch selects yes 4. The Maintenance Instance is reactivated and the dispatch rep is brought back to the View Vehicle Schedules screen (active records).		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 3 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	10 times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reactive Other Vehicle Unavailability

<b>Use Case ID:</b>	DISP.1.5		
<b>Use Case Name:</b>	Reactivate Other Vehicle Unavailability Instance		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch un-cancels a scheduled Other Unavailability		
<b>Trigger:</b>	1. Dispatch needs to bring back canceled Other Unavailability		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Other Unavailability Instance has already been scheduled and deactivated		
<b>Postconditions:</b>	1. Other Unavailability is reactivated and blocks out that section of time		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Vehicle Schedules Screen to show deactivated records 2. Dispatch presses an action button on the view vehicle schedules screen 3. Dispatch is asked to confirm if the scheduled Other Unavailability should be reactivated, dispatch selects yes 4. The Other Unavailability is reactivated and the dispatch rep is brought back to the View Vehicle Schedules screen (active records).		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 3 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	10 times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reschedule Vehicle Maintenance

<b>Use Case ID:</b>	DISP.1.6		
<b>Use Case Name:</b>	Create Vehicle Maintenance Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reschedules a vehicle maintenance		
<b>Trigger:</b>	<ul style="list-style-type: none"> <li>1. The vehicle needs to be available for other duties, or that maintenance window won't work for other reasons</li> </ul>		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Logged in employee has dispatch role</li> <li>2. The vehicle is scheduled for maintenance</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Vehicle is assigned to maintenance for a new time window.</li> </ul>		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Dispatch presses a button from the View Vehicle Schedules window</li> <li>2. Dispatch enters the new time window the vehicle maintenance will take place</li> <li>3. Dispatch presses an action button to submit</li> <li>4. Dispatch confirms their choice</li> <li>5. The new schedule data is entered in</li> </ul>		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	<p>1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable</p> <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed and the user is brought back to the View Vehicle Schedules window</li> </ul> <p>1b. In step 4 of the normal flow, the user may submit a time window that would conflict with other obligations.</p> <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed and the user is brought back to step 2 of the normal flow.</li> </ul>		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## View Driver Schedules

<b>Use Case ID:</b>	DISP.2.1		
<b>Use Case Name:</b>	View Driver Schedules		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch views a list of all drivers and their schedules		
<b>Trigger:</b>	<ul style="list-style-type: none"> <li>1. Dispatch needs to assign, update or deactivate/reactivate Route Instances</li> <li>2. Dispatch needs to assign, update or deactivate/reactivate Driver Assignments to Ride Services</li> <li>3. Dispatch needs to assign, update or deactivate/reactivate Driver Assignments to Charters</li> <li>4. Dispatch needs to assign, update, or deactivate/reactivate Driver Schedules</li> <li>5. Dispatch needs to assign, update, or deactivate/reactivate Driver Unavailability</li> </ul>		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Logged in employee has dispatch role</li> <li>2. (for Trigger 1 through 3) Dispatch has a vehicle waiting a driver assignment for a Route Instance, Ride Service, or Charter</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Dispatch sees a list of all drivers and their schedules</li> </ul>		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Dispatch rep presses a button from the main screen</li> <li>2. Dispatch rep is presented with a list of drivers and their schedules</li> </ul>		
<b>Alternative Flows:</b>	<ul style="list-style-type: none"> <li>1a. Dispatch rep has assigned a vehicle to a route instance, ride service, or charter(where a driver is needed).</li> <li>2. Dispatch rep is brought directly to this screen with the data of the route instance, ride service, or charter(where a driver is needed) is stored in memory for assignment.</li> </ul>		
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>1a. In step 1 of the normal flow or step 1a of the alternative flow, The databases containing the data may be unavailable <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed and the user is brought back to the prior screen</li> </ul> </li> </ul>		
<b>Includes:</b>	DISP.1.1		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	Charter has driver request Driver is already in system		
<b>Notes and Issues:</b>	N/A		

## Create Driver Charter Assignment

<b>Use Case ID:</b>	DISP.2.2		
<b>Use Case Name:</b>	Create Driver Charter Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch assigns a driver to a Charter Assignment		
<b>Trigger:</b>	1. Dispatch assigned a vehicle to a Charter assignment 2. Charter assignment requires a driver		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Charter and stops have already been created 3. Vehicle has been selected		
<b>Postconditions:</b>	1. Driver is assigned to Charter for a set time window.		
<b>Normal Flow:</b>	1. Dispatch finishes assigning vehicle to Charter 2. Dispatch is brought to a screen viewing all Driver schedules, with the filters for the time window the vehicle was scheduled for and the minimum driver's license qualification. 3. Dispatch may further filter, then select a driver from the list. 4. Dispatch is asked to confirm assignment. 5. When confirmed, the driver is attached to the Charter instance		
<b>Alternative Flows:</b>	Dispatch may choose to cancel this flow at any point. 1. If so, they would be brought back to the view vehicle schedules screen and the vehicle Charter assignment would be canceled.		
<b>Exceptions:</b>	1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Charter Request window		
<b>Includes:</b>	DISP.2.1 View Driver Schedules DISP.1.2 Create Vehicle Charter Assignment		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	The system should not allow the user to change the timeframe or Drivers License qualification filters.		

## Create Driver Ride Service Assignment

<b>Use Case ID:</b>	DISP.2.2		
<b>Use Case Name:</b>	Create Driver Ride Service Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch assigns a driver to a Ride Service Assignment		
<b>Trigger:</b>	1. Dispatch assigned a vehicle to a Ride Service assignment		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Ride Service and stops have already been created 3. Vehicle has been selected		
<b>Postconditions:</b>	1. Driver is assigned to Ride Service for a set time window.		
<b>Normal Flow:</b>	1. Dispatch finishes assigning vehicle to Ride Service 2. Dispatch is brought to a screen viewing all Driver schedules, with the filters for the time window the vehicle was scheduled for and the minimum driver's license qualification. 3. Dispatch may further filter, then select a driver from the list. 4. Dispatch is asked to confirm assignment. 5. When confirmed, the driver is attached to the Ride Service instance		
<b>Alternative Flows:</b>	Dispatch may choose to cancel this flow at any point. 1. If so, they would be brought back to the view vehicle schedules screen and the vehicle Ride Service assignment would be canceled.		
<b>Exceptions:</b>	1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Ride Service window		
<b>Includes:</b>	DISP.2.1 View Driver Schedules DISP.1.2 Create Vehicle Ride Service Assignment		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	The system should not allow the user to change the timeframe or Drivers License qualification filters.		

## Create Driver Route Assignment

<b>Use Case ID:</b>	DISP.2.2		
<b>Use Case Name:</b>	Create Driver Route Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch assigns a driver to a Route Assignment		
<b>Trigger:</b>	1. Dispatch assigned a vehicle to a route assignment		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Route and stops have already been created 3. Vehicle has been selected		
<b>Postconditions:</b>	1. Driver is assigned to route for a set time window.		
<b>Normal Flow:</b>	1. Dispatch finishes assigning vehicle to route 2. Dispatch is brought to a screen viewing all Driver schedules, with the filters for the time window the vehicle was scheduled for and the minimum driver's license qualification. 3. Dispatch may further filter, then select a driver from the list. 4. Dispatch is asked to confirm assignment. 5. When confirmed, the driver is attached to the route instance		
<b>Alternative Flows:</b>	Dispatch may choose to cancel this flow at any point. 1. If so, they would be brought back to the view vehicle schedules screen and the vehicle route assignment would be canceled		
<b>Exceptions:</b>	1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window		
<b>Includes:</b>	DISP.2.1 View Driver Schedules DISP.1.2 Create Vehicle Route Assignment		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	The system should not allow the user to change the timeframe or Drivers License qualification filters.		

## Create Driver Other Unavailability Assignment

<b>Use Case ID:</b>	DISP.2.2		
<b>Use Case Name:</b>	Create Driver other unavailability Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/11/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch marks a Driver as unavailable		
<b>Trigger:</b>	1. Dispatch needs to mark a Driver as unavailable for an abnormal reason		
<b>Preconditions:</b>	1. Logged in employee has dispatch role		
<b>Postconditions:</b>	1. Driver is marked unavailable for the set time window.		
<b>Normal Flow:</b>	1. Dispatch presses a button from the view Driver schedules screen associated with a specific Driver. 2. Dispatch is then presented with a question of what timeframe the unavailability needs to have the Driver, and which days of the week 3. Dispatch enters a description in a text field to describe the reason the Driver is unavailable. 4. Once the dispatch rep has selected those options, they then press one final action button to submit the assignment. 5. The Driver(s) are then attached to the other unavailability.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View other unavailability window		
<b>Includes:</b>	DISP.1.1 View Driver Schedules		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Update Driver Charter Assignment

<b>Use Case ID:</b>	DISP.2.3		
<b>Use Case Name:</b>	Update Driver Charter Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reassigns a different Driver to a charter request that already had a Driver		
<b>Trigger:</b>	1. Dispatch needs to reassign a Driver to a charter request		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Route and stops have already been created 3. A Driver has already been attached to charter request		
<b>Postconditions:</b>	1. A different Driver is assigned to that route for that set time window. 2. The original assigned Driver is freed from that route and time window.		
<b>Normal Flow:</b>	1. Dispatch selects a button from the view charter request window. 2. Dispatch is brought to the View Driver Schedules screen, with the filters for time already narrowed down to match the charter request. 3. Dispatch selects one Driver that matches the requirements from this screen and presses an action button. 4. The Driver is updated in the charter request, and the dispatch rep is brought back to the view charter request screen.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 1-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View charter request window		
<b>Includes:</b>	DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Update Driver Ride Service Assignment

<b>Use Case ID:</b>	DISP.2.3		
<b>Use Case Name:</b>	Update Driver Ride Service Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reassigns a different Driver to a Ride Service that already had a Driver		
<b>Trigger:</b>	1. Dispatch needs to reassign a Driver to a Ride Service		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Route and stops have already been created 3. A Driver has already been attached to Ride Service		
<b>Postconditions:</b>	1. A different Driver is assigned to that route for that set time window. 2. The original assigned Driver is freed from that route and time window.		
<b>Normal Flow:</b>	1. Dispatch selects a button from the view Ride Service window. 2. Dispatch is brought to the View Driver Schedules screen, with the filters for time already narrowed down to match the Ride Service . 3. Dispatch selects one Driver that matches the requirements from this screen and presses an action button. 4. The Driver is updated in the Ride Service, and the dispatch rep is brought back to the view Ride Service screen.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 1-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Ride Service window		
<b>Includes:</b>	DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Update Driver Route Assignment

<b>Use Case ID:</b>	DISP.2.2		
<b>Use Case Name:</b>	Update Driver Route Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reassigns a different Driver to a route instance that already had a Driver		
<b>Trigger:</b>	1. Dispatch needs to reassign a Driver to a route instance		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Route and stops have already been created 3. A Driver has already been attached to Route Instance		
<b>Postconditions:</b>	1. A different Driver is assigned to that route for that set time window. 2. The original assigned Driver is freed from that route and time window.		
<b>Normal Flow:</b>	1. Dispatch selects a button from the view Route Instance window. 2. Dispatch is brought to the View Driver Schedules screen, with the filters for time already narrowed down to match the route instance. 3. Dispatch selects one Driver that matches the requirements from this screen and presses an action button. 4. The Driver is updated in the route instance, and the dispatch rep is brought back to the view route instance screen.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 1-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route Instance window		
<b>Includes:</b>	DISP.2.1 View Drivers		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Deactivate Other Driver Unavailability

<b>Use Case ID:</b>	DISP.2.4		
<b>Use Case Name:</b>	Deactivate Other Driver Unavailability Instance		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch cancels a scheduled Other Unavailability		
<b>Trigger:</b>	<ol style="list-style-type: none"> <li>1. Dispatch needs to cancel Other Unavailability for a number of reasons</li> </ol>		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Logged in employee has dispatch role</li> <li>2. Other Unavailability Instance has already been scheduled</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Other Unavailability is canceled</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Dispatch presses a button from the View Driver Schedules Screen</li> <li>2. Dispatch is asked to confirm if the scheduled Other Unavailability should be canceled, dispatch selects yes</li> <li>3. The Other Unavailability is canceled and the dispatch rep is brought back to the View Driver Schedules screen.</li> </ol>		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	<p>1a. In step 3 of the normal flow, The databases containing the data may be unavailable</p> <ol style="list-style-type: none"> <li>1. In this situation, an error is displayed and the user is brought back to the View Route window</li> </ol>		
<b>Includes:</b>	DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	10 times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reactivate Other Driver Unavailability

<b>Use Case ID:</b>	DISP.1.5		
<b>Use Case Name:</b>	Reactivate Other Driver Unavailability Instance		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch un-cancels a scheduled Other Unavailability		
<b>Trigger:</b>	1. Dispatch needs to bring back canceled Other Unavailability		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. Other Unavailability Instance has already been scheduled and deactivated		
<b>Postconditions:</b>	1. Other Unavailability is reactivated and blocks out that section of time		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Driver Schedules Screen to show deactivated records 2. Dispatch presses an action button on the view Driver schedules screen 3. Dispatch is asked to confirm if the scheduled Other Unavailability should be reactivated, dispatch selects yes 4. The Other Unavailability is reactivated and the dispatch rep is brought back to the View Driver Schedules screen (active records).		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 3 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window		
<b>Includes:</b>	DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	10 times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reschedule Driver Unavailability

<b>Use Case ID:</b>	DISP.1.6		
<b>Use Case Name:</b>	Create Vehicle Maintenance Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reschedules a driver unavailability		
<b>Trigger:</b>	1. The driver's situation changes and their unavailability shifts		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. The driver is scheduled for other unavailability		
<b>Postconditions:</b>	1. Driver is assigned to be unavailable for a new time window.		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Driver Schedules window 2. Dispatch enters the new time window the driver unavailability will take place 3. Dispatch presses an action button to submit 4. Dispatch confirms their choice 5. The new schedule data is entered in		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Driver Schedules window 1b. In step 4 of the normal flow, the user may submit a time window that would conflict with other obligations. 1. In this situation, an error is displayed and the user is brought back to step 2 of the normal flow.		
<b>Includes:</b>	DISP.1.1 View Vehicles		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reschedule Vehicle Driver Assignment

<b>Use Case ID:</b>	DISP.3.1		
<b>Use Case Name:</b>	Reschedule Vehicle+Driver Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch reschedules a Route Assignment		
<b>Trigger:</b>	1. Dispatch needs to reschedule a route assignment, perhaps to adjust vehicle or driver availability		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. The Assignment has already been created (vehicle/driver already scheduled)		
<b>Postconditions:</b>	1. Both vehicle and driver are rescheduled to work new hours		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Vehicle Schedules or View Driver Schedules windows 2. Dispatch is brought to the Reschedule Assignment screen, which shows both the vehicle and driver schedules. 3. Dispatch selects the new times that the Route/Ride Service/Charter Assignment should cover 4. Dispatch presses an action button to submit 5. The Assignment is rescheduled		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed and the user is brought back to the View Route window</li> </ul> 1b. In step 4 of the normal flow, Dispatch may attempt to schedule at a time where either the driver or vehicle is already scheduled <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed and Dispatch is brought back to step 2 of the normal flow.</li> </ul>		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Deactivate Vehicle Driver Assignment

<b>Use Case ID:</b>	DISP.3.2		
<b>Use Case Name:</b>	Deactivate Vehicle+Driver Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch deactivates a Route Assignment		
<b>Trigger:</b>	1. Dispatch needs to deactivate a route assignment, perhaps to adjust vehicle or driver availability		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. The Assignment has already been created (vehicle/driver already scheduled)		
<b>Postconditions:</b>	1. Both vehicle and driver are free to work other services		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Vehicle Schedules or View Driver Schedules windows 2. Dispatch is brought to the Reschedule Assignment screen, which shows both the vehicle and driver schedules. 3. Dispatch selects an action button that states "deactivate" 4. The Assignment is deactivated		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Route window.		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reactivate Vehicle Driver Assignment

<b>Use Case ID:</b>	DISP.3.3		
<b>Use Case Name:</b>	Reactivate Vehicle+Driver Assignment		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/13/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Dispatch		
<b>Description:</b>	Dispatch Reactivates a Route Assignment		
<b>Trigger:</b>	1. Dispatch needs to Reactivate a route assignment, perhaps to adjust vehicle or driver availability		
<b>Preconditions:</b>	1. Logged in employee has dispatch role 2. The Assignment has already been created (vehicle/driver already scheduled)		
<b>Postconditions:</b>	1. Both vehicle and driver are free to work other services		
<b>Normal Flow:</b>	1. Dispatch presses a button from the View Vehicle Schedules or View Driver Schedules windows to show deactivated records 2. Dispatch presses a button from the View Vehicle Schedules or View Driver Schedules windows 3. Dispatch is brought to the Reschedule Assignment screen, which shows both the vehicle and driver schedules. 4. Dispatch selects an action button that states "Reactivate" 5. The Assignment is Reactivated		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	1a. In step 2-5 of the normal flow, The databases containing the data may be unavailable <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed and the user is brought back to the View Route window.</li> </ul>		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	20+ times per day		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Create Driver

<b>Use Case ID:</b>	OPS.5.1		
<b>Use Case Name:</b>	Create Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager assigns an employee to be a driver		
<b>Trigger:</b>	1. An employee has agreed to be a driver		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The new driver's driving schedule is known 3. The new driver's license class is known 4. The new driver is in the database as an employee.		
<b>Postconditions:</b>	1. The driver is added to the database of available drivers.		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen 2. Operations Manager is brought to the add driver window, where operations manager enters the driver's weekly available hours and Driver's License Class 3. Operations Manager selects an action button to save the details 4. Details are stored in database 5. Operations manager is brought back to the driver schedule screen.		
<b>Alternative Flows:</b>	In steps 2-3 of the normal flow, Operations Manager may choose to cancel the action. 1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled and the Ops Manager is brought back to the driver schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## View Drivers

<b>Use Case ID:</b>	OPS.5.2		
<b>Use Case Name:</b>	Create Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager views a driver's information		
<b>Trigger:</b>	1. A driver's availability or license need to be known		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The driver is already in the database		
<b>Postconditions:</b>	1. The driver's information is displayed		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen 2. Operations Manager is brought to the view driver window		
<b>Alternative Flows:</b>	1. N/A		
<b>Exceptions:</b>	1a. In step 2 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Update Driver

<b>Use Case ID:</b>	OPS.5.3		
<b>Use Case Name:</b>	Create Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager modifies a driver's hours or license		
<b>Trigger:</b>	1. A driver's availability or license change		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The driver's new information is known 3. The driver is already in the database		
<b>Postconditions:</b>	1. The driver is updated to reflect the new information.		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen 2. Operations Manager is brought to the view driver window 3. Operations Manager presses a button to change to edit driver 4. Operations Manager then enters the modified data in the correct fields 5. Operations Manager selects an action button to save the details 6. Details are stored in database 7. Operations manager is brought back to the driver schedule screen.		
<b>Alternative Flows:</b>	In steps 3-5 of the normal flow, Operations Manager may choose to cancel the action. 1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled and the Ops Manager is brought back to the driver schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules OPS.5.2 View Driver		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Deactivate Driver

<b>Use Case ID:</b>	OPS.5.4		
<b>Use Case Name:</b>	Deactivate Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager deactivates a driver		
<b>Trigger:</b>	1. The driver no longer works for the company		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The driver is already in the database		
<b>Postconditions:</b>	1. The driver is deactivated in the database and no longer appears in normal view		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen 2. Operations Manager is brought to the view driver window 3. Operations Manager presses a button to change to deactivate driver 4. Operations Manager is prompted to confirm the change 5. Operations Manager selects an action button to save the details 6. Details are stored in database 7. Operations manager is brought back to the driver schedule screen.		
<b>Alternative Flows:</b>	In steps 3-5 of the normal flow, Operations Manager may choose to cancel the action. 1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled and the Ops Manager is brought back to the driver schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules OPS.5.2 View Driver		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reactivate Driver

<b>Use Case ID:</b>	OPS.5.5		
<b>Use Case Name:</b>	Reactivate Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager reactivates a driver		
<b>Trigger:</b>	1. The driver works for the company again		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The driver is already in the database 3. The driver is deactivated		
<b>Postconditions:</b>	1. The driver is deactivated in the database and no longer appears in normal view		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen to show inactive drivers 2. Operations manager selects a button from the driver schedule screen 3. Operations Manager is brought to the view driver window 4. Operations Manager presses a button to change to deactivate driver 5. Operations Manager is prompted to confirm 6. Operations Manager selects an action button to save the details 7. Details are stored in database 8. Operations manager is brought back to the driver schedule screen.		
<b>Alternative Flows:</b>	In steps 4-6 of the normal flow, Operations Manager may choose to cancel the action. 1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled and the Ops Manager is brought back to the driver schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules OPS.5.2 View Driver		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Create Vehicle

<b>Use Case ID:</b>	OPS.6.1		
<b>Use Case Name:</b>	Create Vehicle		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager adds a new vehicle to the database		
<b>Trigger:</b>	1. A vehicle is purchased or otherwise added to the fleet		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The new Vehicle's basic information is known		
<b>Postconditions:</b>	1. The Vehicle is added to the database of available Vehicles.		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the Vehicle schedule screen 2. Operations Manager is brought to the add Vehicle window, where operations manager enters the vehicle's information. 3. Operations Manager selects an action button to save the details 4. Details are stored in database 5. Operations manager is brought back to the Vehicle schedule screen.		
<b>Alternative Flows:</b>	In steps 2-3 of the normal flow, Operations Manager may choose to cancel the action. 1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled and the Ops Manager is brought back to the Vehicle schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Vehicle Schedules window.		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Deactivate Vehicle

<b>Use Case ID:</b>	OPS.6.1		
<b>Use Case Name:</b>	Deactivate Vehicle		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager removes a vehicle from active service		
<b>Trigger:</b>	<ul style="list-style-type: none"> <li>1. A vehicle's time in active service has come to an end due to aging out</li> <li>2. A vehicle's time in active service has come to an end due to an accident</li> </ul>		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Logged in employee has Operations Manager role</li> <li>2. The vehicle is active in the system</li> </ul>		
<b>Postconditions:</b>	The Vehicle is removed from the list of active vehicles		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Operations Manager clicks a button from the Vehicle schedule screen</li> <li>2. Operations Manager is asked to confirm they wish to deactivate the selected vehicle</li> <li>3. Operations Manager selects an action button to confirm</li> <li>4. Details are stored in database</li> <li>5. Operations manager is brought back to the Vehicle schedule screen.</li> </ul>		
<b>Alternative Flows:</b>	<p>In steps 2-3 of the normal flow, Operations Manager may choose to cancel the action.</p> <ul style="list-style-type: none"> <li>1. Operations Manager selects a cancel button or closes the window.</li> <li>2. The action is canceled and the Ops Manager is brought back to the Vehicle schedule screen.</li> </ul>		
<b>Exceptions:</b>	<p>1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable</p> <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed and the user is brought back to the View Vehicle Schedules window.</li> </ul>		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reactivate Vehicle

<b>Use Case ID:</b>	OPS.6.3		
<b>Use Case Name:</b>	Reactivate Vehicle		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager brings a vehicle back into active service		
<b>Trigger:</b>	1. A vehicle that was previously considered unfit for work needs to be used for work.		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The vehicle is deactivated in the system		
<b>Postconditions:</b>	1. The Vehicle is brought back to the list of active vehicles		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the Vehicle schedule screen to view inactive vehicles 2. Operations manager clicks an action button associated with a specific vehicle 3. Operations Manager is asked to confirm they wish to reactivate the selected vehicle 4. Operations Manager selects an action button to confirm 5. Details are stored in database 6. Operations manager is brought back to the Vehicle schedule screen.		
<b>Alternative Flows:</b>	In steps 2-3 of the normal flow, Operations Manager may choose to cancel the action. 1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled and the Ops Manager is brought back to the Vehicle schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, The databases containing the data may be unavailable 1. In this situation, an error is displayed and the user is brought back to the View Vehicle Schedules window.		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

# Employee – Fleet-Manager

## Add Vehicle to Inventory

<b>Use Case ID:</b>	FADM 1.01		
<b>Use Case Name:</b>	Add Vehicle		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/07/2023	<b>Last Revision Date:</b>	09/07/2023
<b>Actors:</b>	Fleet Manager		
<b>Description:</b>	Add a vehicle to inventory for use.		
<b>Trigger:</b>	New vehicle is purchased		
<b>Preconditions:</b>	1. New vehicle is delivered 2. Fleet Manager is Authenticated		
<b>Postconditions:</b>	1. Vehicle is available for use		
<b>Normal Flow:</b>	1. Fleet Manager Logs In 2. Navigates to vehicle list 3. Selects to add vehicle 4. Enters VIN 5. Enters Make 6. Enter Model 7. Enters Model Year 8. Enters starting mileage 9. Enter License Plate 10. Submit		
<b>Alternative Flows:</b>	9.a After step 8 for used/donated vehicles add maintenance history 1. Enter maintenance information 2. Repeat until all the history is entered. 3. Return to step 9		
<b>Exceptions:</b>	4a Validate VIN 1. Prompt user to verify the VIN 2. Return user to step 4 to update the VIN.		
<b>Includes:</b>	9.1.3 Lookup Vehicle List		
<b>Frequency of Use:</b>	Every time a vehicle is purchased		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Manager has access to vehicle to check information.		
<b>Notes and Issues:</b>			

## Deactivate Vehicle from Inventory

<b>Use Case ID:</b>	FADM 1.02		
<b>Use Case Name:</b>	Deactivate Vehicle		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/07/2023	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Fleet manager		
<b>Description:</b>	Remove a vehicle from the list if wrecked, stolen or sold.		
<b>Trigger:</b>	1a. Vehicle is damaged beyond repair. 1b. Vehicle is stolen and unable to be retrieved 1c. Vehicle is sold.		
<b>Preconditions:</b>	1. Vehicle is permanently not in use 2. Fleet manager is authenticated		
<b>Postconditions:</b>	1. Vehicle is no longer available. 2. Vehicle data is moved to archive		
<b>Normal Flow:</b>	1. Fleet Manager logs in 2. Navigates to the vehicle list 3. Selects vehicle from list 4. Selects to deactivate vehicle 5. Enters the reason for vehicle deactivation 6. Enters the final mileage 7. Submit 8. Confirm removal		
<b>Alternative Flow:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	9.1.3 Lookup Vehicle List		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Manager is able to access the vehicle to get the mileage		
<b>Notes and Issues:</b>			

## Look up Vehicle List

<b>Use Case ID:</b>	FADM 1.03		
<b>Use Case Name:</b>	Lookup vehicle		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/07/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Employee		
<b>Description:</b>	Lookup a list of current vehicles		
<b>Trigger:</b>	Need information about a vehicle Need a list of current vehicles		
<b>Preconditions:</b>	1. User is authenticated		
<b>Postconditions:</b>	1. List is displayed to the user 2. Shows brief summary of truck information		
<b>Normal Flow:</b>	1. User logs in 2. Select to view vehicle list		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	Database not available 1. Display Unable to connect to database		
<b>Includes:</b>			
<b>Frequency of Use:</b>	Multiple times a day		
<b>Special Requirements:</b>	Sort the list Filter the list		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1. Vehicle Number, Make, Model, Mileage		

## Look up Vehicle Information

<b>Use Case ID:</b>	FADM 1.04		
<b>Use Case Name:</b>	Lookup Vehicle Info		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/07/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager Operations Manager Maintenance Manager Mechanic		
<b>Description:</b>	Lookup a vehicle and get the detailed information		
<b>Trigger:</b>	Need specific information about a vehicle		
<b>Preconditions:</b>	1. User is authenticated		
<b>Postconditions:</b>	1. Shows detailed information about the vehicle. 2. Option to see vehicle maintenance logs		
<b>Normal Flow:</b>	1. User logs in 2. Selects to view vehicle list 3. Searches for vehicle 4. Selects the found vehicle 5. Displays vehicle detailed information.		
<b>Alternative Flows:</b>	5.a Selects to view vehicle maintenance logs 1. Selects to view vehicle maintenance logs 2. Displays list of logs 3. Selects log to view 4. Closes and returns to step 5		
<b>Exceptions:</b>	1.		
<b>Includes:</b>	<a href="#">9.1.3 Lookup vehicle list</a> <a href="#">9.2.2 Lookup maintenance logs</a>		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1. Vehicle Number, VIN, Make, Model, Model Year, Mileage, License Plate, Date Entered		

## Change Vehicle Information

<b>Use Case ID:</b>	FADM 1.05		
<b>Use Case Name:</b>	Change Vehicle Info		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/07/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager		
<b>Description:</b>	Lookup a vehicle and edit the detailed information		
<b>Trigger:</b>	Need to change the information for a vehicle		
<b>Preconditions:</b>	1. User is authenticated 2. User has new information for a vehicle		
<b>Postconditions:</b>	1. Shows detailed information about the vehicle.		
<b>Normal Flow:</b>	1. User logs in 2. Selects to view vehicle list 3. Searches for vehicle 4. Selects the found vehicle 5. Displays vehicle detailed information. 6. Update the information as needed 7. Submit 8. Confirm change of information		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	9.1.3 Lookup vehicle list		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Track Vehicle

<b>Use Case ID:</b>	FADM 1.06		
<b>Use Case Name:</b>	Track Vehicle		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/08/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Tracks where a vehicle is with GPS and displays on a map to the user		
<b>Trigger:</b>	Vehicle is unable to be located at the designated storage location Unable to contact who has the vehicle		
<b>Preconditions:</b>	1. Fleet Manager or Operations Manager authenticated 2. Vehicle is deemed missing		
<b>Postconditions:</b>	1. Vehicle location is provided to the user		
<b>Normal Flow:</b>	1. Manager logs in 2. Selects vehicle lookup list 3. Searches for vehicle 4. Selects to view vehicle information 5. Selects to track vehicle 6. Vehicle location is retrieved and displayed on a map		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	6a. GPS data is not available for the vehicle 1. Display message to user that vehicle is unable to be located 2. Inform user data may not be available if vehicle is not outside or if the GPS system on the vehicle is not functioning properly 3. Return to vehicle information		
<b>Includes:</b>	9.1.4 Lookup vehicle info		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	GPS data needs to be able to be retrieved from GPS system or a service provider		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Remove Part from Inventory

<b>Use Case ID:</b>	FADM 2.07		
<b>Use Case Name:</b>	Remove Part from Inventory		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/08/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager Parts Person		
<b>Description:</b>	Change the quantity of parts		
<b>Trigger:</b>	Part count is different than what is in the system Part is no longer needed and sold		
<b>Preconditions:</b>	1. User is authenticated		
<b>Postconditions:</b>	1. Part count is updated		
<b>Normal Flow:</b>	1. User logs in 2. User selects maintenance system 3. User selects to view parts list 4. Part is selected from the list 5. Selects to view detailed information 6. User selects to update qty 7. User changes the qty listed 8. Prompts user to confirm the change 9. Part qty is updated 10. User is returned to parts list		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	9.2.3 Lookup parts list 9.2.4 Parts Detail		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Import Parts List

<b>Use Case ID:</b>	FADM 2.08		
<b>Use Case Name:</b>	Import Parts List		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager Parts Person		
<b>Description:</b>	Import a vendors parts list for ordering parts		
<b>Trigger:</b>	Parts list is obtained from vendor		
<b>Preconditions:</b>	1. User is logged in 2. Parts list is .cvs or other acceptable format 3. Parts list has correct columns 4. Parts list is on computer		
<b>Postconditions:</b>	1. Parts list is available in parts system		
<b>Normal Flow:</b>	1. Navigate to parts system 2. Selects vendor from list 3. Selects to upload parts list 4. Select parts list from file system 5. Review list 6. Confirm upload (replace previous list)		
<b>Alternative Flows:</b>	5a Parts added/removed from vendor 1. Notify new parts available 2. Notify parts no longer available (will still be in inventory) 3. Confirm 4. return to step 5		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	Monthly/Yearly		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Parts PO Detail

<b>Use Case ID:</b>	FADM 2.09		
<b>Use Case Name:</b>	Parts PO Detail		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/28/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager Parts Person		
<b>Description:</b>	Lookup a purchase order(PO) that was already created		
<b>Trigger:</b>	Need to review a PO that has been created		
<b>Preconditions:</b>	1. PO has already been created		
<b>Postconditions:</b>	1. Purchase order is displayed to user		
<b>Normal Flow:</b>	1. User is authenticated 2. User selects maintenance system 3. User selects parts list 4. User selects to search POs 5. User selects PO from list 6. PO is displayed to user		
<b>Alternative Flows:</b>	6a Cancel PO 1. User selects to cancel PO 2. Prompts user to verify the vendor has been contacted to cancel it 3. PO is marked as canceled and archived 4. User is returned to Parts List		
<b>Exceptions:</b>			
<b>Includes:</b>	9.2.3 Lookup parts list		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1. Make sure to include vendor parts		

## Look up Fuel Receipts

<b>Use Case ID:</b>	FADM 3.01		
<b>Use Case Name:</b>	Lookup Fuel Receipts		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/09/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Lookup a list of fuel receipts		
<b>Trigger:</b>	User wants to review fuel purchases		
<b>Preconditions:</b>	1. Driver has entered fuel receipts for a vehicle		
<b>Postconditions:</b>	Fuel receipt information is shown		
<b>Normal Flow:</b>	1. User is authenticated 2. User selects to review fuel receipts 3. User selects to review by driver 4. User selects the driver from a list 5. User is shown a list of the drivers fuel purchases 6. User returns to main screen		
<b>Alternative Flows:</b>	3a User selects to review by vehicle 1. User selects to review by vehicle 2. User selects the vehicle from a list 3. User is shown a list of the vehicles fuel receipts 4. User returns to the main screen		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Look up Vehicle Inspection Report

<b>Use Case ID:</b>	FADM 4.01		
<b>Use Case Name:</b>	Lookup Vehicle Inspection Report		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/09/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager Operations Manager Maintenance Manager Mechanic		
<b>Description:</b>	Lookup the inspection report from a vehicles use		
<b>Trigger:</b>	Inspection Report needs to be reviewed by maintenance Inspection Report needs to be reviewed by Police/DOT		
<b>Preconditions:</b>	1. Vehicle has been inspected and used		
<b>Postconditions:</b>	1. Inspection Report is displayed		
<b>Normal Flow:</b>	1. User is logs in 2. User selects to view vehicle list 3. Searches for vehicle 4. Selects the found vehicle 5. Displays vehicle detailed information. 6. User selects to view Inspection Report list 7. User selects the Inspection Report 8. Report is displayed to the user		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	9.1.3 Lookup Vehicle List 9.1.4 Lookup Vehicle Details		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1. Not all vehicles require inspections		

## Add Rental Vehicle

<b>Use Case ID:</b>	FADM 5.01		
<b>Use Case Name:</b>	Add Rental Vehicle		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Jared Roberts
<b>Date Created:</b>	09/07/2023	<b>Last Revision Date:</b>	10/29/2023
<b>Actors:</b>	Fleet Manager		
<b>Description:</b>	Add a Rental vehicle to inventory for use.		
<b>Trigger:</b>	Vehicle is rented to fill need		
<b>Preconditions:</b>	1. Vehicle is picked up from rental agency 2. Fleet Manager is Authenticated		
<b>Postconditions:</b>	1. Vehicle is available for use		
<b>Normal Flow:</b>	1. Fleet Manager Logs In 2. Navigates to vehicle list 3. Selects to add rental vehicle 4. Enters VIN 5. Enters Make 6. Enters Model 7. Enters Model Year 8. Enters starting mileage 9. Enters License Plate 10. Enters Rental Agency 11. Enters Rental Date 12. Enters Rental Return Date 13. Enters Rental Cost/Day 14. Submit		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	9.1.3 Lookup Vehicle List		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Manager has access to vehicle to check information.		
<b>Notes and Issues:</b>			



## Employee – Maintenance

### View Vehicles with Pending Work

<b>Use Case ID:</b>	UC-MNTC.01.01		
<b>Use Case Name:</b>	View Vehicles Pending Work		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/9/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	View a list of vehicles with pending work orders		
<b>Trigger:</b>			
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view the vehicle's details.		
<b>Postconditions:</b>			
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the user has an incorrect login 1. The login is rejected. 2. Display an error message. 3. Use Case resumes on step 2 of normal flow.		
<b>Includes:</b>	User Login		
<b>Frequency of Use:</b>	50 times per day.		
<b>Special Requirements:</b>	A manual refresh button		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	This should only return a list of vehicles with pending work orders. We need a stored procedure for that. Another filter that's able to limit the selections to cars in the garage would also be very useful.		

## View Vehicle Details

<b>Use Case ID:</b>	UC-MNTC.02.01		
<b>Use Case Name:</b>	View Vehicle Details		
<b>Created By:</b>	Nik Bell	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	10/9/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	View the details of a vehicle with pending work orders		
<b>Trigger:</b>	The user clicks on a vehicle in UC-MNTC.01.01		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view the vehicle's details.		
<b>Postconditions:</b>	None.		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to its details page.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	5a. If the vehicle selected has no pending work orders. This can happen if the database is down, or someone completed the work orders. 1. Identify whether the database is down or there are no pending work orders. 2. Create a relevant error box popup. 3. Return user to step 4.		
<b>Includes:</b>	User Login View Vehicle Work Orders		
<b>Frequency of Use:</b>	50 times per day.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Update Vehicle Details

<b>Use Case ID:</b>	UC-MNTC.02.02		
<b>Use Case Name:</b>	View Vehicle Details		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/7/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	View the details of a vehicle with pending work orders		
<b>Trigger:</b>	The user clicks on a vehicle in UC-MNTC.01.01		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view the vehicle's details.		
<b>Postconditions:</b>	1. The edited information is saved and updated in the DB 2. The details page reloads the View Details tab with the updated information.		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to its details page. 6. The user clicks the Edit button. 7. The details page reloads allowing the details to be edited. 8. The user edits the details as needed and clicks save.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	8a. If the user decides that they don't have any edits 1. The user clicks the Cancel button. 2. The details page reloads the View Details tab with the updated information.		
<b>Includes:</b>	View Vehicle Work Orders		
<b>Frequency of Use:</b>	1 times per week.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## View Work Order List

<b>Use Case ID:</b>	UC-MTNC.03.01		
<b>Use Case Name:</b>	View Detailed Work Order		
<b>Created By:</b>	Nik Bell	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/29/2023	<b>Last Revision Date:</b>	10/7/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	View a breakdown of all pending work orders on a vehicle.		
<b>Trigger:</b>	User clicks on a vehicle in UC-MNTC.01.01		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view the vehicle's details..		
<b>Postconditions:</b>	The user will be viewing a list of pending work orders.		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details page. 6.The user selects the work orders tab and is brought to a list of orders.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	6a. The user clicks on a vehicle with no pending work orders. This can happen because all orders were completed or the database is unreachable. 1. Identify whether the database is down or there are no pending work orders. 2. Create a relevant error box popup. 3. Return user to step 4.		
<b>Includes:</b>	1. User Login 2. View Vehicle's Pending Work		
<b>Frequency of Use:</b>	50 times per day.		
<b>Special Requirements:</b>	A refresh button		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Update Work Orders

<b>Use Case ID:</b>	UC-MTNC.03.02		
<b>Use Case Name:</b>	Update Work Orders		
<b>Created By:</b>	Nik Bell	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	9/22/2023	<b>Last Revision Date:</b>	9/29/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	The user can update a work order clarifying that some work has been done but the issue remains unresolved, or the issue has been properly diagnosed in case of an unclear problem was submitted as a work order.		
<b>Trigger:</b>	The update button is clicked on the Work Orders tab of the details page		
<b>Preconditions:</b>	1. The user is registered in the system. 2. The user has sufficient application rights to view and edit the vehicle's maintenance logs.		
<b>Postconditions:</b>	1. A new work order is created that acts as a substitute for the previous order. Accessing the order should now return the more up to date version. 2. Archive the old work order (UC-MTNC.03.04)		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details UC-MTNC.01.02 6. The user selects the work orders tab and is brought to a list of orders. 7. The user selects updates for a specific order and is brought to a UI where details are entered. 8. The user saves the order and is brought back to step 6.		
<b>Alternative Flows</b>	7a. If the user needs to record inventory used during work. 1. Refer to UC-MTNC.04.01 Deduct Inventory. Can be called multiple times for every variety of item.		
<b>Exceptions:</b>	7a/8a If the database is unable to retrieve the order. 1. Create a warning box about the issue, 2. Return user to step 6. 8b. If the work order has been updated since the user began typing information 1. Before saving, show the user a maintenance log of the most recent update. 2. Ask the user if they wish to still save. 3a. If no, return to step 6. 3b. If yes, save data and return to step 6. 8c. If the order they are attempting to save has already been completed 1. Inform the user the order has already been resolved. 2. return to step 6.		
<b>Includes:</b>	User Login View Vehicle Work Orders View Maintenance Logs		
<b>Frequency of Use:</b>	10 times a day.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			

<b>Notes and Issues:</b>	This should not overwrite the previous work order. Instead this should create a new order entirely, we will need to create pointers and a system to mark the new order as the up to date version.
--------------------------	---

## Complete Work Order

<b>Use Case ID:</b>	UC-MTNC.03.03		
<b>Use Case Name:</b>	Complete Work Order		
<b>Created By:</b>	Nik Bell	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	9/22/2023	<b>Last Revision Date:</b>	9/29/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	Marks work orders as completed		
<b>Trigger:</b>	The user has completed maintenance on a vehicle and needs to report a completed task		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view and edit the vehicle's maintenance logs.		
<b>Postconditions:</b>	1. The work order is completed and is removed from a vehicle. 2. If all work orders are complete, remove the vehicle from the maintenance list. 3. Create Maintenance Log UC-MNTC-03.04		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details UC-MTNC.01.02 6. The user selects the work orders tab and is brought to a list of pending Work Orders. 7. The user selects the Complete button on a specific Work Order and is brought to a UI where completion details are entered. 8. The user enters the information and and is brought back to step 6.		
<b>Alternative Flows:</b>	7a. If the user needs to record inventory used during work. 1. Refer to UC-MTNC.04.01 Deduct Inventory. Can be called multiple times for every variety of item.		
<b>Exceptions:</b>	7a/8a If the database is unable to retrieve the order. 1. Create a warning box about the issue, 2. Return user to step 6. 8b. If the work order has been updated since the user began typing information 1. Before saving, show the user a maintenance log of the most recent update. 2. Ask the user if they wish to still save. 3a. If no, return to step 6. 3b. If yes, save data and return to step 6. 8c. If the order they are attempting to save has already been completed 1. Inform the user the order has already been resolved. 2. return to step 6.		
<b>Includes:</b>	User Login View Vehicle Work Orders View Maintenance Logs Create Maintenance Logs		
<b>Frequency of Use:</b>	2-10 times a day		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			

<b>Notes and Issues:</b>	This entire system is prone to merge conflicts and need to be resolved.
--------------------------	---

### Create Work Order

<b>Use Case ID:</b>	UC-MTNC.03.04		
<b>Use Case Name:</b>	Create Work Order		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/7/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Any staff member		
<b>Description:</b>	Creates a work order which reports an issue or recurring checkup required for the vehicle.		
<b>Trigger:</b>	Staff members who have identified an issue with the vehicles. Automated recurring maintenance system.		
<b>Preconditions:</b>			
<b>Postconditions:</b>	A new active work order is attached to the vehicle.		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details page. 6. The user selects the work orders tab and is brought to a list of work orders. 7. The user selects and adds a new work order. 8. The Create Work Order page opens up. 9. The user enters the work order information. 10. The user clicks the Add Work Order button and is brought back to step 6.		
<b>Alternative Flows:</b>	8a. If the user decides that they don't need to add a work order 1. The user clicks the Cancel button. 2. The Create Work Order page closes, discarding any entered information.. 3. The flow resumes at step 6.		
<b>Exceptions:</b>	Database not found		
<b>Includes:</b>	User Login View Vehicle Work Orders		
<b>Frequency of Use:</b>	0 to 10 times a day		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	This use case is probably going to need to be accessible from so many different employee UIs. Generic issues about the car can also be reported through this system. If the issue doesn't really exist, the order can just be completed.		

## View Maintenance Logs List

<b>Use Case ID:</b>	UC-MNTC.04.01		
<b>Use Case Name:</b>	View Maintenance Logs List		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/30/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	View detailed logs of all maintenance for each vehicle.		
<b>Trigger:</b>	Selecting the Maintenance Log tab on a vehicle's details page.		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view the vehicle's maintenance logs.		
<b>Postconditions:</b>	None, read only function		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details page. 6. The user selects the maintenance log tab and is brought to a list of completed work orders. 7. When done the user clicks the Close button. 8. The Details page closes and returns to the Pending work page.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	6a. If the database returns nothing or is unreachable 1. No log information will display. 2. An alternative error message will be displayed. 3. Display the list as empty		
<b>Includes:</b>	User Login View Vehicle Work Orders		
<b>Frequency of Use:</b>	5 times per day.		
<b>Special Requirements:</b>	A refresh button Vehicles can have really extensive log histories. These should only be fetched in limited quantities.		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	Maintenance log refers to the work order history, and returns all work orders and all versions of them in chronological order.		

## Create Maintenance Log

<b>Use Case ID:</b>	UC-MTNC.03.03		
<b>Use Case Name:</b>	Create Maintenance Log		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/8/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	Marks work orders as completed		
<b>Trigger:</b>	The user has completed maintenance on a vehicle and needs to report a completed task		
<b>Preconditions:</b>	1. The user is registered in the system. 2. The user has sufficient application rights to view and edit the vehicle's maintenance logs.		
<b>Postconditions:</b>	1. The work order is completed and is removed from pending list. 2. If all work orders are complete, remove the vehicle from the maintenance list.		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details page 6. The user selects the work orders tab and is brought to a list of pending Work Orders. 7. The user selects the Complete button on a specific Work Order 8. The Complete Work Order page opens where completion details are entered. 9. The user enters the information 10. The user clicks the Confirm Completion button. 11. The Complete Work Order page closes.		
<b>Alternative Flows:</b>	7a. If the user needs to record inventory used during work. 1. Refer to UC-MTNC.04.01 Deduct Inventory. Can be called multiple times for every variety of item.		
<b>Exceptions:</b>	7a/8a If the database is unable to retrieve the order. 1. Create a warning box about the issue, 2. Return user to step 6. 8b. If the work order has been updated since the user began typing information 1. Before saving, show the user a maintenance log of the most recent update. 2. Ask the user if they wish to still save. 3a. If no, return to step 6. 3b. If yes, save data and return to step 6. 8c. If the order they are attempting to save has already been completed 1. Inform the user the order has already been resolved. 2. return to step 6.		
<b>Includes:</b>	User Login View Vehicle Work Orders Complete Work Orders View Maintenance Logs		
<b>Frequency of Use:</b>	2-10 times a day		
<b>Special Requirements:</b>			

<b>Assumptions:</b>	
<b>Notes and Issues:</b>	This entire system is prone to merge conflicts and need to be resolved.

## View Maintenance Log Details

<b>Use Case ID:</b>	UC-MNTC.04.03	
<b>Use Case Name:</b>	View Maintenance Log Details	
<b>Created By:</b>	Nik Bell	<b>Last Updated By:</b>
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager	
<b>Description:</b>	Get a detailed breakdown of a maintenance log/previous work order	
<b>Trigger:</b>	Select a maintenance log	
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view the vehicle's maintenance logs.	
<b>Postconditions:</b>	None, Read only	
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details UC-MTNC.01.02 6. The user selects the maintenance log tab and is brought to a list of previous work orders. 7. The user clicks on the log and is given a detailed breakdown of the work order	
<b>Alternative Flows:</b>		
<b>Exceptions:</b>	6a/7a. If the database returns nothing or is unreachable 1. No log information will display. 2. An alternative error message will be displayed. 3. Display the list as empty	
<b>Includes:</b>	User Login View Vehicle Work Orders View Maintenance Logs	
<b>Frequency of Use:</b>	5 times per day.	
<b>Special Requirements:</b>		
<b>Assumptions:</b>		
<b>Notes and Issues:</b>		

## Create Compatible Part

<b>Use Case ID:</b>	UC-05.01		
<b>Use Case Name:</b>	Create Compatible Part		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/01/2023	<b>Last Revision Date:</b>	10/9/2023
<b>Actors:</b>	Mechanic Fleet Manager		
<b>Description:</b>	Creating a part in the list of parts that are compatible across multiple vehicles.		
<b>Trigger:</b>	Part is added to compatible parts list		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to create new part that can be added to the list.		
<b>Postconditions:</b>	1. Part is added to list of vehicles it is compatible with.		
<b>Normal Flow:</b>	1. User opens compatible parts list. 2. User selects [Create Part] 3. User enters information for parts (including vehicles it is compatible with). 4. User selects [Save] 5. System prompts to confirm the user's selection. 6. Save part to list.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	User Login Compatible Parts List Vehicle List Parts List		
<b>Frequency of Use:</b>	On Demand.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Read Compatible Parts List

<b>Use Case ID:</b>	UC-05.01		
<b>Use Case Name:</b>	Create Compatible Part		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/01/2023	<b>Last Revision Date:</b>	10/9/2023
<b>Actors:</b>	Mechanic Fleet Manager		
<b>Description:</b>	Creating a part in the list of parts that are compatible across multiple vehicles.		
<b>Trigger:</b>	Part is added to compatible parts list		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to create new part that can be added to the list.		
<b>Postconditions:</b>	2. Part is added to list of vehicles it is compatible with.		
<b>Normal Flow:</b>	7. User opens compatible parts list. 8. User selects [Create Part] 9. User enters information for parts (including vehicles it is compatible with). 10. User selects [Save] 11. System prompts to confirm the user's selection. 12. Save part to list.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	User Login Compatible Parts List Vehicle List Parts List		
<b>Frequency of Use:</b>	On Demand.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Read Compatible Parts List

<b>Use Case ID:</b>	UC-05.02		
<b>Use Case Name:</b>	Read Compatible Parts List		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/01/2023	<b>Last Revision Date:</b>	10/9/2023
<b>Actors:</b>	Mechanic Fleet Manager		
<b>Description:</b>	Users can view the list of compatible parts for their vehicles.		
<b>Trigger:</b>	Users wanting to view compatible parts for vehicles.		
<b>Preconditions:</b>	1. The user is registered in the system.		
<b>Postconditions:</b>	1. User can view compatible parts list.		
<b>Normal Flow:</b>	1. User opens information for the vehicle. 2. User selects [Compatible Parts]. 3. Users see a list of parts compatible with the vehicle. 4. User can choose options [Create Part], [Update Part], and [Delete Part] 5. User closes list.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	User Login Compatible Parts List Vehicle List Parts List		
<b>Frequency of Use:</b>	On Demand.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Update Compatible Part

<b>Use Case ID:</b>	UC-5.03		
<b>Use Case Name:</b>	Update Compatible Part		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/01/2023	<b>Last Revision Date:</b>	10/9/2023
<b>Actors:</b>	Mechanic Fleet Manager		
<b>Description:</b>	Updating the information about a part and its compatibility.		
<b>Trigger:</b>	Compatible Part needs updating.		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to update a part in the compatibility list.		
<b>Postconditions:</b>	1. User updated information for parts compatibility.		
<b>Normal Flow:</b>	1. User opens a compatible parts list. 2. User selects [Update Part] 3. Users select what information they want to update. 4. User selects [Save] 5. System prompts to confirm the user's selection. 6. Update part in list.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	User Login Compatible Parts List Vehicle List Parts List		
<b>Frequency of Use:</b>	On Demand.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Delete Compatible Part

<b>Use Case ID:</b>	UC-05.04		
<b>Use Case Name:</b>	Delete Compatible Part		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/01/2023	<b>Last Revision Date:</b>	10/9/2023
<b>Actors:</b>	Mechanic Fleet Manager		
<b>Description:</b>	Deleting part in the compatible part list.		
<b>Trigger:</b>	Part is deleted from the compatible parts list.		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to delete a part from the compatible list.		
<b>Postconditions:</b>	1.The user will delete a part for the compatible parts list.		
<b>Normal Flow:</b>	1. User opens a compatible parts list. 2. User selects [Delete Part] 3. User selects the part to delete. 4. User selects [Save] 5. System prompts to confirm the user's selection. 6. Part is deleted from the list.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	User Login Compatible Parts List Vehicle List Parts List		
<b>Frequency of Use:</b>	On Demand.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Mark Used Inventory

<b>Use Case ID:</b>	UC-MTNC.06.01		
<b>Use Case Name:</b>	Mark Used Inventory		
<b>Created By:</b>	Nik Bell	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	9/22/2023	<b>Last Revision Date:</b>	10/9/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	Marked inventory as used over the course of maintenance.		
<b>Trigger:</b>	Inventory was used over the course of maintenance		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view and edit the vehicle's maintenance logs.		
<b>Postconditions:</b>	The relative inventory quantity is deducted		
<b>Normal Flow:</b>	1. An optional module to be attached to UC-MTNC.03.02 or UC-MTNC.03.03 2. User selects add used inventory on work order 3. User selects the type of item and quantity used 4. Repeat step 3 until all items are added 5. Save the work order		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	5a/5b If the database or inventory system is unreachable 1. Display an error box of the issue. 2. Return the user to step 4 to keep their progress in memory.		
<b>Includes:</b>	User Login View Vehicle Work Orders View Maintenance Logs Inventory System		
<b>Frequency of Use:</b>	2-10 times a day		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## View Scheduled Maintenance List

<b>Use Case ID:</b>	UC-MNTC.07.01		
<b>Use Case Name:</b>	View Scheduled Maintenance List		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/23/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	View a list of vehicles with scheduled maintenance		
<b>Trigger:</b>			
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view the vehicle's details.		
<b>Postconditions:</b>			
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up.		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the user has an incorrect login 1. The login is rejected. 2. Display an error message. 3. Use Case resumes on step 2 of normal flow.		
<b>Includes:</b>	User Login		
<b>Frequency of Use:</b>	50 times per day.		
<b>Special Requirements:</b>	A manual refresh button		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	This should only return a list of vehicles with pending work orders. We need a stored procedure for that. Another filter that's able to limit the selections to cars in the garage would also be very useful.		

## Create Scheduled Maintenance

<b>Use Case ID:</b>	UC-MTNC.07.02		
<b>Use Case Name:</b>	Create Scheduled Maintenance		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	Nik Bell
<b>Date Created:</b>	10/23/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Any staff member		
<b>Description:</b>	Creates a work order which reports an issue or recurring checkup required for the vehicle.		
<b>Trigger:</b>	Staff members who have identified an issue with the vehicles. Automated recurring maintenance system.		
<b>Preconditions:</b>			
<b>Postconditions:</b>	A new active work order is attached to the vehicle.		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details page. 6. The user selects the work orders tab and is brought to a list of work orders. 7. The user selects and adds a new work order. 8. The Create Work Order page opens up. 9. The user enters the work order information. 10. The user clicks the Add Work Order button and is brought back to step 6.		
<b>Alternative Flows:</b>	8a. If the user decides that they don't need to add a work order 1. The user clicks the Cancel button. 2. The Create Work Order page closes, discarding any entered information.. 3. The flow resumes at step 6.		
<b>Exceptions:</b>	Database not found		
<b>Includes:</b>	User Login View Vehicle Work Orders		
<b>Frequency of Use:</b>	0 to 10 times a day		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	This use case is probably going to need to be accessible from so many different employee UIs. Generic issues about the car can also be reported through this system. If the issue doesn't really exist, the order can just be completed.		

## Update Scheduled Maintenance

<b>Use Case ID:</b>	UC-MTNC.07.03		
<b>Use Case Name:</b>	Update Work Orders		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	
<b>Date Created:</b>	10/23/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	The user can update a work order clarifying that some work has been done but the issue remains unresolved, or the issue has been properly diagnosed in case of an unclear problem was submitted as a work order.		
<b>Trigger:</b>	The update button is clicked on the Work Orders tab of the details page		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view and edit the vehicle's maintenance logs.		
<b>Postconditions:</b>	1. A new work order is created that acts as a substitute for the previous order. Accessing the order should now return the more up to date version. 2. Archive the old work order (UC-MTNC.07.02)		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details UC-MTNC.01.02 6. The user selects the work orders tab and is brought to a list of orders. 7. The user selects updates for a specific order and is brought to a UI where details are entered. 8. The user saves the order and is brought back to step 6.		
<b>Alternative Flows</b>	7a. If the user needs to record inventory used during work. 1. Refer to UC-MTNC.04.01 Deduct Inventory. Can be called multiple times for every variety of item.		
<b>Exceptions:</b>	7a/8a If the database is unable to retrieve the order. 1. Create a warning box about the issue, 2. Return user to step 6. 8b. If the work order has been updated since the user began typing information 1. Before saving, show the user a maintenance log of the most recent update. 2. Ask the user if they wish to still save. 3a. If no, return to step 6. 3b. If yes, save data and return to step 6. 8c. If the order they are attempting to save has already been completed 1. Inform the user the order has already been resolved. 2. return to step 6.		
<b>Includes:</b>	User Login View Vehicle Work Orders View Maintenance Logs		
<b>Frequency of Use:</b>	10 times a day.		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			

<b>Notes and Issues:</b>	This should not overwrite the previous work order. Instead this should create a new order entirely, we will need to create pointers and a system to mark the new order as the up to date version.
--------------------------	---

## Complete Scheduled Maintenance

<b>Use Case ID:</b>	UC-MTNC.07.04		
<b>Use Case Name:</b>	Complete Scheduled Maintenance		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	
<b>Date Created:</b>	10/23/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	Marks scheduled maintenance as completed		
<b>Trigger:</b>	The user has completed maintenance on a vehicle and needs to report a completed task		
<b>Preconditions:</b>	1. The user is registered in the system. 2. The user has sufficient application rights to view and edit the vehicle's maintenance logs.		
<b>Postconditions:</b>	1. The work order is completed and is removed from a vehicle. 2. If all work orders are complete, remove the vehicle from the maintenance list. 3. Create Maintenance Log UC-MNTC-03.04		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details UC-MTNC.01.02 6. The user selects the work orders tab and is brought to a list of pending Work Orders. 7. The user selects the Complete button on a specific Work Order and is brought to a UI where completion details are entered. 8. The user enters the information and and is brought back to step 6.		
<b>Alternative Flows:</b>	7a. If the user needs to record inventory used during work. 1. Refer to UC-MTNC.04.01 Deduct Inventory. Can be called multiple times for every variety of item.		
<b>Exceptions:</b>	7a/8a If the database is unable to retrieve the order. 1. Create a warning box about the issue, 2. Return user to step 6. 8b. If the work order has been updated since the user began typing information 1. Before saving, show the user a maintenance log of the most recent update. 2. Ask the user if they wish to still save. 3a. If no, return to step 6. 3b. If yes, save data and return to step 6. 8c. If the order they are attempting to save has already been completed 1. Inform the user the order has already been resolved. 2. return to step 6.		
<b>Includes:</b>	User Login View Vehicle Work Orders View Maintenance Logs Create Maintenance Logs		
<b>Frequency of Use:</b>	2-10 times a day		
<b>Special Requirements:</b>			

<b>Assumptions:</b>	
<b>Notes and Issues:</b>	This entire system is prone to merge conflicts and need to be resolved.

## Delete Scheduled Maintenance

<b>Use Case ID:</b>	UC-MTNC.07.05	
<b>Use Case Name:</b>	Delete Scheduled Maintenance	
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>
<b>Date Created:</b>	10/23/2023	<b>Last Revision Date:</b>
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager	
<b>Description:</b>	Deletes scheduled maintenance	
<b>Trigger:</b>	The user attempts to delete an existing scheduled maintenance	
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view and edit the vehicle's maintenance logs.	
<b>Postconditions:</b>	1. The scheduled maintenance is removed from the system	
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. A page of vehicles with pending maintenance(work orders) shows up. 5. The user clicks on a vehicle and is brought to the vehicle details UC-MTNC.01.02 6. The user selects the work orders tab and is brought to a list of pending Work Orders. 7. The user selects the Delete button on a specific scheduled maintenance and is given a confirmation pop-up. 8. The deletion is saved and the user is brought back to step 6.	
<b>Alternative Flows:</b>		
<b>Exceptions:</b>	7a/8a If the database is unable to retrieve the order. 1. Create a warning box about the issue, 2. Return user to step 6. 8b. If the work order has been updated since the user began typing information 1. Before saving, show the user a maintenance log of the most recent update. 2. Ask the user if they wish to still save. 3a. If no, return to step 6. 3b. If yes, save data and return to step 6. 8c. If the order they are attempting to save has already been completed 1. Inform the user the order has already been resolved. 2. return to step 6.	
<b>Includes:</b>	User Login View Vehicle Work Orders View Maintenance Logs Create Maintenance Logs	
<b>Frequency of Use:</b>	2-10 times a day	
<b>Special Requirements:</b>		
<b>Assumptions:</b>		
<b>Notes and Issues:</b>	This entire system is prone to merge conflicts and need to be resolved.	

# Employee – Mechanic

## View Reports of All Vehicles

<b>Use Case ID:</b>	12.6		
<b>Use Case Name:</b>	View Reports of All Vehicles		
<b>Created By:</b>	Steven Sanchez	<b>Last Updated By:</b>	
<b>Date Created:</b>	9/3/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Mechanic		
<b>Description:</b>	A mechanic should be able to access a well-informed status summary of the entire fleet so they can select vehicles to prioritize maintenance and repairs.		
<b>Trigger:</b>	An authorized mechanic requests a summary report for all vehicles.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The mechanic must have authorization.</li> <li>2. The system should have a database of all vehicles including their maintenance and performance records.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The mechanic is presented with a comprehensive summary report for all vehicles.</li> <li>2. The mechanic can make informed decisions on maintenance and repairs based on summary.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Mechanic logs into fleet management system.</li> <li>2. The mechanic navigates to “summary report” or something similar.</li> <li>3. System retrieves data and displays reports for mechanic.</li> <li>4. The mechanic reviews the report and identifies a vehicle that requires attention.</li> <li>5. Mechanic proceeds to take necessary action to ensure the vehicle of importance gets taken care of.</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> <li>2a. in step 2 of the normal flow, if a mechanic wants to view a more detailed report on a specific vehicle.             <ol style="list-style-type: none"> <li>1. The mechanic should be allowed to click on a vehicle from the list.</li> <li>2. The system displays the data.</li> </ol> </li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1a. in step 1 of the normal flow, if mechanic does not have authorization.</li> <li>1. the systems alerts mechanic to seek a supervisor or administrator for assistance.</li> </ol>		
<b>Includes:</b>	Work orders		
<b>Frequency of Use:</b>	On demand.		
<b>Special Requirements:</b>	Authorization to view sensitive data.		
<b>Assumptions:</b>	The data stored in the system is accurate and up to date.		
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. May need to consider allowing the mechanic to make changes on the report and then save to system</li> </ol>		

## Create a Work Order

<b>Use Case ID:</b>	12.7		
<b>Use Case Name:</b>	Create a Work Order		
<b>Created By:</b>	steven Sanchez	<b>Last Updated By:</b>	Jonathan Beck
<b>Date Created:</b>	9/3/2023	<b>Last Revision Date:</b>	9/17
<b>Actors:</b>	Mechanic		
<b>Description:</b>	A mechanic should be able to create a new work order so that they can assign a vehicle to a mechanic that needs repairing or maintenance		
<b>Trigger:</b>	A mechanic identifies a vehicle that needs services		
<b>Preconditions:</b>	1. The mechanic is logged into the fleet management system. 2. The system has a record of the vehicle. 3. The mechanic has the necessary information about the vehicle.		
<b>Postconditions:</b>	1. A new work order is created. 2. System saves work order in the database. 3. The work order has a unique identifier. 4. The mechanic can review, edit, and assign the work order to a mechanic.		
<b>Normal Flow:</b>	1. The mechanic navigates to "Create Work Order" from the mechanics center of the app or something similar. 2. 3. The mechanic selects a vehicle that needs maintenance or repairs. 4. The mechanic provides details about the request including description of issue, required tasks, priority level and any additional instructions. 5. The system generates the new work order and assigns a unique identifier. 6. The System saves the order and updates its status to "pending". 7. The mechanic then assigns the work order to a mechanic that can complete the order.		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	4a. In step 4 of the normal flow, if mechanic needs to edit work order, 1.the system should display a message to confirm work order. 2.if mechanic needs to make changes the system should have an edit option. 3. mechanic edit changes and check if everything is correct. 4. Use case continues to step 5 of the normal flow.		
<b>Exceptions:</b>	1a. in step 1 of the normal flow if mechanic does not have authorization. 1. System alerts user to contact supervisor or administrator for assistance.		
<b>Includes:</b>	Append to Work Order		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	Authorization level		
<b>Assumptions:</b>	The mechanic has a good understanding of the maintenance or repair required for the vehicle.		
<b>Notes and Issues:</b>	1. will need to add status reports for each vehicle undergoing maintenance/repairs.		

## Enter Service Records

<b>Use Case ID:</b>	12.9		
<b>Use Case Name:</b>	Enter Service Records		
<b>Created By:</b>	Steven Sanchez	<b>Last Updated By:</b>	Jonathan Beck
<b>Date Created:</b>	9/4/2023	<b>Last Revision Date:</b>	9/17
<b>Actors:</b>	Mechanic		
<b>Description:</b>	An authorized mechanic will need to flag vehicles either available (ready to use in the fleet) or unavailable (needs repairs or maintenance)		
<b>Trigger:</b>	<ol style="list-style-type: none"> <li>1. When a vehicle needs to be put in the fleet.</li> <li>2. A mechanic needs to document current status of vehicle.</li> </ol>		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The vehicle that is being serviced needs to be registered in the system.</li> <li>2. Maintenance or repairs need to be completed if available.</li> <li>3. Vehicles need to be evaluated if it is unavailable.</li> <li>4. The mechanic should have access to service records and details.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Service records are entered into the system.</li> <li>2. The system updates the status of the vehicle.</li> <li>3. The vehicles maintenance history is updated.</li> <li>4. The mechanic can retrieve and review the updated service record.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Mechanic logs into the fleet management system.</li> <li>2. User navigates to “service records” or something similar.</li> <li>3. Vehicles are sorted / filtered by various criteria</li> <li>4. User selects vehicle that needs service record entered.</li> <li>5. User provides details about the service.</li> <li>6. User selects to flag the vehicle as “available” or “unavailable”.</li> <li>7. User reviews before proceeding with services (if needed).</li> <li>8. System saves service record.</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<p>6a. in step 6 of the normal flow, if User finds an error entered into the records</p> <ol style="list-style-type: none"> <li>1. System provides an editing option.</li> <li>2. User corrects any errors.</li> <li>3. User reviews the record thoroughly.</li> <li>4. Use case Continues on step 7 of the normal flow.</li> </ol>		
<b>Exceptions:</b>	<p>5a. In step 4 of the normal flow if user fails to provide all necessary details of the service record.</p> <ol style="list-style-type: none"> <li>1. System highlights all missing fields that need data entered.</li> <li>2. User provides data for all missing fields</li> <li>3. Use case continues to step 5 of the normal flow.</li> </ol> <p>3a. In step 3 of the normal flow, there are no mechanics that match this criteria.</p> <ol style="list-style-type: none"> <li>1. System will state that there are no such mechanics</li> </ol>		
<b>Includes:</b>	Creating work order, append to work order.		
<b>Frequency of Use:</b>	On demand.		
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. Mechanic needs authorization.</li> <li>2. Maintenance records needed.</li> </ol>		
<b>Assumptions:</b>	There's a head mechanic that assigns services to other mechanics.		
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. To limit confusion with the system only a few authorized personnel should be allowed to enter service records.</li> </ol>		

## View Maintenance Records

<b>Use Case ID:</b>	12.10		
<b>Use Case Name:</b>	View maintenance Records - by worker		
<b>Created By:</b>	Steven Sanchez	<b>Last Updated By:</b>	Jonathan Beck
<b>Date Created:</b>	9/4/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Mechanic		
<b>Description:</b>	A supervising mechanic will need to review maintenance record so they can evaluate specific technicians or mechanics performance		
<b>Trigger:</b>	When a supervisor or administrator needs to assess the performance and work history of a specific technician or mechanic.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>System needs a comprehensive database of maintenance records including repairs.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>The supervising mechanic can access, and review maintenance records associated with technician or mechanic.</li> <li>The user can assess the specific technician or mechanics performance, work history, and the quality of completed maintenance orders.</li> <li>The User can use this information for evaluation, feedback, training, or performance improvement discussions.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>The User is logged into the system.</li> <li>User navigates to “view maintenance records” or something similar.</li> <li>various filters are selected</li> <li>The user selects a worker that they want to review performance for.</li> <li>The system retrieves and display the data for the user.</li> <li>User can filter specific maintenance records or repairs done by the selected worker.</li> <li>User reviews records.</li> <li>User makes assessments based on the provided records.</li> <li>Based on evaluation, user can provide feedback, initiate training, or make decisions regarding the technician or mechanic.</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<p>5a. In step 5 of the normal flow, if user wants to narrow down the list of maintenance.</p> <ol style="list-style-type: none"> <li>User selects date maintenance was completed.</li> <li>User selects Type of vehicle.</li> <li>system refreshes list.</li> <li>Use case continues to step 6 of the normal flow.</li> </ol>		
<b>Exceptions:</b>	<p>5a. In step 4 of the normal flow, If system fails to retrieve data.</p> <ol style="list-style-type: none"> <li>system alerts user there was an error retrieving data.</li> <li>system advises user to try again.</li> <li>If works correctly use case continues to step 5 of normal flow.</li> </ol> <p>5b. if trying to retrieve data fails.</p> <ol style="list-style-type: none"> <li>system alerts user</li> <li>use case ends and sends user to home screen.</li> </ol> <p>3a. In step 3 of the normal flow, there are no mechanics that match this criteria.</p> <ol style="list-style-type: none"> <li>System will state that there are no such mechanics</li> </ol>		
<b>Includes:</b>	Work order database		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	Detailed maintenance records.		
<b>Assumptions:</b>	User has knowledge of maintenance work and can make informed assessments.		
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>May want to consider adding a trainer use case.</li> </ol>		



## In-Service Alert

<b>Use Case ID:</b>	12.4		
<b>Use Case Name:</b>	In-Service Alert		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Jonathan Beck
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/18/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	A toggle for alerting the software that a particular vehicle is in service.		
<b>Trigger:</b>	Toggling the In-Service selector in the Maintenance Log Page.		
<b>Preconditions:</b>	1. Open up the fleet management software. 2. Navigate to the vehicle or garage page in the software. 3. Select the Maintenance Log button on a vehicle to open its log page. 4. The Maintenance Log page opens up with an In-Service toggle switch in the top left of the log page. 5. The user switches the In-Service toggle to change the vehicle's status.		
<b>Postconditions:</b>	1. The software updates the vehicle's status.		
<b>Normal Flow:</b>	1. The user opens the fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. The user's landing page opens. 5. The user selects the garage button from their list. 6. The garage page opens up. 7. The user scrolls to find a vehicle and clicks the Maintenance Log button. 8. The vehicle's log page opens with an In-Service toggle switch in the top left of the page. 9. The user switches the In-Service toggle to change the vehicle's status as needed 10. The user inputs a note if appropriate. 11. After the user is done they click the save & close button. 12. The maintenance log page saves, closes, and returns to the garage page.		
<b>Alternative Flows:</b>	7a. In step 7 of the normal flow, if the selected vehicle doesn't have a log 1. A new maintenance log page will need to be written into memory. 2. The vehicle's new log page opens up displaying a blank maintenance log. 3. The user clicks on the Work Orders tab. 4. The Work Orders tab displays a first-time maintenance work order. 3. Use Case resumes at step 11.  12a. In step 12 of the normal flow, if the maintenance log is a new entry 1. The information entered will need to be written to the database as a new entry. 2. Use Case resumes at step 12		
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the user has an incorrect login 1. The login is rejected. 2. Display an error message. 3. Use Case resumes on step 2 of normal flow.		
<b>Includes:</b>	1. User Login 2. Garage/Garage Manager 3. Maintenance Log		
<b>Frequency of Use:</b>	2 times per day.		
<b>Special Requirements:</b>	1. A vehicle information database.		
<b>Assumptions:</b>			

<b>Notes and Issues:</b>	If the vehicle has already been borrowed no maintenance can be done until after it is returned.
--------------------------	---

## View Flags

<b>Use Case ID:</b>	12.8		
<b>Use Case Name:</b>	View Flags		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Bill Tobias
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	A list of vehicle flags/alerts.		
<b>Trigger:</b>	Opening the Flags tab of a vehicle's Maintenance Log page.		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view and edit the vehicle's maintenance logs and associated information.		
<b>Postconditions:</b>			
<b>Normal Flow:</b>	1. The user opens the fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. The user's landing page opens. 5. The user selects the garage button from their list. 6. The garage page opens up. 7. The user scrolls to find a vehicle and clicks the Maintenance Log button. 8. The vehicle's log page opens up displaying the maintenance log with Work Orders and Flags tabs at the top. 9. The user clicks on the Flags tab. 10. The Flags tab opens displaying all the vehicle's alert flags and their descriptions. 11. After the user is done they click the save & close button. 12. The maintenance log page saves, closes, and returns to the garage page.		
<b>Alternative Flows:</b>	7a. In step 7 of the normal flow, if the selected vehicle doesn't have a log 1. A new maintenance log page will need to be written into memory. 2. The vehicle's new log page opens up displaying a blank maintenance log. 3. The user clicks on the Work Orders tab. 4. The Work Orders tab displays a first-time maintenance work order. 3. Use Case resumes at step 11.  11a. In step 11 of the normal flow, if the maintenance log is a new entry 1. The information entered will need to be written to the database as a new entry. 2. Use Case resumes at step 12.  11b. In step 11, if the user has made changes to the information on the form and then elects to close the form rather than save, the user is prompted to save or abandon the information changes made. Once the selection is made, the user returns to the Garage page.		
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the user has an incorrect login 1. The login is rejected. 2. Display an error message. 3. Use Case resumes on step 2 of normal flow.		
<b>Includes:</b>	1. User Login 2. Garage/Garage Manager 3. Maintenance Log		

<b>Frequency of Use:</b>	4 times per day.
<b>Special Requirements:</b>	1. A vehicle information database.
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

## Maintenance List

<b>Use Case ID:</b>	12.11		
<b>Use Case Name:</b>	View Maintenance List		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Bill Tobias
<b>Date Created:</b>	9/10/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	View a list of vehicles that need maintenance.		
<b>Trigger:</b>	Clicking the Maintenance List button on the Garage page.		
<b>Preconditions:</b>	1. The user is registered in the system.  2. The user has sufficient application rights to view the maintenance list.		
<b>Postconditions:</b>	None		
<b>Normal Flow:</b>	1. The user opens the fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. The user's landing page opens. 5. The user selects the garage button from their list. 6. The garage page opens up. 7. The user clicks the Maintenance List button at the top of the Garage page. 8. The Maintenance List page opens up with a list of vehicles that have work orders or are flagged for service. 9. The user clicks on the close button. 10. The Maintenance List page closes and returns to the Garage page.		
<b>Alternative Flows:</b>	8a. In step 8 of the normal flow, if no vehicles have work orders or are flagged for service 1. A "No maintenance is required" message will be displayed in the Maintenance List box. 2. Use Case resumes at step 9.		
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the user has an incorrect login 1. The login is rejected. 2. Display an error message. 3. Use Case resumes on step 2 of normal flow.		
<b>Includes:</b>	1. User Login 2. Garage/Garage Manager 3. View Work Orders		
<b>Frequency of Use:</b>	10 times per day.		
<b>Special Requirements:</b>	1. A vehicle information database.		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Maintenance Log

<b>Use Case ID:</b>	12.13		
<b>Use Case Name:</b>	Log daily maintenance inventory		
<b>Created By:</b>	Ben Collins	<b>Last Updated By:</b>	Bill Tobias
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Mechanic Fleet Manager Operations Manager		
<b>Description:</b>	View detailed logs of all maintenance for each vehicle.		
<b>Trigger:</b>	Selecting the Maintenance Log button on any vehicle.		
<b>Preconditions:</b>	1.The user is registered in the system. 2.The user has sufficient application rights to view and edit the vehicle's maintenance logs.		
<b>Postconditions:</b>	1. The user log page opens up displaying all maintenance records. 2. The user is able to view and update the maintenance log.		
<b>Normal Flow:</b>	1. User opens fleet management software. 2. The software opens up to the login page. 3. The user enters their username, password, and clicks the login button. 4. The user's landing page opens. 5. The user selects the supplies button from their list. 6. The supplies page opens up. 7. The user scrolls to find the category of item they need to update 8. Current quantity on hand is displayed 9. The user edits the amount to bring it up to date. 10. The user clicks the save & close button. 11. The update quantity log page closes and returns to the garage page.		
<b>Alternative Flows:</b>	7a. In step 7 of the normal flow, if the selected vehicle doesn't have a log 1. A new maintenance log page will need to be written into memory. 2. The vehicle's new log page opens up displaying a blank maintenance log. 3. Use Case resumes at step 9.  9a. In step 9 of the normal flow, if the user doesn't edit the quantity 1. The page will display only a Close button. 2. User clicks the Close button. 3. Use Case resumes at step 11.  10a. In step 10 of the normal flow, if the maintenance log is a new entry 1. The information entered will need to be written to the database as a new entry. 2. Use Case resumes at step 11.  11a. In step 11, if the user has made changes to the information on the form and then elects to close the form rather than save, the user is prompted to save or abandon the information changes made. Once the selection is made, the user returns to the Garage page.		
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the user has an incorrect login 1. The login is rejected. 2. Display an error message. 3. Use Case resumes on step 2 of normal flow.		
<b>Includes:</b>	1. User Login 2. Garage/Garage Manager		

<b>Frequency of Use:</b>	75 times per day.
<b>Special Requirements:</b>	1. A vehicle information database.
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

# Employee – Parts Person

## Confirm Part Requests

<b>Use Case ID:</b>	UC-PARTS.01.01		
<b>Use Case Name:</b>	Confirm Part Requests		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Primary: Parts Person Secondary: Mechanic		
<b>Description:</b>	With mechanics being able to submit parts requests, the parts person must approve or reject them and alert the mechanics on the requests status.		
<b>Trigger:</b>	The parts person will click a button stating "Approve".		
<b>Preconditions:</b>	1. mechanic creates a part request 2. mechanic sends part request to parts person 3. parts person reviews the request		
<b>Postconditions:</b>	1. Request is removed from the list of requests. 2. Automated message is sent to the mechanic letting them know the request has been approved. 3. Part or parts in the request are added to the parts order 'cart'		
<b>Normal Flow:</b>	1. Parts person reviews the details of the request 2. The parts person clicks an approval button 3. A message prompts the parts person to confirm approval of the request 4. A flag is sent to the mechanic alerting them to the approval 5. The part(s) in the request are added to an order 6. The request is removed from the approval pending list of requests 7. The request is archived		
<b>Alternative Flows:</b>	3a. 1. The parts person declines the confirmation prompt 2. Use Case resumes on step 1 of the normal flow		
<b>Exceptions:</b>			
<b>Includes:</b>	UC-PARTS.01.02 - View Part Requests UC-PARTS.02.01 View Part Options UC-PARTS.01.03 Sort Part Requests UC-PARTS.01.04 Reject Part Request		
<b>Frequency of Use:</b>	4 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands english or spanish The part requested is needed for maintenance of a vehicle		
<b>Notes and Issues:</b>	1. Exceptions TBD 2. Includes TBD 3. Special Requirements TBD 4. There may be more alternative flows necessary for this use case		

## View Part Requests

<b>Use Case ID:</b>	UC-PARTS.01.02		
<b>Use Case Name:</b>	View Part Requests		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Parts Person Mechanics		
<b>Description:</b>	To begin the process of figuring out which parts to order, the parts person will need to be able to see the current parts requests		
<b>Trigger:</b>	The parts person clicks a button that states 'Part Requests'		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Mechanic creates a part request</li> <li>2. Mechanic sends part request to parts person</li> </ol>		
<b>Postconditions:</b>			
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person clicks to view the requests</li> <li>2. A window appears with the requests, each described by some basic information</li> <li>3. The parts person is able to go on to do other tasks with this information</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	2-8 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands english or spanish		
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Postconditions</li> <li>2. Alternate flows</li> <li>3. Exceptions</li> <li>4. Special Requirements</li> </ol>		

## Sort Part Requests

<b>Use Case ID:</b>	UC-PARTS.01.03		
<b>Use Case Name:</b>	Sort Part Requests		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/9/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Parts Person Mechanics		
<b>Description:</b>	In the case of many requests with a variety of vehicle manufacturers, maintenance needed, and many other differences, the parts person will need to be able to narrow down the number of requests that are displayed, to be able to more quickly determine what they will order on a given day or week.		
<b>Trigger:</b>	The parts person clicks a button to choose a field to sort by, then clicks a button to confirm		
<b>Preconditions:</b>	1. Mechanic creates a part request 2. Mechanic sends part request to parts person 3. Parts person views the request list		
<b>Postconditions:</b>	1. The part request list is sorted by the field selected		
<b>Normal Flow:</b>	1. The parts person selects a field to sort by from a drop down list 2. The parts person selects a confirmation button to sort the list 3. The list is reloaded sorted to the chosen field		
<b>Alternative Flows:</b>	2a. The parts person decides they do not need to sort the list, and clicks away from the sort drop down 1. The drop down field disappears, leaving the list unaffected		
<b>Exceptions:</b>			
<b>Includes:</b>	UC-PARTS.01.02 View Part Requests		
<b>Frequency of Use:</b>	2-6 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>	Exceptions TBD Includes TBD Special Requirements TBD		

## Reject Part Requests

<b>Use Case ID:</b>	UC-PARTS.01.04		
<b>Use Case Name:</b>	<b>Reject Part Requests</b>		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Parts Person Mechanics		
<b>Description:</b>	There will be times that the requested part is not necessary, or the mechanic finds an alternative solution, so the parts person should be able to reject parts requests. Both for record-keeping and communication with the mechanic, the parts person will attach a note with a reason why the request was rejected.		
<b>Trigger:</b>	The parts person clicks a reject button to reject the mechanics part request		
<b>Preconditions:</b>	1. Mechanic creates a part request 2. Mechanic sends part request to parts person 3. Parts person views the request details		
<b>Postconditions:</b>	1. The request is removed from the list 2. The request is archived 3. A message is sent to the mechanic that submitted the part request with the attached reason for the rejection 4. The window returns to the request list		
<b>Normal Flow:</b>	1. The parts person views the list of requests 2. The parts person clicks a button to reject the parts request 3. A window appears with a dropdown list of reasons to reject the request 4. Upon selecting a reason or typing a custom one, the parts person can confirm the rejection 5. The window returns to the main list of parts requests, minus the rejected request		
<b>Alternative Flows:</b>	2a. the parts person mistakenly clicked to reject the request 1. the parts person hits a cancel button 2. return to step 1 of the normal flow		
<b>Exceptions:</b>			
<b>Includes:</b>	UC-PARTS.01.02 View Part Requests		
<b>Frequency of Use:</b>	1 time per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands english or spanish		
<b>Notes and Issues:</b>	Exceptions TBD Includes TBD Special Requirements TBD		

## View Part Request Details

<b>Use Case ID:</b>	UC-PARTS.01.05	
<b>Use Case Name:</b>	View Part Request Details	
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>
<b>Actors:</b>	Parts Person Mechanics	
<b>Description:</b>	To begin the process of figuring out which parts to order, the parts person will need to be able to see the details of specific parts requests	
<b>Trigger:</b>	The parts person clicks on a request after UC-PARTS.01.02	
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Mechanic creates a part request</li> <li>2. Mechanic sends part request to parts person</li> </ol>	
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The detailed information for one part request are displayed</li> </ol>	
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person clicks to view a part request's details</li> <li>2. A window appears with the request's specific information</li> <li>3. The parts person is able to go on to do other tasks with this information</li> </ol>	
<b>Alternative Flows:</b>		
<b>Exceptions:</b>		
<b>Includes:</b>	UC-PARTS.01.02 View Part Requests	
<b>Frequency of Use:</b>	2-8 times per week	
<b>Special Requirements:</b>		
<b>Assumptions:</b>	The parts person understands english or spanish	
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Postconditions</li> <li>2. Alternate flows</li> <li>3. Exceptions</li> <li>4. Special Requirements</li> </ol>	

## View Part Options

<b>Use Case ID:</b>	UC-PARTS.02.01		
<b>Use Case Name:</b>	View Part Options		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Parts Person Mechanics Suppliers		
<b>Description:</b>	After seeing what part or supplies a mechanic is requesting, it is the part person's job to find the best part for that vehicle, whether it needs to be from a specific brand, or looking at different price points.		
<b>Trigger:</b>	The parts person clicks a button attached to a request to bring up options		
<b>Preconditions:</b>	1. Mechanic creates a part request 2. Mechanic sends part request to parts person 3. Parts person views the request		
<b>Postconditions:</b>	1. A part is selected, ready to be added to an order 2. The window returns to the request details view		
<b>Normal Flow:</b>	1. The parts person clicks to view request details 2. The parts person clicks to view part options 3. A part is selected 4. The window returns to the request details		
<b>Alternative Flows:</b>	3a. An acceptable part cannot be/is not found 1. the parts person clicks a close button to exit the part selection window 2. return to step 1 of the normal use case		
<b>Exceptions:</b>			
<b>Includes:</b>	UC-PARTS.01.02 View Part Requests		
<b>Frequency of Use:</b>	2-6 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands english or spanish		
<b>Notes and Issues:</b>	1. Exceptions TBD 2. Includes TBD 3. Special Requirements TBD		

## Add Parts To Inventory

<b>Use Case ID:</b>	UC-PARTS.02.02		
<b>Use Case Name:</b>	Add Parts To Inventory		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	10/5/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Primary: Parts Person Secondary: Fleet Manager		
<b>Description:</b>	After parts are ordered and they arrive, the parts will need to be added to the on hand inventory. This will be done by inputting data from a packing		
<b>Trigger:</b>	The parts person clicks on a tab that states 'Receiving'		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. mechanic creates a part request</li> <li>2. mechanic sends part request to parts person</li> <li>3. parts person reviews the request</li> <li>4. Parts person approves parts request, and sends an order for the part(s)</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The on-hand inventory number is increased by an amount equal to the number of parts received</li> <li>2. The number of parts received, any notes about receiving, and the date received are recorded</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Parts person clicks to the Receiving tab</li> <li>2. They input the packing slip number on the packing slip that they received from the deliverer</li> <li>3. They input the part id number for the part they have received</li> <li>4. They input how many of the part have been received</li> <li>5. They click to confirm completion</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. There is a problem or something of note about receiving the parts</p> <ol style="list-style-type: none"> <li>1. The parts person enters notes about the received parts</li> <li>2. The flow continues at step 5 of the normal flow</li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	4 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Exceptions TBD</li> <li>2. Includes TBD</li> <li>3. Special Requirements TBD</li> <li>4. There may be more alternative flows necessary for this use case</li> </ol>		

## View Parts Inventory

<b>Use Case ID:</b>	UC-Parts.02.03		
<b>Use Case Name:</b>	View Part Inventory		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Everett DeVaux
<b>Date Created:</b>	10/16/2023	<b>Last Revision Date:</b>	10/23/2023
<b>Actors:</b>	Parts Person		
<b>Description:</b>	The parts person will need to be able to check what the current inventory is		
<b>Trigger:</b>	The parts person clicks a button to view inventory		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Logged in as an account allowed to view inventory</li> <li>2. Part has been added to inventory</li> <li>3. Parts person views the request</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. A list of all parts and their associated current inventory numbers are displayed</li> <li>2. The window will return to the previous screen</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person selects the inventory button</li> <li>2. The parts person views the inventory of parts</li> <li>3. The parts person closes inventory</li> <li>4. The window returns to the previous screen</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	2-6 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Exceptions TBD</li> <li>2. Includes TBD</li> <li>3. Special Requirements TBD</li> </ol>		

## Sort Part Inventory

<b>Use Case ID:</b>	UC-PARTS-02.04		
<b>Use Case Name:</b>	Sort Part Inventory		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	10/16/2023	<b>Last Revision Date:</b>	10/28/2023
<b>Actors:</b>	Parts Person		
<b>Description:</b>	In the case of many parts with a variety of manufacturers and other differences, the parts person will need to be able to narrow down the number of parts that are displayed, to be able to more quickly determine which parts are currently available		
<b>Trigger:</b>	The parts person clicks a button to choose a field to sort by, then clicks a button to confirm		
<b>Preconditions:</b>			
<b>Postconditions:</b>	1. The part inventory list is sorted by the field selected		
<b>Normal Flow:</b>	1. The parts person selects a field to sort by from a drop down list 2. The parts person selects a confirmation button to sort the list 3. The list is reloaded sorted to the chosen field		
<b>Alternative Flows:</b>	2a. The parts person decides they do not need to sort the list, and clicks away from the sort drop down 1. The drop down field disappears, leaving the list unaffected		
<b>Exceptions:</b>			
<b>Includes:</b>	UC-PARTS.02.03 View Part Inventory		
<b>Frequency of Use:</b>	2-6 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands english or spanish		
<b>Notes and Issues:</b>	Exceptions TBD Includes TBD Special Requirements TBD		

## View Part Details

<b>Use Case ID:</b>	UC-PARTS.02.05		
<b>Use Case Name:</b>	Parts Detail		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	Max Fare
<b>Date Created:</b>	09/08/2023	<b>Last Revision Date:</b>	10/30/2023
<b>Actors:</b>	Fleet Manager Parts Person		
<b>Description:</b>	Lookup detailed information about a part		
<b>Trigger:</b>	Need specific information about a part		
<b>Preconditions:</b>	1. User is authenticated		
<b>Postconditions:</b>	1. Parts detail is displayed to user		
<b>Normal Flow:</b>	1. User logs in 2. User selects inventory system 3. User searches for part 4. Part is selected from the list 5. Selects to view detailed information		
<b>Alternative Flows:</b>	5a Update the parts information (Fleet Manager ONLY) 1. User updates the part details 2. Selects to submit the changes 3. Confirm the changes to the part details		
<b>Exceptions:</b>	1.		
<b>Includes:</b>			
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	Vehicle parts list has been obtained from vehicle manufacturer and entered into system		
<b>Notes and Issues:</b>	1. Part name, vendor part number, part number(if different from vendor), part specifications, part quantity, part vendor, part order qty., Part delivery time, part photo, part description, ordered (not received qty)		

## Create Purchase Orders

<b>Use Case ID:</b>	UC-PARTS.3.1		
<b>Use Case Name:</b>	Create Purchase Orders		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Max Fare
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	10/30/2023
<b>Actors:</b>	Parts Person		
<b>Description:</b>	To order parts from a vendor, the parts person creates a purchase order		
<b>Trigger:</b>	The Parts Person needs to order new part items		
<b>Preconditions:</b>	1. Parts Person has a list of parts or items they need 2. The parts person is viewing purchase orders		
<b>Postconditions:</b>	1. A purchase order is entered into the system with the parts the parts person entered		
<b>Normal Flow:</b>	1. The parts person selects a button from the "view purchase orders" screen 2. A screen is brought up with a list of part requests 3. continue at step 1 of UC-PARTS 01.05		
<b>Alternative Flows:</b>	1. In steps 2 through 5 of the normal flow, the parts person may decide to cancel the creation. 1a. The parts person selects a button to "cancel" 2a. The parts person is brought back to the view Purchase Orders screen		
<b>Exceptions:</b>	1. The database system may not be available at this time a. The system displays an error, which sends the parts person back to the home screen.		
<b>Includes:</b>	PARTS.03.02 View Purchase Orders PARTS.06.02 View Vendor Catalogs PARTS.01.02 View Part Requests		
<b>Frequency of Use:</b>	2-8 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands english or spanish		
<b>Notes and Issues:</b>			

## View Purchase Orders

<b>Use Case ID:</b>	UC-PARTS.03.02		
<b>Use Case Name:</b>	View past part orders		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/28/2023
<b>Actors:</b>	Parts Person Mechanics		
<b>Description:</b>	To warranty a part, it may be necessary to see what was ordered in the past.		
<b>Trigger:</b>	The parts person clicks a button to view purchase orders instead of viewing the request list		
<b>Preconditions:</b>	Parts person wishes to view orders		
<b>Postconditions:</b>	The window shows a list of past orders with the specified search criteria		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person clicks the button to view purchase orders</li> <li>2. They input the search criteria for which past orders they wish to see</li> <li>3. The list is displayed</li> <li>4. The parts person can then select an order to see more details, including seeing the details of past parts orders and maintenance requests</li> </ol>		
<b>Alternative Flows:</b>	<p>3a. there are no past orders to display with the given search criteria</p> <ol style="list-style-type: none"> <li>1. A message is displayed on the screen that would otherwise have the list, stating that no orders fit the search criteria</li> <li>2. The parts person can then choose to input new search criteria</li> <li>3. Return to step 4 of the normal use case flow</li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	1 time per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>	Exceptions TBD Includes TBD Special Requirements TBD		

## Send Part Orders

<b>Use Case ID:</b>	UC-PARTS.03.03		
<b>Use Case Name:</b>	Send Part Orders		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/28/2023
<b>Actors:</b>	Parts Person Mechanics Suppliers		
<b>Description:</b>	After Approving part requests, the parts person will check if the part is in stock. If the part is out they will send out an order for the part to the suppliers, where they will fulfill the order and send parts to the maintenance shop.		
<b>Trigger:</b>	The parts person clicks a button to send out the order to suppliers		
<b>Preconditions:</b>	Mechanic creates a part request Mechanic sends part request to parts person Parts person reviews the request The parts person approves the request		
<b>Postconditions:</b>	An order is sent to suppliers A confirmation message stating an order was sent successfully is shown The order is archived The window returns to the list of part requests		
<b>Normal Flow:</b>	The parts person calls, emails, or visits the web store of the supplier to order the part The parts person clicks a button to confirm the order is placed A window pops up asking if they are sure that the order has been placed Upon clicking confirm, the order page is closed		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	UC-PARTS.01.02 View Part Requests UC-PARTS.01.01 Request Confirmation		
<b>Frequency of Use:</b>	1 time per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>	Exceptions TBD Includes TBD Special Requirements TBD		

## Modify Purchase Orders

<b>Use Case ID:</b>	UC-PARTS.03.04		
<b>Use Case Name:</b>	Modify Purchase Orders		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	10/30/2023
<b>Actors:</b>	Parts Person		
<b>Description:</b>	The parts person updates an unsent purchase order.		
<b>Trigger:</b>	The Parts Person needs to update an unsent purchase order, because they need to order more or fewer items from a vendor.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The purchase order is already created in the system</li> <li>2. The purchase order is unsent</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The purchase order in the system is modified to show new details</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person selects a button from the "view purchase orders" screen</li> <li>2. A screen is brought up with details of the selected purchase order and the option to modify each field</li> <li>3. The parts person may change the amounts of existing items on the list, or add new items with amounts from the same vendor to the purchase order.</li> <li>4. The parts person selects a button to move to the next screen.</li> <li>5. The screen displays a summary of the purchase order, with an "update" button. The parts person selects the update button.</li> <li>6. The purchase order is updated in the system.</li> </ol>		
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. In steps 2 through 5 of the normal flow, the parts person may decide to cancel the update.             <ol style="list-style-type: none"> <li>1a. The parts person selects a button to "cancel"</li> <li>2a. The parts person is brought back to the view Purchase Orders screen</li> </ol> </li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. The database system may not be available at this time             <ol style="list-style-type: none"> <li>a. The system displays an error, which sends the parts person back to the home screen.</li> </ol> </li> </ol>		
<b>Includes:</b>	PARTS.03.02 View Purchase Orders		
<b>Frequency of Use:</b>	2-8 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>			

## Contact Supplier

<b>Use Case ID:</b>	UC-PARTS.03.03		
<b>Use Case Name:</b>	Contact Supplier		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/28/2023
<b>Actors:</b>	Parts Person Mechanics Suppliers		
<b>Description:</b>	In the event that an order cannot be fulfilled for one reason or another, it may be beneficial to send a message to one or more part suppliers asking for an alternative way to get the part needed.		
<b>Trigger:</b>	The parts person clicks a button to view suppliers' contact information		
<b>Preconditions:</b>	1. Mechanic creates a part request 2. Mechanic sends part request to parts person 3. Parts person views the request 4. The parts person approves the request 5. Parts person sends an order out for fulfillment		
<b>Postconditions:</b>	1. The contact information for a supplier is displayed, with links included to make sending a message easier		
<b>Normal Flow:</b>	1. The parts person clicks to view the details of a sent order 2. The parts person clicks to view the information of a supplier included in the order 3. The parts person can then contact the supplier in whatever manner they determine to be the best		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	UC-PARTS.03.02 View Past Part Orders		
<b>Frequency of Use:</b>	1 time per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>	Exceptions TBD Includes TBD Special Requirements TBD		

## Reactive Purchase Orders

<b>Use Case ID:</b>	UC-PARTS.3.5		
<b>Use Case Name:</b>	Reactivate Purchase Orders		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	10/30/2023
<b>Actors:</b>	Parts Person		
<b>Description:</b>	The parts person reactivates a purchase order		
<b>Trigger:</b>	The Parts Person needs to reactivate an unsent purchase order, because of a business rule or because the order is now needed.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The purchase order is already created in the system</li> <li>2. The purchase order is unsent</li> <li>3. The purchase order is deactivated</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The purchase order in the system is modified display in normal view</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person selects a button from the “view purchase orders” screen to show inactive purchase orders</li> <li>2. The parts person selects a button next to a purchase order to mark it as active</li> <li>3. The system prompts to ensure the user is sure they wish to deactivate the order.</li> <li>4. The parts person selects “yes”</li> <li>5. The system properly updates.</li> </ol>		
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. In steps 3-4 of the normal flow, the parts person may decide to cancel the update.             <ol style="list-style-type: none"> <li>1a. The parts person selects a button to “cancel”</li> <li>2a. The parts person is brought back to the view Purchase Orders screen</li> </ol> </li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. The database system may not be available at this time             <ol style="list-style-type: none"> <li>a. The system displays an error, which sends the parts person back to the home screen.</li> </ol> </li> </ol>		
<b>Includes:</b>	PARTS.3.2 View Purchase Orders PARTS.6.2 View Vendor Catalogues PARTS.6.3 View list of Vendors		
<b>Frequency of Use:</b>	2-8 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>			

## View Purchase Order Details

<b>Use Case ID:</b>	UC-PARTS.03.07		
<b>Use Case Name:</b>	View Purchase Order Details		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Parts Person Mechanics		
<b>Description:</b>	To warranty a part or make changes to an order, it will be necessary to view the details of a purchase order		
<b>Trigger:</b>	The parts person clicks a button to view purchase orders, and clicks on a single purchase order to view the details		
<b>Preconditions:</b>	Parts person wishes to view order details of a single purchase order		
<b>Postconditions:</b>	The window shows a list of past orders with the specified search criteria		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person clicks the button to view purchase orders</li> <li>2. They input the search criteria for which past orders they wish to see</li> <li>3. The list is displayed</li> <li>4. The parts person can then select an order to see more details, including seeing the details of past parts orders and maintenance requests</li> </ol>		
<b>Alternative Flows:</b>	<p>3a. there are no past orders to display with the given search criteria</p> <ol style="list-style-type: none"> <li>1. A message is displayed on the screen that would otherwise have the list, stating that no orders fit the search criteria</li> <li>2. The parts person can then choose to input new search criteria</li> <li>3. Return to step 4 of the normal use case flow</li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	1 time per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>	Exceptions TBD Includes TBD Special Requirements TBD		

## Create Change Order

<b>Use Case ID:</b>	UC-PARTS.04.01		
<b>Use Case Name:</b>	View Change Order		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Max Fare
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	10/30/2023
<b>Actors:</b>	Parts Person		
<b>Description:</b>	If the purchase order sent to a vendor has an incorrect quantity ordered, the parts person sends that vendor a Change Order to change how many should be shipped.		
<b>Trigger:</b>	The Parts Person needs to order a different number of parts		
<b>Preconditions:</b>	1. Parts Person has a list of parts or items they need 2. The parts person is viewing Purchase Orders		
<b>Postconditions:</b>	1. A Change order is entered into the system with the parts the parts person entered		
<b>Normal Flow:</b>	1. The parts person selects to view the details of a purchase order 2. upon selecting a button to 'Add a Change', a screen pops up to accept changes in quantity, including to remove the item from the order. 3. The parts person inputs the corrected data 4. The parts person selects the create button. 5. The Change order is created in the system.		
<b>Alternative Flows:</b>	1. In steps 2 through 4 of the normal flow, the parts person may decide to cancel the creation. 1a. The parts person selects a button to "cancel" 2a. The parts person is brought back to the view Purchase Orders screen		
<b>Exceptions:</b>	1. The database system may not be available at this time a. The system displays an error, which sends the parts person back to the home screen.		
<b>Includes:</b>	PARTS.03.02 - View Purchase Orders PARTS.03.0		
<b>Frequency of Use:</b>	2-8 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>			

## View Change Orders

<b>Use Case ID:</b>	UC-PARTS.4.5	
<b>Use Case Name:</b>	View Change Orders	
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>
<b>Actors:</b>	Parts Person	
<b>Description:</b>	The parts person needs to view the list of change orders	
<b>Trigger:</b>	The Parts Person needs to deactivate, reactivate, update, or submit change orders	
<b>Preconditions:</b>	1. The user logged in is a Parts Person	
<b>Postconditions:</b>	1. The Change Order List is displayed	
<b>Normal Flow:</b>	1. The parts person selects an option from the main menu 2. The parts person views a list of change orders	
<b>Alternative Flows:</b>	1. In steps 3-4 of the normal flow, the parts person may decide to cancel the update. 1a. The parts person selects a button to "cancel" 2a. The parts person is brought back to the view Change Orders screen	
<b>Exceptions:</b>	1. The database system may not be available at this time a. The system displays an error, which sends the parts person back to the home screen.	
<b>Includes:</b>		
<b>Frequency of Use:</b>	2-8 times per week	
<b>Special Requirements:</b>		
<b>Assumptions:</b>	The parts person understands English or Spanish	
<b>Notes and Issues:</b>		

## Modify Change Orders

<b>Use Case ID:</b>	UC-PARTS.4.3		
<b>Use Case Name:</b>	Modify Change Orders		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	10/30/2023
<b>Actors:</b>	Parts Person		
<b>Description:</b>	The parts person updates an unsent Change order.		
<b>Trigger:</b>	The Parts Person needs to update an unsent Change order, because they need to order more or fewer items from a vendor.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Change order is already created in the system</li> <li>2. The Change order is unsent</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The Change order in the system is modified to show new details</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person selects a button from the “view Change orders” screen</li> <li>2. A screen is brought up with details of the selected Change order and the option to modify each field</li> <li>3. The parts person may change the amounts of existing items on the list or add new items with amounts from the same vendor to the Change order.</li> <li>4. The parts person selects a button to move to the next screen.</li> <li>5. The screen displays a summary of the Change order, with an “update” button. The parts person selects the update button.</li> <li>6. The Change order is updated in the system.</li> </ol>		
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. In steps 2 through 5 of the normal flow, the parts person may decide to cancel the update.             <ol style="list-style-type: none"> <li>1a. The parts person selects a button to “cancel”</li> <li>2a. The parts person is brought back to the view Change Orders screen</li> </ol> </li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. The database system may not be available at this time             <ol style="list-style-type: none"> <li>a. The system displays an error, which sends the parts person back to the home screen.</li> </ol> </li> </ol>		
<b>Includes:</b>	PARTS.4.2 View Change Orders PARTS.6.3 View list of Vendors		
<b>Frequency of Use:</b>	2-8 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>			

## Reactivate Change Orders

<b>Use Case ID:</b>	UC-PARTS.3.5		
<b>Use Case Name:</b>	Reactivate Change Orders		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Jacob Rohr
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	10/30/2023
<b>Actors:</b>	Parts Person		
<b>Description:</b>	The parts person reactivates a Change order		
<b>Trigger:</b>	The Parts Person needs to reactivate an unsent Change order, because of a business rule or because the order is now needed.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Change order is already created in the system</li> <li>2. The Change order is unsent</li> <li>3. The Change order is deactivated</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The Change order in the system is modified display in normal view</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person selects a button from the “view Change orders” screen to show inactive Change orders</li> <li>2. The parts person selects a button next to a Change order to mark it as active</li> <li>3. The system prompts to ensure the user is sure they wish to deactivate the order.</li> <li>4. The parts person selects “yes”</li> <li>5. The system properly updates.</li> </ol>		
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. In steps 3-4 of the normal flow, the parts person may decide to cancel the update.             <ol style="list-style-type: none"> <li>1a. The parts person selects a button to “cancel”</li> <li>2a. The parts person is brought back to the view Change Orders screen</li> </ol> </li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. The database system may not be available at this time             <ol style="list-style-type: none"> <li>a. The system displays an error, which sends the parts person back to the home screen.</li> </ol> </li> </ol>		
<b>Includes:</b>	PARTS.4.2 View Change Orders PARTS.6.3 View list of Vendors		
<b>Frequency of Use:</b>	2-8 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>			

## Submit Change Orders

<b>Use Case ID:</b>	UC-PARTS.4.6	
<b>Use Case Name:</b>	Submit Change Orders	
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>
<b>Actors:</b>	Parts Person	
<b>Description:</b>	The parts person submits a change order to a vendor	
<b>Trigger:</b>	The Parts Person needs to submit a change order to a vendor	
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The user logged in is a Parts Person</li> <li>2. The parts person is viewing the change orders list</li> </ol>	
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The Change Order is submitted to the Vendor</li> </ol>	
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The parts person selects a button next to a specific change order</li> <li>2. The parts person views a summary screen and selects a button to confirm the order</li> <li>3. The parts order is sent to the vendor</li> </ol>	
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. In steps 2-3 of the normal flow, the parts person may decide to cancel the update.             <ol style="list-style-type: none"> <li>1a. The parts person selects a button to "cancel"</li> <li>2a. The parts person is brought back to the view Change Orders screen</li> </ol> </li> </ol>	
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. The database system may not be available at this time             <ol style="list-style-type: none"> <li>a. The system displays an error, which sends the parts person back to the home screen.</li> </ol> </li> </ol>	
<b>Includes:</b>	4.2 View Change Orders	
<b>Frequency of Use:</b>	2-8 times per week	
<b>Special Requirements:</b>		
<b>Assumptions:</b>	The parts person understands English or Spanish	
<b>Notes and Issues:</b>		

## View Maintenance Order Details

<b>Use Case ID:</b>	UC-PARTS.04.01		
<b>Use Case Name:</b>	View Maintenance Order Details		
<b>Created By:</b>	Max Fare	<b>Last Updated By:</b>	Ben Collins
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	10/28/2023
<b>Actors:</b>	Parts Person Mechanic Fleet Manager		
<b>Description:</b>	To determine the validity of a request the parts person should view the details of not only the part request, but also the maintenance work order that created the need for the request.		
<b>Trigger:</b>	The parts person clicks a button to view the maintenance order linked to a part request		
<b>Preconditions:</b>	1. Mechanic creates a part request 2. Mechanic sends part request to parts person 3. Parts person views the request		
<b>Postconditions:</b>	1. The window showing the maintenance order will be closed		
<b>Normal Flow:</b>	1. The parts person views a list with the details of the part request 2. Wanting to know more, the parts person clicks on a view maintenance order button 3. A window appears with the details of the maintenance order 4. The parts person closes the window, returning to the part request window		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	UC-PARTS.01.02 View Part Requests		
<b>Frequency of Use:</b>	2 times per week		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	The parts person understands English or Spanish		
<b>Notes and Issues:</b>	1. Exceptions TBD 2. Includes TBD 3. Special Requirements TBD		

# Employee – Operations Manager

## Create Driver

<b>Use Case ID:</b>	OPS.1.1		
<b>Use Case Name:</b>	Create Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager assigns an employee to be a driver		
<b>Trigger:</b>	1. An employee has agreed to be a driver		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The new driver's driving schedule is known 3. The new driver's license class is known 4. The new driver is in the database as an employee.		
<b>Postconditions:</b>	1. The driver is added to the database of available drivers.		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen 2. Operations Manager is brought to the add driver window, where operations manager enters the driver's weekly available hours and Driver's License Class 3. Operations Manager selects an action button to save the details 4. Details are stored in database 5. Operations manager is brought back to the driver schedule screen.		
<b>Alternative Flows:</b>	In steps 2-3 of the normal flow, Operations Manager may choose to cancel the action.  1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled, and the Ops Manager is brought back to the driver schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, the databases containing the data may be unavailable  1. In this situation, an error is displayed, and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## View Driver

<b>Use Case ID:</b>	OPS.1.2		
<b>Use Case Name:</b>	View Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager views a driver's information		
<b>Trigger:</b>	1. A driver's availability or license need to be known		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The driver is already in the database		
<b>Postconditions:</b>	1. The driver's information is displayed		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen 2. Operations Manager is brought to the view driver window		
<b>Alternative Flows:</b>	1. N/A		
<b>Exceptions:</b>	1a. In step 2 of the normal flow, the databases containing the data may be unavailable  1. In this situation, an error is displayed, and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Update Driver

<b>Use Case ID:</b>	OPS.1.3		
<b>Use Case Name:</b>	Create Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager modifies a driver's hours or license		
<b>Trigger:</b>	1. A driver's availability or license change		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The driver's new information is known 3. The driver is already in the database		
<b>Postconditions:</b>	1. The driver is updated to reflect the new information.		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen 2. Operations Manager is brought to the view driver window 3. Operations Manager presses a button to change to edit driver 4. Operations Manager then enters the modified data in the correct fields 5. Operations Manager selects an action button to save the details 6. Details are stored in database 7. Operations manager is brought back to the driver schedule screen.		
<b>Alternative Flows:</b>	In steps 3-5 of the normal flow, Operations Manager may choose to cancel the action.  1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled, and the Ops Manager is brought back to the driver schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, the databases containing the data may be unavailable  1. In this situation, an error is displayed, and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules OPS.5.2 View Driver		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Deactivate Driver

<b>Use Case ID:</b>	OPS.1.4		
<b>Use Case Name:</b>	Deactivate Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager deactivates a driver		
<b>Trigger:</b>	1. The driver no longer works for the company		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The driver is already in the database		
<b>Postconditions:</b>	1. The driver is deactivated in the database and no longer appears in normal view		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen 2. Operations Manager is brought to the view driver window 3. Operations Manager presses a button to change to deactivate driver 4. Operations Manager is prompted to confirm the change 5. Operations Manager selects an action button to save the details 6. Details are stored in database 7. Operations manager is brought back to the driver schedule screen.		
<b>Alternative Flows:</b>	In steps 3-5 of the normal flow, Operations Manager may choose to cancel the action.  1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled, and the Ops Manager is brought back to the driver schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, the databases containing the data may be unavailable  1. In this situation, an error is displayed, and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules  OPS.5.2 View Driver		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reactivate Driver

<b>Use Case ID:</b>	OPS.1.5		
<b>Use Case Name:</b>	Reactivate Driver		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager reactivates a driver		
<b>Trigger:</b>	1. The driver works for the company again		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The driver is already in the database 3. The driver is deactivated		
<b>Postconditions:</b>	1. The driver is deactivated in the database and no longer appears in normal view		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the driver schedule screen to show inactive drivers 2. Operations manager selects a button from the driver schedule screen 3. Operations Manager is brought to the view driver window 4. Operations Manager presses a button to change to deactivate driver 5. Operations Manager is prompted to confirm 6. Operations Manager selects an action button to save the details 7. Details are stored in database 8. Operations manager is brought back to the driver schedule screen.		
<b>Alternative Flows:</b>	In steps 4-6 of the normal flow, Operations Manager may choose to cancel the action.  1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled, and the Ops Manager is brought back to the driver schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, the databases containing the data may be unavailable  1. In this situation, an error is displayed, and the user is brought back to the View Driver Schedules window.		
<b>Includes:</b>	DISP.2.1 View Driver Schedules  OPS.5.2 View Driver		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Create Vehicle

<b>Use Case ID:</b>	OPS.2.1		
<b>Use Case Name:</b>	Create Vehicle		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager adds a new vehicle to the database		
<b>Trigger:</b>	1. A vehicle is purchased or otherwise added to the fleet		
<b>Preconditions:</b>	1. Logged in employee has Operations Manager role 2. The new Vehicle's basic information is known		
<b>Postconditions:</b>	1. The Vehicle is added to the database of available Vehicles.		
<b>Normal Flow:</b>	1. Operations Manager clicks a button from the Vehicle schedule screen 2. Operations Manager is brought to the add Vehicle window, where operations manager enters the vehicle's information. 3. Operations Manager selects an action button to save the details 4. Details are stored in database 5. Operations manager is brought back to the Vehicle schedule screen.		
<b>Alternative Flows:</b>	In steps 2-3 of the normal flow, Operations Manager may choose to cancel the action.  1. Operations Manager selects a cancel button or closes the window. 2. The action is canceled, and the Ops Manager is brought back to the Vehicle schedule screen.		
<b>Exceptions:</b>	1a. In step 2-4 of the normal flow, the databases containing the data may be unavailable  1. In this situation, an error is displayed, and the user is brought back to the View Vehicle Schedules window.		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Deactivate Vehicle

<b>Use Case ID:</b>	OPS.2.2		
<b>Use Case Name:</b>	Deactivate Vehicle		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager removes a vehicle from active service		
<b>Trigger:</b>	<ul style="list-style-type: none"> <li>1. A vehicle's time in active service has come to an end due to aging out</li> <li>2. A vehicle's time in active service has come to an end due to an accident</li> </ul>		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Logged in employee has Operations Manager role</li> <li>2. The vehicle is active in the system</li> </ul>		
<b>Postconditions:</b>	1. The Vehicle is removed from the list of active vehicles		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Operations Manager clicks a button from the Vehicle schedule screen</li> <li>2. Operations Manager is asked to confirm they wish to deactivate the selected vehicle</li> <li>3. Operations Manager selects an action button to confirm</li> <li>4. Details are stored in database</li> <li>5. Operations manager is brought back to the Vehicle schedule screen.</li> </ul>		
<b>Alternative Flows:</b>	<p>In steps 2-3 of the normal flow, Operations Manager may choose to cancel the action.</p> <ul style="list-style-type: none"> <li>1. Operations Manager selects a cancel button or closes the window.</li> <li>2. The action is canceled, and the Ops Manager is brought back to the Vehicle schedule screen.</li> </ul>		
<b>Exceptions:</b>	<p>1a. In step 2-4 of the normal flow, the databases containing the data may be unavailable</p> <ul style="list-style-type: none"> <li>1. In this situation, an error is displayed, and the user is brought back to the View Vehicle Schedules window.</li> </ul>		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Reactivate Vehicle

<b>Use Case ID:</b>	OPS.2.3		
<b>Use Case Name:</b>	Reactivate Vehicle		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	11/27/2023	<b>Last Revision Date:</b>	11/27/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	Operations Manager brings a vehicle back into active service		
<b>Trigger:</b>	<p>1. A vehicle that was previously considered unfit for work needs to be used for work.</p>		
<b>Preconditions:</b>	<p>1. Logged in employee has Operations Manager role</p> <p>2. The vehicle is deactivated in the system</p>		
<b>Postconditions:</b>	1. The Vehicle is brought back to the list of active vehicles		
<b>Normal Flow:</b>	<p>1. Operations Manager clicks a button from the Vehicle schedule screen to view inactive vehicles</p> <p>2. Operations manager clicks an action button associated with a specific vehicle</p> <p>3. Operations Manager is asked to confirm they wish to reactivate the selected vehicle</p> <p>4. Operations Manager selects an action button to confirm</p> <p>5. Details are stored in database</p> <p>6. Operations manager is brought back to the Vehicle schedule screen.</p>		
<b>Alternative Flows:</b>	<p>In steps 2-3 of the normal flow, Operations Manager may choose to cancel the action.</p> <p>1. Operations Manager selects a cancel button or closes the window.</p> <p>2. The action is canceled, and the Ops Manager is brought back to the Vehicle schedule screen.</p>		
<b>Exceptions:</b>	<p>1a. In step 2-4 of the normal flow, the databases containing the data may be unavailable</p> <p>1. In this situation, an error is displayed, and the user is brought back to the View Vehicle Schedules window.</p>		
<b>Includes:</b>	DISP.1.1 View Vehicle Schedules		
<b>Frequency of Use:</b>	A few times per week		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	N/A		
<b>Notes and Issues:</b>	N/A		

## Create New Stop

<b>Use Case ID:</b>	OPS.3.1		
<b>Use Case Name:</b>	Create New Stop		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Operations Manager.		
<b>Description:</b>	Create a new stop, which is a geographic point on a map which is then associated with a route to form the group of stops on the route.		
<b>Trigger:</b>	A need for a new stop in the system.		
<b>Preconditions:</b>	(1) The Operations Manager needs to be known to the system and have the right to create a stop.  (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	A new stop is created in the system.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user navigates to the bus route management section of the application.</li> <li>3. The user selects the “Create New Stop” menu option/button. The New Stop form appears.</li> <li>4. The user enters top level stop information (street address, zip code, latitude, longitude, active).</li> <li>5. Once finished, the user saves the entered data, and the new stop is created.</li> </ol>		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	<p>1a. In step 1 of the normal flow, Operations Manager inserts incorrect login information.</p> <ol style="list-style-type: none"> <li>1. System will prompt user that entered info is incorrect</li> <li>2. User will enter in correct log in</li> </ol> <p>Normal flow will continue step 2</p>		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	This use case creates a stop (a geographic point on a map). This is contrasted with a route stop, which is the association of a stop with a route.		

## List Existing Stops

<b>Use Case ID:</b>	OPS.3.2		
<b>Use Case Name:</b>	List Existing Stops		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Dispatcher, Operations Manager.		
<b>Description:</b>	View the list of existing stops in the system.		
<b>Trigger:</b>	A need to view all stops in the system.		
<b>Preconditions:</b>	(1) The Dispatcher/Operations Manager needs to be known to the system and have rights to generate a list of all stops and filter results. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	A list of all stops in the system is presented to the user.		
<b>Normal Flow:</b>	1. The user logs into the system. 2. The user navigates to the bus route management section of the application. 3. The user selects the “List Stops” menu option/button. The Stop List form appears. 4. The user can view all stops in the database, filtering/refreshing the list to focus their search. 5. Once finished, the user closes the Stop List form.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, Dispatcher/Operations Manager inserts incorrect login information. <ul style="list-style-type: none"> <li>1. System will prompt user that entered info is incorrect</li> <li>2. User will enter in correct log in</li> </ul> Normal flow will continue on step 2		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	A stop is a geographic point on a map. This is contrasted with a route stop, which is the association of a stop with a route.		

## Update Existing Stop

<b>Use Case ID:</b>	OPS.3.3		
<b>Use Case Name:</b>	Update Existing Stop		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Operations Manager.		
<b>Description:</b>	Update an existing stop, which is a geographic point on a map which is then associated with a route to form the group of stops on the route.		
<b>Trigger:</b>	A need for a new stop in the system.		
<b>Preconditions:</b>	(1) The Operations Manager needs to be known to the system and have the right to create a stop.  (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	A new stop is created in the system.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user navigates to the bus route management section of the application.</li> <li>3. The user selects the “Update Stop” menu option/button. The Update Stop form appears.</li> <li>4. The user edits the existing stop data (street address, zip code, latitude, longitude, active).</li> <li>5. Once finished, the user saves the entered data, and the new stop is created.</li> </ol>		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	<p>1a. In step 1 of the normal flow, Operations Manager inserts incorrect login information.</p> <ol style="list-style-type: none"> <li>1. System will prompt user that entered info is incorrect</li> <li>2. User will enter in correct log in</li> </ol> <p>Normal flow will continue to step 2</p>		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	A stop is a geographic point on a map. This is contrasted with a route stop, which is the association of a stop with a route.		

## Deactivate Stop

<b>Use Case ID:</b>	OPS.3.4		
<b>Use Case Name:</b>	Deactivate Stop		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Operations Manager.		
<b>Description:</b>	Deactivate an existing stop.		
<b>Trigger:</b>	The need to activate a stop to prevent its use by the system.		
<b>Preconditions:</b>	(1) The Operations Manager needs to be known to the system and have the access level to deactivate a stop. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	The stop is deactivated and not available for use by the system.		
<b>Normal Flow:</b>	1. User logs into the system. 2. User navigates to the stop management section of the application. 3. User searches for the stop to be deactivated; when found, the user opens the stop details update form for the route. 4. The user unchecks the active checkbox on the update form. 5. The user clicks the Update button.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, Operations Manager inserts incorrect login information. 1. System will prompt user that entered info is incorrect 2. User will enter in correct log in Normal flow will continue on step 2		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## Activate Stop

<b>Use Case ID:</b>	OPS.3.5		
<b>Use Case Name:</b>	Activate Stop		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Operations Manager.		
<b>Description:</b>	Activate an existing stop.		
<b>Trigger:</b>	The need to activate a stop for use by the system.		
<b>Preconditions:</b>	(1) The Operations Manager needs to be known to the system and have rights to activate a stop. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	The stop is activated for use by the system.		
<b>Normal Flow:</b>	1. User logs into the system. 2. User navigates to the stop management section of the application. 3. User searches for the stop to be updated; when found, the user opens the stop details update form for the route. 4. The user checks the active checkbox on the update form. 5. The user clicks the Update button.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, Operations Manager inserts incorrect login information. 1. System will prompt user that entered info is incorrect 2. User will enter in correct log in Normal flow will continue on step 2		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## Create New Bus Route

<b>Use Case ID:</b>	OPS.4.1		
<b>Use Case Name:</b>	Create New Bus Route		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Operations Manager.		
<b>Description:</b>	Create a new bus route.		
<b>Trigger:</b>	A need for a new bus route in the system.		
<b>Preconditions:</b>	(1) The Operations Manager needs to be known to the system and have rights to create a bus route. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	A new bus route is created in the system.		
<b>Normal Flow:</b>	1. The user logs into the system. 2. The user navigates to the bus route management section of the application. 3. The user selects the “Create New Route” menu option/button. The New Route form appears. 4. The user enters top level route information (name, effective start date, length, etc). Also entered is the assignment of stops to the route. 5. Once finished, the user saves the entered data, and the new route is created.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, Operations Manager inserts incorrect login information. 1. System will prompt user that entered info is incorrect 2. User will enter in correct log in Normal flow will continue on step 2		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	It should be noted that creating a new bus route is different from creating a bus schedule for a given route. A bus route describes the route (path), not the schedule (bus stop times). Many buses drive the same route repeatedly throughout the day and make stops at the same points many times during the day. A way to establish a bus schedule with start/stop points and times		

	needs to be developed and a way to associate those schedules with a bus route needs to be created.
--	--

## List Existing Bus Routes

<b>Use Case ID:</b>	OPS.4.2		
<b>Use Case Name:</b>	List Existing Bus Routes		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Dispatcher, Operations Manager.		
<b>Description:</b>	List Existing Bus Routes		
<b>Trigger:</b>	A need to search for a route for business processes (e.g. - dispatching a driver to a route)		
<b>Preconditions:</b>	(1) The Dispatcher/Operations Manager needs to be known to the system and have rights to search/view existing bus route details. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	The user finds bus route information for business purposes effectively.		
<b>Normal Flow:</b>	1. The user logs into the system. 2. The user navigates to the bus route management section of the application. 3. The user selects the “Search for Routes” menu option/button. The Search form appears. 4. The user performs searches by entering various search terms and filters until they find the results, they need for their business purpose. The search results have links/buttons that allow drill-down on individual routes for more granular detail.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## View Bus Route Details

<b>Use Case ID:</b>	OPS.4.3		
<b>Use Case Name:</b>	View Bus Route Details		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Dispatcher, Operations Manager.		
<b>Description:</b>	View the details of a bus route.		
<b>Trigger:</b>	Need to determine details of an existing bus route.		
<b>Preconditions:</b>	(1) The Dispatcher/Operations Manager needs to be known to the system and have rights to create a bus route. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	The details of the bus route are presented to the user.		
<b>Normal Flow:</b>	1. The user logs into the system. 2. The user navigates to the bus route management section of the application. 3. The user selects the “Search for Routes” menu option/button. The Search form appears. 4. The user performs searches by entering various search terms and filters until they find the route for which they wish to see the details. On the route search details line, they click the “details” button - a dialog will appear with the bus route details.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, Dispatcher/Operations Manager inserts incorrect login information. <ul style="list-style-type: none"> <li>1. System will prompt user that entered info is incorrect</li> <li>2. User will enter in correct log in</li> </ul> Normal flow will continue on step 2		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## Update Bus Route Details

<b>Use Case ID:</b>	OPS.4.4		
<b>Use Case Name:</b>	Update Bus Route Details		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Operations Manager.		
<b>Description:</b>	Update the details of an existing bus route.		
<b>Trigger:</b>	There is a need to update the details of an existing bus route		
<b>Preconditions:</b>	(1) The Operations Manager needs to be known to the system and have rights to update the details of a bus route. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	The Bus Route details are successfully updated.		
<b>Normal Flow:</b>	1. User logs into the system. 2. User navigates to the route management section of the application. 3. User searches for the route to be updated; when found, the user opens the bus route details update form for the route. 4. The user changes the necessary data for the update, including updating the assignment of stops to the route. 5. The user clicks the Update button.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, Operations Manager inserts incorrect login information. 1. System will prompt user that entered info is incorrect 2. User will enter in correct log in Normal flow will continue on step 2		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## Deactivate Bus Route

<b>Use Case ID:</b>	OPS.4.5		
<b>Use Case Name:</b>	Deactivate Bus Route		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Operations Manager.		
<b>Description:</b>	Deactivate an existing bus route.		
<b>Trigger:</b>	The need to deactivate a bus route (make unavailable for use the system).		
<b>Preconditions:</b>	(1) The Operations Manager needs to be known to the system and have rights to deactivate a bus route. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	The bus route is deactivated/made unavailable for use by the system.		
<b>Normal Flow:</b>	1. User logs into the system. 2. User navigates to the route management section of the application. 3. User searches for the route to be updated; when found, the user opens the bus route details update form for the route. 4. The user un-checks the active checkbox on the update form. 5. The user clicks the Update button.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, Operations Manager inserts incorrect login information. 1. System will prompt user that entered info is incorrect 2. User will enter in correct log in Normal flow will continue on step 2		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## Activate Bus Route

<b>Use Case ID:</b>	OPS.4.6		
<b>Use Case Name:</b>	Activate Bus Route		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	10/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Operations Manager.		
<b>Description:</b>	Activate an existing bus route.		
<b>Trigger:</b>	The need to activate a bus route for use by the system.		
<b>Preconditions:</b>	(1) The Operations Manager needs to be known to the system and have the privileges to activate a bus route. (2) A data store must exist to store the updated data.		
<b>Postconditions:</b>	The bus route is activated for use by the system.		
<b>Normal Flow:</b>	1. User logs into the system. 2. User navigates to the route management section of the application. 3. User searches for the route to be updated; when found, the user opens the bus route details update form for the route. 4. The user checks the active checkbox on the update form. 5. The user clicks the Update button.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, Operations Manager inserts incorrect login information. 1. System will prompt user that entered info is incorrect 2. User will enter in correct log in Normal flow will continue on step 2		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Periodically (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## Create Transit Service

<b>Use Case ID:</b>	OPS.5.1		
<b>Use Case Name:</b>	Create Transit Service		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/12/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	The Operations Manager must create a new transit service		
<b>Trigger:</b>	The Operations Manager has a need for a new transit service		
<b>Preconditions:</b>	1. Operations Manager is authenticated.		
<b>Postconditions:</b>	1. A new transit service is shown in the list of transit services		
<b>Normal Flow:</b>	1. Operations Manager navigates to the “Manage Transit Services” page 2. Operations Manager clicks the “Create New Service” button 3. Operations Manager is presented with form requesting transit service name, vehicle type, and minimum and maximum seats 4. Operations Manager enters request info 5. Operations Manager clicks “Confirm” button 6. Application returns Operations Manager to the transit service list		
<b>Alternative Flows:</b>  [Alternative Flow - Return back to route details]	4a. In step 4 of the normal flow if the Operations Manager decides to cancel the creation of the transit service  1. Manager clicks “Cancel” button 2. Application returns Operations Manager to the transit service list		
<b>Exceptions:</b>	1. If system fails to save transit service, inform Operations Manager that saving failed. Do not leave “New Service” form.		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	On demand.		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	NA		

## Deactivate Transit Service

<b>Use Case ID:</b>	OPS.5.2		
<b>Use Case Name:</b>	Deactivate Transit Service		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/12/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	The Operations Manager must deactivate an existing active transit service		
<b>Trigger:</b>	The Operations Manager has a need to deactivate a transit service		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>Operations Manager is authenticated.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>The deactivated transit service is removed from the list of active transit services</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Operations Manager navigates to the “Manage Transit Services” page</li> <li>Operations Manager is shown the list of active transit services</li> <li>Operations Manager clicks on the “Edit” button on the service they wish to deactivate</li> <li>Application displays “Edit Service” form</li> <li>Operations Manager clicks the “Deactivate Service” button</li> <li>Application shows modal requesting Operations Manager to confirm deactivation</li> <li>Operations Manager clicks “Confirm” button</li> <li>System updates transit service</li> <li>System deletes any future scheduled rides</li> <li>System deletes any vehicle/driver ride connections</li> <li>System notifies clients and drivers of ride cancellations</li> <li>Application returns to list of transit services</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow - Return back to route details]	<p>7.a. If Operations Manager chooses to cancel deactivation</p> <ol style="list-style-type: none"> <li>Operations Manager clicks “Cancel” button</li> <li>Application returns to “Edit Service” form</li> </ol>		
<b>Exceptions:</b>	<p>5ex.a. If transit service is currently running</p> <ol style="list-style-type: none"> <li>Inform Operations Manager that service cannot be deactivated while it is running</li> <li>Remain on “Edit Service” form</li> </ol> <p>8ex.a. If system fails to update transit service,</p> <ol style="list-style-type: none"> <li>Inform Operations Manager that updating failed.</li> <li>Remain on “Edit Service” form.</li> </ol>		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	On demand.		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	At least one active transit service exists.		

<b>Notes and Issues:</b>	NA
--------------------------	----

## Reactivate Transit Service

<b>Use Case ID:</b>	OPS.5.3		
<b>Use Case Name:</b>	Reactivate Transit Service		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/12/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	The Operations Manager must reactivate an existing inactive transit service		
<b>Trigger:</b>	The Operations Manager has a need to reactivate a transit service		
<b>Preconditions:</b>	1. Operations Manager is authenticated.		
<b>Postconditions:</b>	1. The reactivated transit service is added to the list of active transit services		
<b>Normal Flow:</b>	1. Operations Manager navigates to the “Manage Transit Services” page 2. Operations Manager is shown the list of active transit services 3. Operations Manager chooses to filter by “Inactive” or “All” transit services 4. Operations Manager clicks on the “Edit” button on the service they wish to deactivate 5. Application displays “Edit Service” form 6. Operations Manager clicks the “Activate Service” button 7. Application shows modal requesting Operations Manager to confirm reactivation 8. Operations Manager clicks “Confirm” button 9. System updates transit service 10. Application returns to list of transit services		
<b>Alternative Flows:</b>  [Alternative Flow - Return back to route details]	7.a. If Operations Manager chooses to cancel reactivation 1. Operations Manager clicks “Cancel” button 2. Application returns to “Edit Service” form		
<b>Exceptions:</b>	8ex.a. If system fails to update transit service, 1. Inform Operations Manager that updating failed. 2. Remain on “Edit Service” form.		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	On demand.		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	At least one inactive transit service exists.		
<b>Notes and Issues:</b>	NA		

## Assign Vehicle to Transit Service

<b>Use Case ID:</b>	OPS.5.4		
<b>Use Case Name:</b>	Assign Vehicle to Transit Service		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	11/12/2023	<b>Last Revision Date:</b>	11/12/2023
<b>Actors:</b>	Operations Manager		
<b>Description:</b>	The Operations Manager must add a vehicle to a transit service		
<b>Trigger:</b>	The Operations Manager has a need to add a vehicle to a transit service		
<b>Preconditions:</b>	1. Operations Manager is authenticated.		
<b>Postconditions:</b>	1. The relevant vehicle appears in the list of vehicles for relevant transit service		
<b>Normal Flow:</b>	1. Operations Manager navigates to the “Manage Transit Services” page 2. Operations Manager is shown the list of active transit services 3. Operations Manager clicks on the “Edit” button on the service they wish to deactivate 4. Application displays “Edit Service” form 5. Operations Manager clicks the “Add Vehicle” button 6. Application displays modal showing all vehicles which are not in use 7. Operations Manager selects a vehicle 8. System creates service-vehicle connection 9. Application returns to “Edit Service” form		
<b>Alternative Flows:</b>  [Alternative Flow - Return back to route details]	7.a. If Operations Manager chooses to cancel adding vehicle 1. Operations Manager clicks “Cancel” button 2. Application returns to “Edit Service” form		
<b>Exceptions:</b>	8ex.a. If system fails to update transit service, 1. Inform Operations Manager that updating failed. 2. Remain on “Select Vehicle” modal.		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	On demand.		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	At least one active transit service exists. At least one unassigned vehicle exists.		
<b>Notes and Issues:</b>	NA		

# Employee – Fleet & Operations

## Look up Scheduled Vehicles

<b>Use Case ID:</b>	8.1.1		
<b>Use Case Name:</b>	Lookup Scheduled Vehicles		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Lookup a list of vehicles scheduled		
<b>Trigger:</b>	Manager needs to see upcoming scheduled vehicles		
<b>Preconditions:</b>	1. User is logged in 2. Vehicles have been scheduled for use		
<b>Postconditions:</b>	List of scheduled vehicles is displayed to user		
<b>Normal Flow:</b>	1. User selects to view scheduled vehicles 2. List of vehicles and times they are scheduled is displayed		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Frequency of Use:</b>	Daily or OnDemand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1. Needs sorting and filtering		

## Look up Vehicle User Info

<b>Use Case ID:</b>	8.1.2		
<b>Use Case Name:</b>	Lookup Vehicle User Info		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Allows a manager to lookup the info about a vehicle user		
<b>Trigger:</b>	Manager needs to contact a vehicle user before or after use		
<b>Preconditions:</b>	1. Manager logged in		
<b>Postconditions:</b>	1. User info is displayed		
<b>Normal Flow:</b>	1. User searches for vehicle reservation 2. User selects a reservation 3. User selects to see vehicle user information		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	1.		
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	Need to verify user is a manager to release customer info		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1.		

## Look up Vehicle Waitlist

<b>Use Case ID:</b>	8.1.3		
<b>Use Case Name:</b>	Lookup Vehicle Waitlist		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Lookup a list of people that want vehicles that aren't available for scheduling		
<b>Trigger:</b>	Vehicle is wanted by a user but already scheduled		
<b>Preconditions:</b>	1. Waitlist is enabled 2. Vehicle user enters to be on a waitlist		
<b>Postconditions:</b>	Waitlist is displayed to user		
<b>Normal Flow:</b>	1. User selects to view the waitlist 2. Waitlist is displayed to the user		
<b>Alternative Flows:</b>	1. User looks up a vehicle 2. Selects to see the waitlist for the vehicle 3. Filtered waitlist is displayed to user		
<b>Exceptions:</b>			
<b>Includes:</b>	Alternate includes 9.1.4 vehicle lookup		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	Waitlist enabled		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Look up Employee Information

<b>Use Case ID:</b>	8.2.1		
<b>Use Case Name:</b>	Lookup Employee Info		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Lookup employee info		
<b>Trigger:</b>	Manager needs to view employee info		
<b>Preconditions:</b>	1. Manager is logged in		
<b>Postconditions:</b>	1. Employee info is displayed		
<b>Normal Flow:</b>	1. Selects employee list 2. Selects employee from list to view 3. Submits 4. Employee Info is displayed		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	1.		
<b>Includes:</b>	8.2.2 Lookup Employee List		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1.		

## Look up Employee List

<b>Use Case ID:</b>	8.2.2		
<b>Use Case Name:</b>	Lookup Employee List		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Display a list of employees		
<b>Trigger:</b>	Manager needs to view a list of employees		
<b>Preconditions:</b>	1. Manager is logged in		
<b>Postconditions:</b>	1. Employee list is displayed		
<b>Normal Flow:</b>	1. Selects to view employee list 2. Employee list is displayed		
<b>Alternative Flows:]</b>			
<b>Exceptions:</b>	1.		
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1.		

## Add Employee

<b>Use Case ID:</b>	8.2.3		
<b>Use Case Name:</b>	Add employee		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Enter a new employee into the system		
<b>Trigger:</b>	Employee is hired		
<b>Preconditions:</b>	1. Manager is logged in 2. Employee information has been obtained		
<b>Postconditions:</b>	1. User is entered into the system		
<b>Normal Flow:</b>	1. Selects employee list 2. Selects to add employee 3. Enters employee information 4. Submits 5. Confirm submission 6. Employee number is displayed		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	8.2.2 Lookup Employee List		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1. First Name, Last Name, Address1, Address2, City, State, Zip, Phone, Email, Alternate Phone, Emergency Contact Name, Emergency Contact Relation, Emergency Contact Phone, Emergency Contact Email, Hire Date, Pay, Manager, Active, Termination Date		

## Deactivate Employee

<b>Use Case ID:</b>	8.2.4		
<b>Use Case Name:</b>	Deactivate Employee		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Deactivate an employee that is no longer employed		
<b>Trigger:</b>	Employee is no longer employed		
<b>Preconditions:</b>	1. Manager logged in 2. Employee is no longer employed		
<b>Postconditions:</b>	1. Employee status is deactivated 2. Date of termination if not today		
<b>Normal Flow:</b>	Selects to view employee list Selects employee from list Selects to view employee info Selects to deactivate employee Enters termination date Submits termination Confirms to deactivate employee		
<b>Alternative Flows:</b>	6a No termination date is entered		
[Alternative Flow 1 - Not in Network]	1. Prompts user if they want to use today's date		
<b>Exceptions:</b>	1.		
<b>Includes:</b>	9.2.1 Lookup Employee Info		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	Needs to remove employee permissions		
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1.		

## Change Employee Information

<b>Use Case ID:</b>	8.2.5		
<b>Use Case Name:</b>	Change Employee Info		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Employee		
<b>Description:</b>	Update employee info		
<b>Trigger:</b>	User needs to update their info		
<b>Preconditions:</b>	1. User logged in		
<b>Postconditions:</b>	User info is updated		
<b>Normal Flow:</b>	1. User selects to change their employee info 2. User updates the fields 3. User submits 4. Confirm submission		
<b>Alternative Flows:</b>			
	1.		
<b>Includes:</b>			
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Look up Mileage Driven by Vehicle

<b>Use Case ID:</b>	8.3.1		
<b>Use Case Name:</b>	Lookup Mileage Driven by Vehicle		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/1/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Look up the miles for a vehicle over a set time		
<b>Trigger:</b>	Manager wants to view recent vehicle use		
<b>Preconditions:</b>	User logged in Vehicle has been used recently		
<b>Postconditions:</b>	1. Mileage is displayed for vehicle		
<b>Normal Flow:</b>	1. Navigate to vehicle info 2. Select to see vehicle mileage 3. Enter time range for mileage 4. Mileage is displayed		
<b>Alternative Flows</b>			
<b>Exceptions:</b>	1.		
<b>Includes:</b>	9.1.4 Lookup Vehicle Info		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>	1.		

## Look Up Mileage Driven by Driver

<b>Use Case ID:</b>	8.3.2		
<b>Use Case Name:</b>	Lookup Mileage Driven by Driver		
<b>Created By:</b>	Chris Baenziger	<b>Last Updated By:</b>	
<b>Date Created:</b>	09/18/2023	<b>Last Revision Date:</b>	
<b>Actors:</b>	Fleet Manager Operations Manager		
<b>Description:</b>	Lookup the number of miles driven by a driver		
<b>Trigger:</b>	Manager wants to see how many miles a driver is traveling a day		
<b>Preconditions:</b>	User is logged in Driver has driven in timeframe		
<b>Postconditions:</b>	Driver mileage is displayed		
<b>Normal Flow:</b>	1. User selects to view a driver 2. Selects to see mileage 3. Enters a time range 4. Mileage is displayed to user		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>	8.2.1 Lookup Employee Info		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

# Employee – Customer Support

## View Customer Support Tickets

<b>Use Case ID:</b>	5.1		
<b>Use Case Name:</b>	View list of Customer Support Tickets		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	2023-09-10	<b>Last Revision Date:</b>	2023-11-27
<b>Actors:</b>	Customer Support Representative		
<b>Description:</b>	View the whole list of customer support tickets		
<b>Trigger:</b>	Customer Support Representative signed in		
<b>Preconditions:</b>	<ol style="list-style-type: none"><li>1. The current user is signed in.</li><li>2. The current user is authorized to view customer support tickets.</li></ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"><li>1. The user can view all Customer Support Tickets</li></ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"><li>1. Customer Support Representative selects a button from Home Screen</li><li>2. Customer Support Representative is presented with a list of all tickets and a set of filters to narrow down tickets.</li></ol>		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	<p>2b. In step 2 of the normal flow, there may be no tickets, or the database may not be available.</p> <ol style="list-style-type: none"><li>1. The system notifies the user and presents them with the option to reload the screen or go back to the Home screen.</li></ol>		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	Users are able to submit tickets		
<b>Notes and Issues:</b>	None		

## View Individual Customer Support Ticket

<b>Use Case ID:</b>	5.2		
<b>Use Case Name:</b>	View individual customer support ticket		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	2023-09-10	<b>Last Revision Date:</b>	2023-11-27
<b>Actors:</b>	Customer Support Representative		
<b>Description:</b>	View an individual customer support ticket		
<b>Trigger:</b>	Customer Support Representative signed in.  There is at least 1 Customer Support Ticket.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Signed in user is a Customer Support Representative</li> <li>2. Customer Support Representative is already viewing the list of tickets</li> </ol>		
<b>Postconditions:</b>	1. Customer Support Representative is viewing the record of an individual Customer Support Ticket		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Customer Support Representative selects an individual record from the list.</li> <li>2. Customer Support Representative is presented with a view of the record and all details of the ticket.</li> </ol>		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	<p>2b. In step 1 of the normal flow, the database may not be available.</p> <ol style="list-style-type: none"> <li>1. The system notifies the user and presents them with the option to reload the screen or go back to the Home screen.</li> </ol>		
<b>Includes:</b>	5.1 view list of customer support tickets		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	Users are able to submit tickets		
<b>Notes and Issues:</b>	None		

## Update Customer Support Ticket

<b>Use Case ID:</b>	5.3		
<b>Use Case Name:</b>	Update Customer Support Ticket		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	2023-09-10	<b>Last Revision Date:</b>	2023-11-27
<b>Actors:</b>	Customer Support Representative		
<b>Description:</b>	Update a Customer Support Ticket		
<b>Trigger:</b>	<p>Customer Support Representative signed in.</p> <p>Customer Support Representative has performed troubleshooting or problem solving on the ticket.</p>		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Signed in user is a Customer Support Representative</li> <li>2. Customer Support Representative is already viewing a ticket</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Ticket is updated with Customer Support Representative's notes.</li> <li>2. Ticket may be marked as resolved.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User selects a button to update the ticket.</li> <li>2. User enters notes in applicable fields.</li> <li>3. User selects button to save and exit ticket.</li> </ol> <p>User is then returned to list of tickets.</p>		
<b>Alternative Flows:</b>	<p>3a. In step 3 of the Normal Flow, the user may decide to exit the ticket without saving any changes.</p> <ol style="list-style-type: none"> <li>1. In this case, the only change is that the user selects to exit without saving.</li> <li>2. A prompt will appear to confirm the user's decision.</li> <li>3. The user will then be returned to the list of tickets if they confirm their choice, or back to just before step 3 if they deny.</li> </ol>		
<b>Exceptions:</b>	<p>2b. In step 3 of the normal flow, the database may not be available.</p> <ol style="list-style-type: none"> <li>1. The system notifies the user and presents them with the option to try to resubmit the changes or go back to the Home screen.</li> </ol>		

<b>Includes:</b>	5.1 view list of customer support tickets 5.2 view individual customer support ticket
<b>Frequency of Use:</b>	Many times a day
<b>Special Requirements:</b>	None
<b>Assumptions:</b>	Users are able to submit tickets
<b>Notes and Issues:</b>	None

## Mark Customer Support Ticket as Resolved

<b>Use Case ID:</b>	5.4		
<b>Use Case Name:</b>	Mark Customer Support Ticket as resolved (Deactivate)		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	2023-09-10	<b>Last Revision Date:</b>	2023-11-27
<b>Actors:</b>	Customer Support Representative		
<b>Description:</b>	The Customer Support Representative marks a ticket as resolved.		
<b>Trigger:</b>	Customer Support Representative signed in.  Customer Support Representative has resolved a user's issue.		
<b>Preconditions:</b>	1. Signed in user is a Customer Support Representative. 2. Customer Support Representative is viewing the user's ticket.		
<b>Postconditions:</b>	1. Ticket is marked as resolved. 2. Customer Support Representative is brought back to List of Tickets.		
<b>Normal Flow:</b>	1. User selects a button to update the ticket. 2. User enters notes in applicable fields and checks a box to mark ticket as resolved. 3. User selects button to save and exit ticket. 4. User is then returned to List of tickets.		
<b>Alternative Flows:</b>	3a. In step 3 of the Normal Flow, the user may decide to exit the ticket without saving any changes. <ol style="list-style-type: none"> <li>In this case, the only change is that the user selects to exit without saving.</li> <li>A prompt will appear to confirm the user's decision.</li> </ol> The user will then be returned to the list of tickets if they confirm their choice, or back to just before step 3 if they deny.		
<b>Exceptions:</b>	3b. In step 3 of the normal flow, the database may not be available. <ol style="list-style-type: none"> <li>The system notifies the user and presents them with the option to try to resubmit the changes or go back to the Home screen.</li> </ol>		

<b>Includes:</b>	5.1 view list of tickets. 5.2 view individual ticket. 5.3 update individual ticket.
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	None
<b>Assumptions:</b>	Users are able to submit tickets
<b>Notes and Issues:</b>	None

## Create Related Ticket

<b>Use Case ID:</b>	CSR.01.05		
<b>Use Case Name:</b>	Create Related Ticket		
<b>Created By:</b>	Jacob Rugger	<b>Last Updated By:</b>	Isabella Rosenbohm
<b>Date Created:</b>	2023-09-10	<b>Last Revision Date:</b>	2023-11-27
<b>Actors:</b>	Customer Support Representative		
<b>Description:</b>	The Customer Support Representative takes a ticket and divides it up.		
<b>Trigger:</b>	<p>Customer Support Representative signed in.</p> <p>The Customer Support Representative is viewing a ticket that lists multiple issues.</p>		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The user is a Customer Support Representative.</li> <li>2. The Customer Support Representative is viewing the user's ticket.</li> </ol>		
<b>Postconditions:</b>	1. A new ticket is created and linked to the original.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Customer Support Representative selects the option to update the ticket.</li> <li>2. Customer Support Representative then selects a button to create a sub(or linked) ticket.</li> <li>3. Customer Support Representative is then brought to a "Create new ticket" field.</li> </ol>		
<b>Alternative Flows:</b>	<p>3a. In step 3 of the Normal Flow, the user may decide to exit the ticket without saving any changes.</p> <ol style="list-style-type: none"> <li>1. In this case, the only change is that the user selects to exit without saving.</li> <li>2. prompt will appear to confirm the user's decision.</li> </ol> <p>The user will then be returned to the list of tickets if they confirm their choice, or back to just before step 3 if they deny.</p>		
<b>Exceptions:</b>	<p>3b. In step 3 of the normal flow, the database may not be available.</p> <ol style="list-style-type: none"> <li>1. The system notifies the user and presents them with the option to try to resubmit the changes or go back to the Home screen.</li> </ol>		

<b>Includes:</b>	5.1 view list of tickets. 5.2 view individual ticket. 5.3 update individual ticket.
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	None
<b>Assumptions:</b>	Users are able to submit tickets
<b>Notes and Issues:</b>	None

## Riders – Charter with Driver

### Verify a Reservation Start for a Chartered Vehicle

<b>Use Case ID:</b>	UC 02.01		
<b>Use Case Name:</b>	Verify Reservation Start		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	9/17/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Charter		
<b>Description:</b>	A user verifies that their reservation has started		
<b>Trigger:</b>	The current time and date is near the reservation start time and date		
<b>Preconditions:</b>	1. User is logged in. 2. User has a scheduled reservation in the system.		
<b>Postconditions:</b>	1. All staff involved with the reservation are notified that the reservation details have been changed and receive all necessary information. 2. The reservation in the database is marked as active 3. A log is made indicating that there was a change made to a pre-existing reservation.		
<b>Normal Flow:</b>	1. Steps 1-4 from UC 25.2 View Reservation: 1. The user logs into their account and navigates to the “Reservation” section. 2. The system shows the user a list of all their active reservations. 3. The user clicks on one of the options. 4. The system gets the data about the chosen reservation and presents it to the user. 2. The user is given the option to confirm that the reservation vehicle and driver have arrived and are ready for use 3. The user selects the aforementioned option to confirm 4. The system sends an email confirming the reservation start		
<b>Alternative Flows:</b>	2a. In step 3, if the user chooses not to verify the reservation start 2a.a. If user does not intend to verify reservation start 1. Contact information is provided alongside reservation details 2. User uses contact information to report reason for not verifying reservation start 2a.b. If an arbitrary amount of time passes with user not verifying reservation and not reporting reason for not verifying reservation start 1. User is sent email reminder to verify reservation or report reason for not verifying 2. If an arbitrarily larger amount of time passes with user not verifying reservation, reservation is automatically canceled and marked as not having start verified 3. User is sent email explaining reservation cancellation		
<b>Exceptions:</b>	5a. In step 1.2, 1.4, or 3 of the normal flow, if the database is unable to be reached 1. System will tell user that there was an error reaching the database and to try again later.		

<b>Includes:</b>	- View Reservation
<b>Frequency of Use:</b>	At specific times
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	1. User has an active account. 2. User understands English.
<b>Notes and Issues:</b>	[List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.

## View a Reservation for a Vehicle with a Driver

<b>Use Case ID:</b>	UC 02.2		
<b>Use Case Name:</b>	View Charter Reservation		
<b>Created By:</b>	Isabella Rosenbohm	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	9/10/2023	<b>Last Revision Date:</b>	9/16/2023
<b>Actors:</b>	Charter		
<b>Description:</b>	A user wants to view the details of their already existing reservation.		
<b>Trigger:</b>	The user opens the details of their reservation.		
<b>Preconditions:</b>	1. User is logged in. 2. The user already has an existing reservation. 3. The current time is before or during the time of the reservation.		
<b>Postconditions:</b>	1. The user has information on the details of their reservation.		
<b>Normal Flow:</b>	1. The user logs into their account and navigates to the “Reservation” section. 2. The system shows the user a list of all their active reservations. 3. The user clicks on one of the options. 4. The system gets the data about the chosen reservation and presents it to the user.		
<b>Alternative Flows:</b>	3a. In step 3 of the normal flow, if the user chooses the back option The system will redirect the user to the home page.		
<b>Exceptions:</b>	2a. In step 2 or 4 of the normal flow, if the database is unable to be reached 1. The system will tell user that there was an error reaching the database and to try again later.		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	1. User has an active account. 2. User understands English.		
<b>Notes and Issues:</b>	[List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.  1. Possibility of using a code that the user receives in their confirmation email that would be used to look up their reservation in the system.		

## Update an Existing Reservation for a Vehicle with a Driver

<b>Use Case ID:</b>	UC 02.3		
<b>Use Case Name:</b>	Update Charter Reservation		
<b>Created By:</b>	Isabella Rosenbohm	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	9/10/2023	<b>Last Revision Date:</b>	11/7/23
<b>Actors:</b>	Charter		
<b>Description:</b>	A user needs to make a change to a currently existing reservation for a charter vehicle with a driver.		
<b>Trigger:</b>	The user initiates the process of editing a reservation.		
<b>Preconditions:</b>	1. User is logged in. 2. User has an active reservation. 3. Reservation not approved/denied		
<b>Postconditions:</b>	1. The vehicle and driver are reserved for the new specified date and time. 2. All staff involved with the reservation are notified that the reservation details have been changed and receive all necessary information. 3. The new reservation information is stored within the system. 4. A log is made indicating that there was a change made to a preexisting reservation.		
<b>Normal Flow:</b>	1. Steps 1-4 from UC 25.2 View Reservation: 1. The user logs into their account and navigates to the "Reservation" section. 2. The system shows the user a list of all their active reservations. 3. The user clicks on one of the options. 4. The system gets the data about the chosen reservation and presents it to the user. 2. Upon viewing reservation details, the system provides user with an option to edit the reservation. 3. User chooses to edit the reservation. 4. Steps 2-10 from UC 25.1 Create Reservation: 1. The user specifies that they will need to reserve a driver 2. The user is shown a list of dates and times on which vehicles will be available to start a reservation on 3. The user selects a start date and time. 4. The system presents a list of vehicles with drivers available to start on the selected date and time to the user, along with the latest available end dates and times. 5. The user selects one of the options. 6. The user enters the pick-up and drop-off locations. 7. The system presents the details for the user to review and prompts the user to either confirm the reservation or go back to editing. 8. The user confirms the reservation. 9. The system processes the request and adds the necessary information to the database. 10. The user is shown some sort of confirmation that their request was processed successfully.		

<b>Alternative Flows:</b>	<p>1. Reservation has been approved/denied:</p> <p>1. Exit normal flow and enter UC 25.1 Create Reservation with previous information populated.</p> <p>4.7a. In step 4.8 of the normal flow, if the user chooses to not confirm the reservation</p> <ul style="list-style-type: none"> <li>1. User chooses to go back to editing.</li> <li>2. System allows the user to select a different vehicle from the list of available options.</li> <li>3. Use case returns to step 4.7.</li> </ul>
<b>Exceptions:</b>	<p>4.2a. In step 4.2 or 4.4 of the normal flow, if the database is unable to be reached</p> <p>The system will tell user that there was an error reaching the database and to try again later.</p>
<b>Includes:</b>	<ul style="list-style-type: none"> <li>- View Reservation</li> <li>- Create Reservation</li> </ul>
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	<ul style="list-style-type: none"> <li>1. User has an active account.</li> <li>2. User understands English.</li> </ul>
<b>Notes and Issues:</b>	<ul style="list-style-type: none"> <li>1. Possibly send some sort of confirmation email to the user upon successful reservation?</li> </ul>

## Cancel a Reservation for a Vehicle with a Driver

<b>Use Case ID:</b>	UC 02.4		
<b>Use Case Name:</b>	Cancel Reservation		
<b>Created By:</b>	Isabella Rosenbohm	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/10/2023	<b>Last Revision Date:</b>	11/07/2023
<b>Actors:</b>	Charters		
<b>Description:</b>	A user needs to cancel their reservation for a charter vehicle with a driver.		
<b>Trigger:</b>	The user initiates the process of canceling a reservation.		
<b>Preconditions:</b>	1. User is logged in. 2. User has an active reservation in the system. 3. Reservation not yet approved/denied.		
<b>Postconditions:</b>	1. All staff involved with the reservation are notified that the reservation details have been changed and receive all necessary information. 2. The canceled reservation is archived in the database. 3. A log is made indicating that there was a change made to a preexisting reservation.		
<b>Normal Flow:</b>	1. Steps 1-4 from UC 25.2 View Reservation: 1. The user logs into their account and navigates to the "Reservation" section. 2. The system shows the user a list of all their active reservations. 3. The user clicks on one of the options. 4. The system gets the data about the chosen reservation and presents it to the user. 2. The user chooses the option to cancel the reservation. 3. The system asks if the user is sure that they want to cancel the reservation. 4. User selects yes. 5. System processes the request and tells the user that the reservation has been canceled successfully. 6. The user is shown some sort of confirmation that their request was processed successfully.		
<b>Alternative Flows:</b>	2a. In step 1.3, 2, and 4 of the normal flow, if the user chooses to go back and not cancel the reservation  The system will take the user back to the view reservation page.  No changes to the reservation will be made.		
<b>Exceptions:</b>	5a. In step 1.2, 1.4, 5 or 9 of the normal flow, if the database is unable to be reached 1. The system will tell user that there was an error reaching the database and to try again later. 2a. The reservation has already been approved.		

	<ol style="list-style-type: none"> <li>1. System informs the user that reservation cannot be canceled at this time and provides feedback as to what to do next.</li> <li>2. Return to Step 1 of normal flow.</li> </ol>
<b>Includes:</b>	- View Reservation
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>1. User has an active account.</li> <li>2. User understands English.</li> </ol>
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Possibly send some sort of confirmation email to the user upon successful cancellation of the reservation?</li> <li>2. Some kind of buffer between the reservation day and canceling it? i.e. User will not be able to cancel reservation if there are less than 24 hours until the start of the reservation time.</li> </ol>

## Verify a Reservation End for a Chartered Vehicle with a Driver

<b>Use Case ID:</b>	UC 02.6		
<b>Use Case Name:</b>	Verify Reservation Ended		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	9/17/2023	<b>Last Revision Date:</b>	11/07/23
<b>Actors:</b>	Charter		
<b>Description:</b>	A user verifies that their reservation has ended		
<b>Trigger:</b>	The current time and date are near the reservation scheduled end time and date		
<b>Preconditions:</b>	1. User is logged in. 2. User has an active reservation in the system. 3. Vehicle has been reported returned		
<b>Postconditions:</b>	1. All staff involved with the reservation are notified that the reservation details have been changed and receive all necessary information. 2. The reservation in the database is marked as ended 3. A log is made indicating that there was a change made to a preexisting reservation.		
<b>Normal Flow:</b>	1. Steps 1-4 from UC 25.2 View Reservation: 1. The user logs into their account and navigates to the "Reservation" section. 2. The system shows the user a list of all their active reservations. 3. The user clicks on one of the options. 4. The system gets the data about the chosen reservation and presents it to the user. 2. The user is given the option to confirm that the reservation vehicle and driver have been returned 3. The user selects the aforementioned option to confirm 4. The system sends an email confirming the reservation end		
<b>Alternative Flows:</b>	2a. In step 3, if the user chooses not to verify the reservation end 2a.a. If user does not intend to verify reservation end 1. Contact information is provided alongside reservation details 2. User uses contact information to report reason for not verifying reservation end 2a.b. If an arbitrary amount of time passes with user not verifying reservation and not reporting reason for not verifying 1. User is sent email reminder to verify reservation or report reason for not verifying		
<b>Exceptions:</b>	5a. In step 1.2, 1.4, or 3 of the normal flow, if the database is unable to be reached 1. The system will tell user that there was an error reaching the database and to try again later.		
<b>Includes:</b>	- View Reservation		
<b>Frequency of Use:</b>	At specific times		
<b>Special Requirements:</b>	N/A		

<b>Assumptions:</b>	1. User has an active account. 2. User understands English.
<b>Notes and Issues:</b>	[List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.

## Riders – Charter without Driver

### Renting Vehicles

<b>Use Case ID:</b>	CWOD.01.01		
<b>Use Case Name:</b>	Reserve Vehicle without Driver		
<b>Created By:</b>	Parker Svoboda	<b>Last Updated By:</b>	Michael Springer
<b>Date Created:</b>	9/03/23	<b>Last Revision Date:</b>	2023/09/11
<b>Actors:</b>	Charterer Charter without driver		
<b>Description:</b>	The charterer wants to rent a vehicle for their business so that employees can get to where they need to be.		
<b>Trigger:</b>	The charterer visits the establishment, or rents the vehicle(s) on the application		
<b>Preconditions:</b>	1. Charterer must have account 2. Charterer account has system permission to charter vehicle 3. Driver(s) associated with reservation must have system approval to drive rented vehicles 4. The trip driver(s) must be confirmed to have a driver's license with a passenger endorsement if the vehicle is a bus.		
<b>Postconditions:</b>	1. System contains records of charter reservation. 2. System assigns default available resources to reservation, e.g., bus, driver. 3. The charterer or driver(s) will be able to receive keys at the office. 4. The charterer or driver(s) will be able to receive the vehicle(s) in the lot.		
<b>Normal Flow:</b>	1. The Charterer selects the [Rent Without Driver] button. 2. Charterer enters vehicle needs (no. passengers, etc.) 3. Charterer schedules when and how long the vehicle is needed 4. Charterer selects any additional accommodations and resources 5. Charterer enters necessary personal information fields (dept., etc.) 6. Charterer enters driver details 7. Charterer submits rental form 8. System validates resource availability and other rules requirements 9. System displays confirmation message summarizing details 10. Charterer confirms 11. System logs reservation 12. System sends notifications to designated parties		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Drivers not validated]</b>	7a. In step 7 of the normal flow if the selected driver(s) account isn't confirmed to have a driver's license or necessary passenger endorsement <ol style="list-style-type: none"> <li>System will prompt the if the charterer wants to confirm their license</li> <li>Charterer selects yes</li> <li>System redirects to the license confirmation feature</li> </ol> 2b. In step 2 of the normal flow if the charterer's account isn't confirmed to have a driver's license with a passenger's endorsement <ol style="list-style-type: none"> <li>System will prompt the if the charterer wants to confirm their license</li> </ol>		

	<p>2. Charterer selects no</p> <p>3. System displays dialog box with link to “Charter with Driver” option</p>
<b>Alternative Flows:</b> <b>[Alternative Flow 2 – Reservation Conflict]</b>	<p>9a. In step 9 of the normal flow a resource or rules conflict prevents guarantee of the reservation</p> <p>1. System displays dialog box with relevant error message</p>
<b>Exceptions:</b>	<p>12a. In step 12 of the normal flow, network or system error prevents completion of reservation</p> <p>1. System displays an error message</p> <p>2. User closes dialog</p> <p>3. System re-displays completed form</p> <p>4. Return to step 8 of normal flow.</p>
<b>Includes:</b>	3.3 Hiring Drivers License Confirmation
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	Passenger Endorsement License. Users must be at least 21 years old to rent certain types of vehicles. Business rules will vary, e.g., driver training, background checks, etc.
<b>Assumptions:</b>	Granting permission to drivers and charterers will be handled outside of the system, with that
<b>Notes and Issues:</b>	<p>1. How could a charter be confirmed to have a passenger endorsement on their driver’s license? How can we be sure of it?</p> <p>a. Photo of both license and face?</p> <p>b. Physical meeting?</p> <p>2. How should the interface be presented?</p>

## Modify Vehicle Reservation

<b>Use Case ID:</b>	CWOD.01.04		
<b>Use Case Name:</b>	Modify Reservation		
<b>Created By:</b>	Parker Svoboda	<b>Last Updated By:</b>	Michael Springer
<b>Date Created:</b>	9/03/2023	<b>Last Revision Date:</b>	2023/09/11
<b>Actors:</b>	Charter Without Driver		
<b>Description:</b>	User wants to modify their reservation		
<b>Trigger:</b>	User opens the application		
<b>Preconditions:</b>	1. User reserved a vehicle 2. It is not after or during the scheduled rental 3. Reservation not yet approved/denied.		
<b>Postconditions:</b>	1. Users modify reservations, additional charges or discounts are applied to lease price fairly on how much the designated time of use is changed.		
<b>Normal Flow:</b>	1. The user selects the [My Reservations] tab. (See 3.7) 2. The charterer selects the vehicle(s) they reserved on the [My Reservations] list via checkboxes. 3. The user selects the [Modify Reservation] button at the top of the page. 4. The user is redirected to a page where they can modify the reservation 5. The user clicks [Submit Modification] button 6. System logs the requested modifications 7. System sends a notification to dispatch regarding the modifications 8. The user is redirected to the [My Reservations] page		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 –No Reservation/ Not Logged In]</b>	1a. In step 1 of the normal flow if User has no reserved or rented vehicles or if the user is not logged in. 1. [My Reservations] tab is hidden 2. Use case ends 1b. Reservation already approved/denied: 1. Reservation is canceled. 2. Exit current use case. 3. Create new reservations with information auto populated with previous reservations.		
<b>Exceptions:</b>	none		
<b>Includes:</b>	3.7 View Reserved Vehicles		
<b>Frequency of Use:</b>	on demand		
<b>Special Requirements:</b>	none		
<b>Assumptions:</b>	The user has a registered account and is logged in.		
<b>Notes and Issues:</b>			

## Remove Reserved Vehicles

<b>Use Case ID:</b>	CWOD.01.05		
<b>Use Case Name:</b>	Remove Reservation		
<b>Created By:</b>	Parker Svoboda	<b>Last Updated By:</b>	Michael Springer
<b>Date Created:</b>	9/03/2023	<b>Last Revision Date:</b>	2023/09/18
<b>Actors:</b>	Charter Without Driver		
<b>Description:</b>	User wants to cancel their reservation because they no longer have a need for a vehicle		
<b>Trigger:</b>	User opens the application		
<b>Preconditions:</b>	1. User reserved a vehicle 2. The cancellation window is still open		
<b>Postconditions:</b>	1. System logs cancellation 2. System updates reservation status in associated systems (fleet, inventory, etc.)		
<b>Normal Flow:</b>	1. Charterer selects the [My Reservations] tab. (See 3.7) 2. The charterer selects the vehicle(s) they reserved on the [My Reservations] list via checkboxes. This will allow for multiple reservations to be canceled. 3. The user selects the [Cancel Reservation] button at the top of the page. 4. The user is prompted for a reason for cancellation of reservation, for business purposes. 5. The [My Reservations] page is refreshed		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow if User has no reserved or rented vehicles or if the user is not logged in. <ol style="list-style-type: none"> <li>[My Rentals] tab is hidden</li> <li>Use case ends</li> </ol>		
<b>Exceptions:</b>	3a. During step 3 of normal flow, the cancelation window for the reservation is already closed <ol style="list-style-type: none"> <li>System visually (and for assistive tech) marks the cancellation button as deactivated</li> <li>User clicks on cancel button</li> <li>System displays an error message with dispatch or customer service contact information</li> </ol>		
<b>Includes:</b>	3.7 View Reserved Vehicles		
<b>Frequency of Use:</b>	on demand		
<b>Special Requirements:</b>	none		
<b>Assumptions:</b>	The user has a registered account and is logged in.		
<b>Notes and Issues:</b>			

## Return Vehicles without Driver

<b>Use Case ID:</b>	CWOD.01.06		
<b>Use Case Name:</b>	Return Vehicle without Driver		
<b>Created By:</b>	Parker Svoboda	<b>Last Updated By:</b>	Michael Springer
<b>Date Created:</b>	9/05/2023	<b>Last Revision Date:</b>	2023/09/11
<b>Actors:</b>	Charter without driver		
<b>Description:</b>	Charterer returns a vehicle after using it to get riders to where they need to be.		
<b>Trigger:</b>	The charterer drives the vehicle to the parking lot and parks in a numbered lot.		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. The charterer must have rented the vehicle without a driver.</li> <li>2. The charterer must have the keys.</li> <li>3. The charterer must have the vehicle.</li> <li>4. The charterer must follow the normal flow before returning the keys</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Attendant or facility has custody of keys and vehicle</li> <li>2. System is logged with vehicle return</li> <li>3. Vehicle is queued for any default post-return processes (cleaning etc.)</li> <li>4. Vehicle is queued for any processes designated by the attendant or other intaking party</li> </ul>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The charterer opens the app.</li> <li>2. The charterer selects the [My Rentals] tab. (See 3.7)</li> <li>3. The charterer selects the vehicle(s) they rented via checkboxes.</li> <li>4. The charterer selects the [return] button at the top of the page.</li> <li>5. The charterer is redirected to a survey to report the condition of the vehicle.</li> <li>6. The charterer fills out the survey, like lot number, damages, and other issues.</li> <li>7. The charterer selects [submit].</li> <li>8. The charterer leaves the vehicle after submission is sent.</li> <li>9. The charterer visits the office.</li> <li>10. The charterer waits for the front desk to receive or find the survey.</li> <li>11. Dispatch confirms receipt of return form</li> <li>12. Charterer returns keys</li> <li>13. Dispatch confirms return</li> <li>14. System logs return</li> <li>15. System queues vehicle for any default post-return processes</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<p>7a. in step 7 if network or system error prevents submission</p> <ol style="list-style-type: none"> <li>1. submission will be stored in the app and will prompt the charterer to go to the front desk in the building, connect to the internet and then go to [My Rentals] tab and select, [Resubmit survey] which will appear if the app detects a stored survey.</li> <li>2. the charterer visits the office</li> <li>3. The charterer connects to the internet and follows the instructions that were given.</li> <li>4. Use Case resume on step 10.</li> </ol> <p>10a. In step 10 of the normal flow, if the charterer hasn't filled out the survey</p> <ol style="list-style-type: none"> <li>1. Return to step 1</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Manual log]	<p>7a. in step 7 if network or system error prevents submission</p> <ol style="list-style-type: none"> <li>1. Charterer returns to dispatch</li> <li>2. Dispatch proceeds with "Return a vehicle borrow"</li> </ol>		

	<p>3. Dispatch manually fills form fields</p> <p>4. Return to step 13</p>
<b>Exceptions:</b>	None
<b>Includes:</b>	<p>3.7 View Reserved Vehicles</p> <p>7.2 Return a Vehicle Borrow</p>
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	None
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	<p>There are numerous ways to combine the steps of the charterer returning the vehicle and the dispatch (or other intake staff) going through the steps of their side of the return process.</p> <p>As this is further developed, it will likely rely on a set of flags in a return queue—waiting for inspection, waiting for cleaning, etc.</p>

## View Available Vehicles

<b>Use Case ID:</b>	3.4		
<b>Use Case Name:</b>	View Available Vehicles		
<b>Created By:</b>	Parker Svoboda	<b>Last Updated By:</b>	Michael Springer
<b>Date Created:</b>	9/3/23	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Charter Without Driver		
<b>Description:</b>	User wants to view available vehicles that they can rent		
<b>Trigger:</b>	User opens the app		
<b>Preconditions:</b>	1. User is logged in.		
<b>Postconditions:</b>	1. User sees a list of available vehicles.		
<b>Normal Flow:</b>	1. User selects the [rental] tab 2. User views available vehicles		
<b>Alternative Flows:</b> <i>[Alternative Flow 1 – Not in Network]</i>	1a. In step 1 of the normal flow, if the user isn't logged in 1. User is brought to the login page (see UC-22.1.1) 2. User selects [login] 3. Use case resumes on step 2  1b. In step 1 of the normal flow, if the user isn't logged in and doesn't have an account 1. User is brought to the login page 2. User selects sign up 3. User is redirected to the sign-up page (see UC-22.2.1) 4. User selects [Sign up] button 5. Use case resumes on step 2 of alternate flow		
<b>Exceptions:</b>	None		
<b>Includes:</b>	UC-22.1.1 Create Account UC-22.2.2 Log in		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	1. Must have at least 3 vehicles available in the database as placeholders. One for renting without driver, another with driver, and the last to be left available		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	The development team will need to settle on an approach to the reservation process—whether it allows all users who make reservations to choose a specific vehicle or not.  If charterers can't choose a specific vehicle (have one designated by dispatch or other staff instead), then this use case wouldn't exist.  If charters can choose a vehicle, then this use case may or may not share features with dispatchers, etc.		

## Rider –Route Rider

### Current Route Tracking

<b>Use Case ID:</b>	18.1		
<b>Use Case Name:</b>	Current Route Tracking		
<b>Created By:</b>	Jared Roberts	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-03	<b>Last Revision Date:</b>	2023/9/11
<b>Actors:</b>	Route Rider		
<b>Description:</b>	Let's Rider see the current position of the bus they're on		
<b>Trigger:</b>	Rider wants to see their status		
<b>Preconditions:</b>	1. Rider is on bus 2. Rider is signed into app 3. Rider has wifi/data		
<b>Postconditions:</b>	1. Rider see's where they are located		
<b>Normal Flow:</b>	1. Rider gets on bus 2. Rider signs into app 3. Rider clicks on tracking tab 4. Rider see's current position of bus		
<b>Alternative Flows:</b>	NA		
<b>Exceptions:</b>	1a. In step 2 of the normal flow, if the customer enters invalid information 1. System will prompt user Cleaner to reenter their information 2. Route Rider will assumedly give the correct information  3. Use Case resumes on step 3 of normal flow		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	As often as they want assuming they are on the ride		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	1. Rider can navigate to tracking tab 2. Rider speaks English		
<b>Notes and Issues:</b>			

## Route Scheduling

<b>Use Case ID:</b>	18.2		
<b>Use Case Name:</b>	Route Scheduling		
<b>Created By:</b>	Jared Roberts	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	2023-09-03	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Route Rider		
<b>Description:</b>	Allows Rider to schedule a ride		
<b>Trigger:</b>	Rider needs a ride		
<b>Preconditions:</b>	1. Rider has the app		
<b>Postconditions:</b>	1. Rider is scheduled for a ride 2. Rider waits for ride		
<b>Normal Flow:</b>	1. Rider is signed in 2. Rider searches for available routes 3. Rider picks ride that is their best fit		
<b>Alternative Flows:</b>	NA		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, if the customer enters invalid information 1. System will prompt Cleaner to reenter their information 2. Route Rider will assumedly give the correct information 3. Use Case resumes on step 3 of normal flow		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	As often as the Rider wants to ride		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	1. Rider can navigate to scheduling tab 2. Rider speaks English 3. Rider has money		
<b>Notes and Issues:</b>			

Feature Name View Available Routes

<b>Use Case ID:</b>	18.3		
<b>Use Case Name:</b>	View available routes		
<b>Created By:</b>	Jared Roberts	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-03	<b>Last Revision Date:</b>	9/11/2023
<b>Actors:</b>	Route Rider		
<b>Description:</b>	Rider see's which routes are available for booking		
<b>Trigger:</b>	Rider is planning to schedule a ride		
<b>Preconditions:</b>	1. Rider is signed in		
<b>Postconditions:</b>	1. Rider has seen available routes		
<b>Normal Flow:</b>	1. Rider signs in 2. Rider views available routes		
<b>Alternative Flows:</b>	NA		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, if the customer enters invalid information 1. System will prompt Cleaner to reenter their information 2. Route Rider will assumedly give the correct information 3. Use Case resumes on step 3 of normal flow		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	As often as the Rider wants to see the available routes		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	1. Rider can speak English		
<b>Notes and Issues:</b>			

## Riders – Ride Scheduler

## **View Active Transit Services**

<b>Use Case ID:</b>	UC-RS.01.01		
<b>Use Case Name:</b>	View Active Transit Services		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Ride Requester		
<b>Description:</b>	A user is shown all the active transit services		
<b>Trigger:</b>	The user has communicated interest in scheduling a ride		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>User is logged in</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>User is shown all active transit services</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>User selects the [Transit Services] tab. User is shown all active transit services</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> <li>a. if user is not logged in             <ol style="list-style-type: none"> <li>User is redirected to log in page</li> <li>User logs in</li> <li>Use case continues on step 2</li> </ol> </li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>a. If no active services exist             <ol style="list-style-type: none"> <li>User is presented an error</li> <li>User is informed that there are no active services</li> <li>User is directed away from page</li> </ol> </li> </ol>		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## View Details of Active Service

<b>Use Case ID:</b>	UC-RS.01.02		
<b>Use Case Name:</b>	View Details of Active Service		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Ride Scheduler		
<b>Description:</b>	A user is shown the details of an active service including the number of running vehicles, areas serviced, and running times		
<b>Trigger:</b>	The user wants to schedule a ride		
<b>Preconditions:</b>	1. The requested service to view details of is active		
<b>Postconditions:</b>	1. User is shown details of transit service		
<b>Normal Flow:</b>	1. User selects a specific transit service on the [Transit Services] tab 2. User is shown details of transit service		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Not in Network]</b>	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	RS-01		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## Enter a Start and End Location / Receive Applicable Active Services

<b>Use Case ID:</b>	UC-RS.01.03		
<b>Use Case Name:</b>	Enter a Start and End Location and Receive Applicable Active Services		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Ride Scheduler		
<b>Description:</b>	A user enters the start and end locations which they are requesting a ride for and is shown the services which service those locations		
<b>Trigger:</b>	User wants to see which active services meet their needs without checking each service		
<b>Preconditions:</b>	User is logged in		
<b>Postconditions:</b>	All available and applicable services are shown		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User selects the [Transit Services] tab.</li> <li>2. User enters a desired start and end location for their ride</li> <li>3. User clicks the [Submit] button</li> <li>4. User is shown a filtered list of services which are active and cover both the start and end locations</li> </ol>		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Not in Network]</b>	<p>1a. If the user enters a desired start location but not an end location  User requests to see applicable services  User is shown a filtered list of services which are running and cover the start location</p> <p>1b. If the user enters a desired end location but not a start location  User requests to see applicable services  User is shown a filtered list of services which are running and cover the end location</p>		
<b>Exceptions:</b>	<p>4a. if no services active cover the start and end location</p> <ol style="list-style-type: none"> <li>1. User gets an empty list with a message saying that no services that cover the specified locations are available.</li> </ol>		
<b>Includes:</b>	RS-01		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## Schedule a Ride

<b>Use Case ID:</b>	UC-RS.01.04		
<b>Use Case Name:</b>	Schedule a Ride		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Ride Scheduler		
<b>Description:</b>	User sends a request to schedule a ride		
<b>Trigger:</b>	User decides which active service they want to request a ride for		
<b>Preconditions:</b>	There is at least one running service At least one service meets the user's ride needs		
<b>Postconditions:</b>	A ride is scheduled		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user selects from the list of available services (see 26-01)</li> <li>2. The user is shown all the available time slots segmented by day</li> <li>3. The user selects a time slot</li> <li>4. The user is prompted to confirm that they want to schedule a ride for the selected time slot</li> <li>5. The user confirms the prompt</li> <li>6. A ride is scheduled</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	26-01		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## Request Information on Scheduled Ride

<b>Use Case ID:</b>	UC-RS.02.01		
<b>Use Case Name:</b>	Request Info on Scheduled Ride		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Ride Requester		
<b>Description:</b>	Show user a time and date for scheduled ride		
<b>Trigger:</b>			
<b>Preconditions:</b>	1. User has scheduled a ride 2. User is logged in		
<b>Postconditions:</b>	1. User gets information on scheduled ride		
<b>Normal Flow:</b>	1. User is logged in 2. User navigates to view scheduled rides 3. User is shown the time and date of the scheduled ride		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	RS-01 RS-04		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## Riders – Ride Requester

### View Running Transit Services

<b>Use Case ID:</b>	UC-RR.01.01		
<b>Use Case Name:</b>	View Running Transit Services		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-17
<b>Actors:</b>	Ride Requester		
<b>Description:</b>	A user is shown all the running transit services		
<b>Trigger:</b>	The user has communicated interest in promptly getting a ride		
<b>Preconditions:</b>	N/A		
<b>Postconditions:</b>	N/A		
<b>Normal Flow:</b>	1. User indicates that they want to request a ride for as soon as possible 2. User selects the [Transit Services] tab. 3. User is shown all running transit services		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Not in Network]</b>	None		
<b>Exceptions:</b>	2a. If no active services exist 1. User is presented an error 2. User is informed that there are no active services 3. User is directed away from page 2b. If no active services are running 1. User is presented an error 2. User is informed there are no running services 3. User is asked if they want to schedule a service instead; otherwise, they are directed away from page		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## View Details of Running Service

<b>Use Case ID:</b>	UC-RR.01.02		
<b>Use Case Name:</b>	View Details of Running Service		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-17
<b>Actors:</b>	Ride Requester		
<b>Description:</b>	A user is shown the details of a running service including the number of running vehicles, areas serviced, and running times		
<b>Trigger:</b>	The user has communicated interest in promptly getting a ride		
<b>Preconditions:</b>	The requested service to view details of is active and running User has communicated interest in requesting a ride User has indicated that they want to request a ride for as soon as possible		
<b>Postconditions:</b>	User is shown details of transit service		
<b>Normal Flow:</b>	1. User selects a specific transit service on the [Transit Services] tab 2. User is shown details of transit service		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Not in Network]</b>	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	RR-01		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## Enter a Start and End Location / Receive Applicable Running Services

<b>Use Case ID:</b>	UC-RR.01.03		
<b>Use Case Name:</b>	Enter a Start and End Location and Receive Applicable Running Services		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-17
<b>Actors:</b>	Ride Requester		
<b>Description:</b>	A user enters the start and end locations which they are requesting a ride for and is shown the services which service those locations		
<b>Trigger:</b>	User wants to see which services meet their needs without checking each service		
<b>Preconditions:</b>	User has communicated interest in requesting a ride User has indicated they want to request a ride for as soon as possible User is logged in		
<b>Postconditions:</b>	All available and applicable services are shown		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User enters a desired start and end location for their ride</li> <li>1. User clicks the [Submit] button</li> <li>2. User is shown a filtered list of services which are running and cover both the start and end locations</li> </ol>		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Not in Network]</b>	<p>1a. If the user enters a desired start location but not an end location  User requests to see applicable services  User is shown a filtered list of services which are running and cover the start location</p> <p>1b. If the user enters a desired end location but not a start location  User requests to see applicable services  User is shown a filtered list of services which are running and cover the end location</p> <p>4a. if no services active cover the start and end location</p> <ol style="list-style-type: none"> <li>1. User gets an empty list with a message saying that no services that cover the specified locations are available.</li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>	RR-01		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## Request an Immediate Ride

<b>Use Case ID:</b>	UC-RR.01.04		
<b>Use Case Name:</b>	Request an immediate ride		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Parker Svoboda
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-17
<b>Actors:</b>	Ride Requester		
<b>Description:</b>	User sends a request for a ride ASAP		
<b>Trigger:</b>	User decides which running service they want to request a ride for		
<b>Preconditions:</b>	There is at least one running service At least one service meets the user's ride needs		
<b>Postconditions:</b>	A ride is scheduled The user is shown the estimated for pickup The user is shown the estimated time for dropoff		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user selects from the list of available services(see 25-01)</li> <li>2. The user is shown the estimated time of arrival and time of drop off</li> <li>3. The user is prompted to confirm that they want to request a ride for ASAP</li> <li>4. The user confirms the prompt</li> <li>5. A ride is scheduled</li> <li>6. The user is provided details on the estimated pickup and drop-off times</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	25-01		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## Request Time of Next Available Ride

<b>Use Case ID:</b>	UC-RR.02.01		
<b>Use Case Name:</b>	Request Time of Next Available Ride		
<b>Created By:</b>	Jared Hutton	<b>Last Updated By:</b>	Jared Hutton
<b>Date Created:</b>	2023-09-11	<b>Last Revision Date:</b>	2023-09-11
<b>Actors:</b>	Ride Requester		
<b>Description:</b>	Show user a time for next available ride		
<b>Trigger:</b>	User wants to see the estimated times for a requested ride		
<b>Preconditions:</b>	User has requested a ride		
<b>Postconditions:</b>	The time of the next available ride is shown		
<b>Normal Flow:</b>	7. User is logged in 8. User navigates to view currently requested ride 9. User is shown the estimated pickup and drop-off timers for requested ride		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	RR-01 RR-04		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>			

## Riders – Passenger

### Current Route Tracking

<b>Use Case ID:</b>	UC-PAS.01.01		
<b>Use Case Name:</b>	Current Route Tracking		
<b>Created By:</b>	Jared Roberts	<b>Last Updated By:</b>	Steven Sanchez
<b>Date Created:</b>	2023-09-03	<b>Last Revision Date:</b>	2023-10-29
<b>Actors:</b>	Passenger		
<b>Description:</b>	Let's passenger see the current position of the bus they're on		
<b>Trigger:</b>	Passenger wants to see their status		
<b>Preconditions:</b>	1. Passenger is on bus 2. Passenger is signed into app 3. Passenger has Wi-Fi/data		
<b>Postconditions:</b>	1. Passenger sees where they are located		
<b>Normal Flow:</b>	1. Passenger gets on bus 2. Passenger signs into app 3. Passenger clicks on tracking tab 4. Passenger see's current position of bus		
<b>Alternative Flows:</b>	NA		
<b>Exceptions:</b>	1a. In step 2 of the normal flow, if the customer enters invalid information 1. System will prompt user to reenter their information 2. Passenger will assumedly give the correct information  3. Use Case resumes on step 3 of normal flow		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	As often as they want assuming they are on the ride		
<b>Special Requirements:</b>			
<b>Assumptions:</b>	1. Passenger can navigate to tracking tab 2. Passenger speaks English		
<b>Notes and Issues:</b>			

## Route Scheduling

<b>Use Case ID:</b>	UC-PAS.01.02		
<b>Use Case Name:</b>	Route Scheduling		
<b>Created By:</b>	Jared Roberts	<b>Last Updated By:</b>	Steven Sanchez
<b>Date Created:</b>	2023-09-03	<b>Last Revision Date:</b>	2023-10-29
<b>Actors:</b>	Passenger		
<b>Description:</b>	Allows Passenger to schedule a ride		
<b>Trigger:</b>	Passenger needs a ride		
<b>Preconditions:</b>	1. Passenger has the app		
<b>Postconditions:</b>	1. Passenger is scheduled for a ride 2. Passenger waits for ride		
<b>Normal Flow:</b>	1. Passenger is signed in 2. Passenger searches for available routes 3. Passenger picks ride that is their best fit		
<b>Alternative Flows:</b>	NA		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, if the customer enters invalid information 1. System will prompt user to reenter their information 2. Passenger will assumedly give the correct information 3. Use Case resumes on step 3 of normal flow		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	As often as the Passenger wants to ride		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	1. Passenger can navigate to scheduling tab 2. Passenger speaks English 3. Passenger has money		
<b>Notes and Issues:</b>			

## View Available Routes

<b>Use Case ID:</b>	UC-PAS.01.03		
<b>Use Case Name:</b>	View Available Routes		
<b>Created By:</b>	Jared Roberts	<b>Last Updated By:</b>	Steven Sanchez
<b>Date Created:</b>	2023-09-03	<b>Last Revision Date:</b>	2023-10-29
<b>Actors:</b>	Passenger		
<b>Description:</b>	Passenger sees which routes are available for booking		
<b>Trigger:</b>	Passenger is planning to schedule a ride		
<b>Preconditions:</b>	1. Passenger is signed in		
<b>Postconditions:</b>	1. Passenger has seen available routes		
<b>Normal Flow:</b>	1. Passenger signs in 2. Passenger views available routes		
<b>Alternative Flows:</b>	NA		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, if the customer enters invalid information 1. System will prompt User to reenter their information 2. Passenger will assumedly give the correct information 3. Use Case resumes on step 3 of normal flow		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	As often as the Passenger wants to see the available routes		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	1. Passenger can speak English		
<b>Notes and Issues:</b>			

## Riders – Person with Disability

Provide Ride Reviews / Feedback

<b>Use Case ID:</b>	UC-10.1		
<b>Use Case Name:</b>	Provide Ride Reviews/Feedback		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Everett DeVaux
<b>Date Created:</b>	09/08/2023	<b>Last Revision Date:</b>	09/17/2023
<b>Actors:</b>	Handicapped Elderly		
<b>Description:</b>	Provide Ride Reviews/Feedback. So that necessary changes that need to be made can be brought to the attention of the managers.		
<b>Trigger:</b>	At the end of the ride or reservation the customer will be prompted to leave a review.		
<b>Preconditions:</b>	1. Customer must set up a reservation and use it. 2. Customer will be prompted to leave review/feedback which is optional.		
<b>Postconditions:</b>	1. Customer leaves review/feedback. 2. Feedback will be saved and can be seen by the company.		
<b>Normal Flow:</b>	1. Customer makes a reservation. 2. Customer utilizes reservation. 3. The system validates if the customer has finished the reservation. 4. System prompts customers to leave review/feedback. 5. Customer leaves feedback. 6. System thanks customer.		
<b>Alternative Flows:</b>	4a. In step 4 of the normal flow if the customer decides to skip the review/feedback. The system will thank the customer.		
<b>Exceptions:</b>	2a. In step 2 of the normal flow, if fails to utilize reservation or cancels. 1. Reservation is canceled. 2. Message to customers thanking them for working with us. 3. The system goes back to the main menu.  5a. In step 5 of the normal flow if the customer does not enter anything into the feedback but submits. 1. System will request feedback. 2. The system thanks the customer. 3. The System returns to the main menu.		
<b>Includes:</b>	Reservation database Completion database		
<b>Frequency of Use:</b>	Once per completed reservation.		
<b>Special Requirements:</b>	Needs to identify whether a customer has finished their reservation.		
<b>Assumptions:</b>	The customer has completed the reservation and went to review it.		
<b>Notes and Issues:</b>	1. Customer just needs to have completed the reservation and the system must recognize that.		

## Schedule Future Ride

<b>Use Case ID:</b>	UC-10.2		
<b>Use Case Name:</b>	Schedule Ride		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Everett DeVaux
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Handicapped Elderly		
<b>Description:</b>	Schedule a ride in a suitable vehicle for my needs (including mobility devices and service animals).		
<b>Trigger:</b>	A customer decides they need to utilize our services in the future		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The customer must be either handicapped or a certain age.</li> <li>2. Customer can have a recognized special use case</li> <li>3. The customer set up a reservation with the above information for pickup.</li> <li>4. Customer makes payment. (Potentially optional)</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Customer will receive reservation and be completed.</li> <li>2. Customer account balance is reduced by the amount of the withdrawal and transaction fees. (Potentially optional)</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Customer starts reservation process.</li> <li>2. Customer enters handicapped/elderly information.</li> <li>3. The system prompts customers to enter if they have a service animal or mobility device.</li> <li>4. The system validates that the vehicle is available.</li> <li>5. The system validates customer input and appends it to the reservation</li> <li>6. The system prompts the user to set up payment.</li> <li>7. Customer enters payment information.</li> <li>8. System confirms reservation and saves information.</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<p>4a. In step 4 of the normal flow if no vehicles are available.</p> <ol style="list-style-type: none"> <li>1. The system will prompt customers to find a different time or day.</li> <li>2. Customer declines.</li> <li>3. Reservation is terminated.</li> <li>4. Use Case resumes on step 1 of normal flow.</li> </ol> <p>7a. In step 7 of the normal flow if the customer decides not to make a reservation.</p> <ol style="list-style-type: none"> <li>1. The system will prompt customers to accept reservations.</li> <li>2. Customer declines.</li> <li>3. Return to the previous screen or restart the reservation process.</li> </ol>		
<b>Exceptions:</b>	<p>5a. In step 5 of the normal flow if the customer enters an invalid PIN.</p> <ol style="list-style-type: none"> <li>1. Transactions are disapproved.</li> <li>2. Message to customer to re-enter PIN.</li> <li>3. Customer enters correct PIN.</li> <li>4. Use Case resumes on step 6 of normal flow.</li> </ol> <p>4a. In step 4 of the normal flow if the vehicle chosen becomes unavailable.</p> <ol style="list-style-type: none"> <li>1. The system will notify the customer that the vehicle is not available.</li> <li>2. Sends customer back to menu with vehicle selection.</li> </ol>		
<b>Includes:</b>	Customer database Reservation database Vehicle database		
<b>Frequency of Use:</b>	On-Demand		

<b>Special Requirements:</b>	Needs to be able to update Reservation database. Needs access to the Reservation database. Have the option to include Disability Accommodations.
<b>Assumptions:</b>	The customer has a valid payment method. The customer has an account or signed in as a guest.
<b>Notes and Issues:</b>	How many reservations can a customer make? What if a reservation is booked at the same time?

## Scheduled Ride with Accommodations

<b>Use Case ID:</b>	UC-10.3		
<b>Use Case Name:</b>	Schedule ride with Accommodations		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Everett DeVaux
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Handicapped		
<b>Description:</b>	Schedule a ride that accommodates my walking stick.		
<b>Trigger:</b>	Scheduling their ride with their disability accommodations as a text box that accepts input.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>Customer has a disability accommodation.</li> <li>The customer has included that information in the reservation process.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>Customer gets a vehicle that can accommodate them.</li> <li>Customer will be prompted with review/feedback.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Customer selects [ride share - Later]</li> <li>Customer inserts fundamental route info (date, time, locations)</li> <li>Customer enters disability and accommodation that is needed</li> <li>. System prompts customer to enter if they have a service animal or mobility device.</li> <li>The system validates that the vehicle is available.</li> <li>The system validates customer feedback and sends information to the driver.</li> <li>The system prompts the user to select a transaction type.</li> <li>Customer finishes reservation.</li> <li>Confirmation is shown by the system.</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> <li>In step 3 of the normal flow if the driver declines the reservation.             <ol style="list-style-type: none"> <li>The system will navigate to a new driver.</li> <li>The driver accepts.</li> <li>Use Case resumes on step 4.</li> </ol> </li> <li>In step 5 of the normal flow the customer cancels the reservation before finishing.             <ol style="list-style-type: none"> <li>The system will generate a “Are you sure?” prompt.</li> <li>Customer answers yes.</li> <li>System will take the customer back to the main menu.</li> </ol> </li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>In step 1 of the normal flow if the customer enters invalid characters.             <ol style="list-style-type: none"> <li>An error will occur.</li> <li>Message to customer to re-enter information with valid characters.</li> <li>Customer enters correct information.</li> <li>Use Case resumes on step 2 of normal flow.</li> </ol> </li> <li>In step 4 of the normal flow if the customer enters incorrect information for billing/card payment             <ol style="list-style-type: none"> <li>An error will occur.</li> <li>Message the customer to re-enter the invalid section.</li> <li>Customer enters correct information.</li> <li>Use Cases resumes on step 5 of normal flow.</li> </ol> </li> </ol>		
<b>Includes:</b>	Account database Driver database Reservation database Customer database Driver database		
<b>Frequency of Use:</b>	On-demand		
<b>Special Requirements:</b>	Needs to be able to pull information from our drivers database. Also needs to be able to save information based on customer.		

<b>Assumptions:</b>	The customer has a disability and needs accommodations.
<b>Notes and Issues:</b>	<ol style="list-style-type: none"><li>1. How many vehicles are available for accommodation?</li><li>2. What type of accommodations does the customer have that may affect available vehicles?</li></ol>

## Set Account Preferences

<b>Use Case ID:</b>	UC-10.4		
<b>Use Case Name:</b>	Account Preferences		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Everett DeVaux
<b>Date Created:</b>	9/8/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Handicapped		
<b>Description:</b>	Setup account preferences to only search for rides that can accommodate for the customers disability accommodation.		
<b>Trigger:</b>	The customer doesn't have to let the driver know every time what their accommodations are.		
<b>Preconditions:</b>	Customer has an account with us.		
<b>Postconditions:</b>	Customer can adjust the account to where their preferences are used.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Customer logs into the account.</li> <li>2. Customer goes to options.</li> <li>3. Customer selects [Account Preferences]</li> <li>4. The system will generate a list of textboxes or checkboxes that can be entered by customers.</li> <li>5. Customer confirms their selection.</li> <li>6. The system will be updated with new preferences.</li> </ol>		
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>5a. In step 5 of the normal flow if the customer does not accept preferences.</li> <li>1. Customer declines adjustments.</li> <li>2. System will terminate edits.</li> <li>3. The customer will be returned to the main menu.</li> <li>4. Use Case will end with step 5</li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1a. In step 1 of the normal flow if the customer inserts incorrect login information.             <ol style="list-style-type: none"> <li>1. Customer enters an invalid field.</li> <li>2. The system will prompt customers to reenter invalid fields.</li> <li>3. Customer corrects field(s).</li> <li>4. Use Case resumes in step 2.</li> </ol> </li> </ol>		
<b>Includes:</b>	Customer database		
<b>Frequency of Use:</b>	On demand.		
<b>Special Requirements:</b>	The customer is required to have an account with us.		
<b>Assumptions:</b>	The customer needs to make edits to how they want the account to be.		
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. How many preferences should we add?</li> <li>2. How many disabilities can we cover with our services?</li> </ol>		

## Accessibility Information

<b>Use Case ID:</b>	UC-10.5		
<b>Use Case Name:</b>	View Accessibility Information		
<b>Created By:</b>	Everett DeVaux	<b>Last Updated By:</b>	Everett DeVaux
<b>Date Created:</b>	09/08/2023	<b>Last Revision Date:</b>	9/17/2023
<b>Actors:</b>	Person with disability		
<b>Description:</b>	Able to see which vehicles have accessibility features.		
<b>Trigger:</b>	The customer can see which vehicles I can board that accommodate their needs.		
<b>Preconditions:</b>	1. Customer must sign in to save information. 2. Customer has access to driver database for what's available.		
<b>Postconditions:</b>	Customer can make a reservation for the specific vehicle.		
<b>Normal Flow:</b>	1. Customer logs into the account. 2. Customer goes to search for drivers/vehicles that are available. 3. Customer clicks on [Filter]. 4. In the filter options, the customer can choose what preferences they want in the vehicle they will ride in. 5. The Customer accepts preferences. 6. The system navigates the customer to continue to schedule pick-up.		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, does not have an account. 1. The system will prompt customers to make an account or sign in as guests. 2. Customer inserts required information needed for booking with or without an account. 3. Use Case resumes on step 2.  5b. In step 5 of the normal flow if the customer declines preferences. 1. The system will prompt the user to accept their decision. 2. Customer declines 3. The system navigates the customer back to the driver menu.		
<b>Exceptions:</b>	1a. In step 1 of the normal flow, if the customer declines to sign in 1. The customer declines to sign in. 2. The system will navigate customers under guest to preferences. 3. The system will not save preferences for customers. 4. Use Case resumes on step 3 of normal flow.		
<b>Includes:</b>	Customer database Sign-in information Driver database Vehicle database		
<b>Frequency of Use:</b>	On-demand		
<b>Special Requirements:</b>	Needs to be able to update Customer database. Needs to be able to access vehicle database.		
<b>Assumptions:</b>	Customer may or may not have an account. Customer needs to find a driver with specific needs for travel.		
<b>Notes and Issues:</b>			



## Riders – Guardian

### Sign Up Children for School Bus Route

<b>Use Case ID:</b>	14.2		
<b>Use Case Name:</b>	Control Children's School Bus Ride need		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	Steven Sanchez.
<b>Date Created:</b>	09/06/2023	<b>Last Revision Date:</b>	09/11/2023
<b>Actors:</b>	Parent		
<b>Description:</b>	Parent can control the child needs a school bus ride on any given day by setting and on/off toggle (in similar fashion to how you can activate/deactivate the use of a credit card by logging into an app and setting the appropriate state). This will allow the bus driver to know when they need to stop at the child's house or expect them at the bus stop.		
<b>Trigger:</b>	The child will/will not be riding the bus on the given day.		
<b>Preconditions:</b>	(1) The parent and child are registered and associated in the system. (2) The child is assigned to a school bus route.		
<b>Postconditions:</b>	The need to pickup/deliver a child on a given bus route will be known on any given day.		
<b>Normal Flow:</b>	(1) Parent logs into the system (2) Parents selects child from lists. (3) Parent activates/deactivates the child for the bus ride.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Rarely – only when a child is sick (not needing a bus ride) or has alternate transportation.		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	Assumes the parent and child are known to the system and the child has an assigned school bus route.		
<b>Notes and Issues:</b>	None		

## Create Child Profile

<b>Use Case ID:</b>	14.3		
<b>Use Case Name:</b>	Create Child Profile		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	Steven Sanchez
<b>Date Created:</b>	9/10/2023	<b>Last Revision Date:</b>	09/11/2023
<b>Actors:</b>	Parent		
<b>Description:</b>	Enter basic demographic/contact information into the system so the child may be (1) known to the system, (2) assigned to a school bus route, and (3) their information is accessible as needed.		
<b>Trigger:</b>	The start of school and a need to ride the school bus.		
<b>Preconditions:</b>	The parent is known to the system.		
<b>Postconditions:</b>	The child and their information are known to the system (saved).		
<b>Normal Flow:</b>	1. Parent logs into the system. 2. Parent navigates to the application section to enter their children's information. 3. Parent fills in all required fields. 4. Parent saves entered information.		
<b>Alternative Flows:</b>	3a. In step 3 of the normal flow, if a field is missing required information, 1. System highlights missing fields 2. Parents include all information. 3. Use case continues to step 4 of normal flow		
<b>Exceptions:</b>	None		
<b>Includes:</b>	Signup children for school bus route		
<b>Frequency of Use:</b>	Rarely (once per child/year)		
<b>Special Requirements:</b>	Must be a parent.		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## Issue Special Instructions for Child's Bus Ride

<b>Use Case ID:</b>	14.4		
<b>Use Case Name:</b>	Issue Special Instructions for Child's Bus Ride		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	Steven Sanchez
<b>Date Created:</b>	09/10/2023	<b>Last Revision Date:</b>	09/11/23
<b>Actors:</b>	Parent		
<b>Description:</b>	Parent provides/requests special accommodation for the child during a school bus ride (for instance, requesting the child sit at the front of the bus because they are on crutches).		
<b>Trigger:</b>	When school has started —AND— the child develops a need for which special accommodation is required.		
<b>Preconditions:</b>	Parent and child are registered and associated in the system. Child is associated to a known school bus route.		
<b>Postconditions:</b>	The nature and duration of the child special accommodation request is stored and accessible in the system.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Parent logs into the system.</li> <li>2. Parent selects the child associated with both the parent and a known school bus route.</li> <li>3. Parent enters special instructions and a time span (start date/end date) into the form presented.</li> <li>4. Parent saves the information in the system.</li> </ol>		
<b>Alternative Flows:</b>	If the process of storing the special instruction in the system is unsuccessful, the system provides contact info for personnel to assist them further.		
<b>Exceptions:</b>	None		
<b>Includes:</b>	Signup children for school bus route		
<b>Frequency of Use:</b>	Rarely (only as a need arises).		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	None		

## Report of Child's School Bus Ride Use

<b>Use Case ID:</b>	14.5		
<b>Use Case Name:</b>	Report of Child's School Bus Ride Use		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	Steven Sanchez
<b>Date Created:</b>	9/10/2023	<b>Last Revision Date:</b>	09/11/2023
<b>Actors:</b>	Parent		
<b>Description:</b>	Parent runs reports to determine their child's school bus use history.		
<b>Trigger:</b>	Start of school year/term –AND—need to determine the school bus usage of a child associated in the system to the adult		
<b>Preconditions:</b>	Parent and child are known to the system and child is assigned to a known school bus route.		
<b>Postconditions:</b>	Parent obtains a report of their child's school bus use history.		
<b>Normal Flow:</b>	1. Parent logs into the system 2. Parent navigates to the child's record associated with the parent (and subject of the report). 3. Parent indicates the time span (start date/end date) of the desired report. 4. Parent submit the report request to the system. 5. System generates the report and provides it to the parent.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	14.1		
<b>Frequency of Use:</b>	Rarely (only as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	The report is shown before or after finalizing the bus route signup.		
<b>Notes and Issues:</b>	None		

## Sign up for Emergency Notifications

<b>Use Case ID:</b>	14.6		
<b>Use Case Name:</b>	Sign up for Emergency Notifications		
<b>Created By:</b>	Nathan Toothaker	<b>Last Updated By:</b>	Steven Sanchez
<b>Date Created:</b>	9/11/2023	<b>Last Revision Date:</b>	09/11/2023
<b>Actors:</b>	Parent		
<b>Description:</b>	Parent signs up to receive automatic notifications from application		
<b>Trigger:</b>	Start of the school year/term.		
<b>Preconditions:</b>	1. Parent and child are known to the system. 2. Parent has application installed on their phone		
<b>Postconditions:</b>	Parent will receive automatic notifications in case of an emergency		
<b>Normal Flow:</b>	1. Parent logs into the system 2. Parent navigates to the child's record associated with the parent. 3. On the child's records, parent clicks button to sign up for notifications. 4. The app will report back to the parent that they are now signed up for notifications.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	3a. In step 3 of the normal flow, the parent may not have notifications allowed on their device for this app. 1. The app will notify the parent, then request permission through the phone to send notifications.		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Rarely (as needed)		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	Notifications are enabled on the parent's device.		
<b>Notes and Issues:</b>	None		

## View My Child's Current Status Position

<b>Use Case ID:</b>	14.7		
<b>Use Case Name:</b>	View My Child's Current Status/Position		
<b>Created By:</b>	Bill Tobias	<b>Last Updated By:</b>	Steven Sanchez
<b>Date Created:</b>	9/10/2023	<b>Last Revision Date:</b>	09/11/2023
<b>Actors:</b>	Parent		
<b>Description:</b>	Parent uses app to determine current location of bus child is on.		
<b>Trigger:</b>	Start of school year/term –AND—need to determine the location of the bus a child is on.		
<b>Preconditions:</b>	1. Parent and child are known to the system. 2. Child is assigned to a known school bus route. 3. Child has boarded the bus for the route.		
<b>Postconditions:</b>	Parent knows child's location if still on bus or if they have already disembarked.		
<b>Normal Flow:</b>	1. Parent logs into the system 2. Parent navigates to the child's record associated with the parent. 3. On the child's records, parent clicks button to "Locate" child. 4. The app will report back to the parent one of the following states 1. The scheduled bus route has not started. 2. The scheduled bus route has ended. 3. The scheduled bus route is in progress, but the child has exited the bus [for whatever reason, such as their bus stop has been reached already]. 4. The scheduled bus route is in progress and based on past historical average progress speed, the bus is approximately (n) miles into its route. 5. Child did not board the bus for the route.		
<b>Alternative Flows:</b>	None		
<b>Exceptions:</b>	None		
<b>Includes:</b>	None		
<b>Frequency of Use:</b>	Rarely (as needed)		
<b>Special Requirements:</b>	GPS tracking on buses.		
<b>Assumptions:</b>	None		
<b>Notes and Issues:</b>	This is a more practical than a solution requiring real time GPS point.		

## Outside Contacts

### Add Special Service Order

<b>Use Case ID:</b>	16.1		
<b>Use Case Name:</b>	Add Special Service Order		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Mechanic		
<b>Description:</b>	Add a special service order, such as body work or major repair that our in-house team can not complete.		
<b>Trigger:</b>	A mechanic has decided service order was deemed out of scope of our capabilities		
<b>Preconditions:</b>	A normal service order will exist		
<b>Postconditions:</b>	A special service order entry will be created in the database		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view service orders”</li> <li>2. A list of service orders appears, with various filters available</li> <li>3. Operator clicks on “view detail” on a service order</li> <li>4. Operator clicks on “flag as special service order”</li> <li>5. Operator adds any needed details</li> <li>6. Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if user fails to provide all necessary details of the service record.</p> <ol style="list-style-type: none"> <li>1. System highlights all missing fields that need data entered.</li> <li>2. User provides data for all missing fields</li> <li>3. Use case continues to step 6 of the normal flow.</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. System will state a no order found message and encourage users to adjust filters.</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>1. A prompt appears to confirm this decision</li> </ol>		
<b>Includes:</b>	Work Order, Vehicle Search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Update Special Service Order

<b>Use Case ID:</b>	16.2		
<b>Use Case Name:</b>	Update Special Service order		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative		
<b>Description:</b>	A special service order needs to be updated		
<b>Trigger:</b>	A mechanic has realized that a special service order needs an update		
<b>Preconditions:</b>	A special service order exists		
<b>Postconditions:</b>	The special service order will be updated to be accurate		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view Special service orders”</li> <li>2. A list of Special service orders appears, with various filters available</li> <li>3. Operator clicks “view detail” on the service order they are interested in.</li> <li>4. Operator clicks on “edit detail” on a service order</li> <li>5. Operator updates any needed fields.</li> <li>6. Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if the user fails to provide all necessary details of the service record.</p> <ol style="list-style-type: none"> <li>1. System highlights all missing fields that need data entered.</li> <li>2. User provides data for all missing fields</li> <li>3. Use case continues to step 6 of the normal flow.</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. System will state a no order found message and encourage users to adjust filters.</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>1. A prompt appears to confirm this decision</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Service Order, Vehicle Search, view all special service orders, view special service order detail		
<b>Frequency of Use:</b>	Rarely		

<b>Special Requirements:</b>	
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

## View Special Service Order Detail

<b>Use Case ID:</b>	16.3		
<b>Use Case Name:</b>	View Special Service order detail		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Mechanic, outside contact person		
<b>Description:</b>	A special service order needs to be viewed		
<b>Trigger:</b>	A mechanic or outside contact person needs to view a special service order		
<b>Preconditions:</b>	A special service order exists		
<b>Postconditions:</b>	No data will be changed, a special service order will have been viewed		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view Special service orders”</li> <li>2. A list of Special service orders appears, with various filters available</li> <li>3. Operator clicks “view detail” on the service order they are interested in.</li> <li>4. Operator closes window when they are finished</li> </ol>		
<b>Alternative Flows:</b>	<p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. System will state a no order found message and encourage users to adjust filters.</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Service Order, view special service orders		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## View All Special Service Order

<b>Use Case ID:</b>	16.4		
<b>Use Case Name:</b>	View all special service orders		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	A special service order needs to be viewed		
<b>Trigger:</b>	A mechanic or outside contact person needs to view a special service order		
<b>Preconditions:</b>	none		
<b>Postconditions:</b>	A list of all special service orders is displayed		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view Special service orders”</li> <li>2. A list of Special service orders appears, with various filters available</li> <li>3. Operator closes window when they are finished</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Service Orders		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Mark Special Service Order Inactive

<b>Use Case ID:</b>	16.5		
<b>Use Case Name:</b>	Mark Special Service Order Inactive		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative		
<b>Description:</b>	A special service order needs to be updated		
<b>Trigger:</b>	A mechanic has realized that a special service order needs an update		
<b>Preconditions:</b>	A special service order exists		
<b>Postconditions:</b>	The special service order will be updated to be inactive		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view Special service orders”</li> <li>2. A list of Special service orders appears, with various filters available</li> <li>3. Operator clicks “view detail” on the service order they are interested in.</li> <li>4. Operator clicks on “mark inactive” on a service order</li> <li>5. Operator updates any needed fields.</li> <li>6. Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if the user fails to provide all necessary details of the service record.</p> <ol style="list-style-type: none"> <li>1. System highlights all missing fields that need data entered.</li> <li>2. User provides data for all missing fields</li> <li>3. Use case continues to step 6 of the normal flow.</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. The system will state a no order found message, and encourage users to adjust filters.</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>1. A prompt appears to confirm this decision</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Service Order, Vehicle Search, view all special service orders, view special service order detail		
<b>Frequency of Use:</b>	Rarely		

<b>Special Requirements:</b>	
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

## Mark Special Service Order Active

<b>Use Case ID:</b>	16.6		
<b>Use Case Name:</b>	Mark Special Service Order Active		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative		
<b>Description:</b>	A special service order needs to be updated		
<b>Trigger:</b>	A mechanic has realized that a special service order needs an update		
<b>Preconditions:</b>	A special service order exists and is marked inactive		
<b>Postconditions:</b>	The special service order marked as active		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view Special service orders”</li> <li>2. A list of Special service orders appears, with various filters available. They set the setting to “view inactive”</li> <li>3. Operator clicks “view detail” on the service order they are interested in.</li> <li>4. Operator clicks on “mark as active” on a service order</li> <li>5. Operator updates any needed fields.</li> <li>6. Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if the user fails to provide all necessary details of the service record.</p> <ol style="list-style-type: none"> <li>1. System highlights all missing fields that need data entered.</li> <li>2. User provides data for all missing fields</li> <li>3. Use case continues to step 6 of the normal flow.</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. The system will state a no order found message, and encourage users to adjust filters.</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>1. A prompt appears to confirm this decision</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Service Order, Vehicle Search, view all special service orders, view special service order detail		
<b>Frequency of Use:</b>	Rarely		

<b>Special Requirements:</b>	
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

## Add Bid to Special Service Order

<b>Use Case ID:</b>	16.5		
<b>Use Case Name:</b>	Add Bid to special service order		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	Add a bid to a special service order record		
<b>Trigger:</b>	A bid is received		
<b>Preconditions:</b>	A special service order exists		
<b>Postconditions:</b>	A bid record will be attached to the special service order		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Operator clicks on “view Special service orders”</li> <li>A list of Special service orders appears, with various filters available</li> <li>Operator clicks “view detail” on the service order they are interested in.</li> <li>Operator clicks “add bid”</li> <li>Operator adds any needed details</li> <li>Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if user fails to provide all necessary details of the bid record.</p> <ol style="list-style-type: none"> <li>System highlights all missing fields that need data entered.</li> <li>User provides data for all missing fields</li> <li>Use case continues to step 6 of the normal flow.</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>System will state a no order found message, and encourage users to adjust filters.</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>System will state that the service is not available, and to check back later</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>A prompt appears to confirm this decision</li> </ol>		
<b>Includes:</b>	Work Order, Vehicle Search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Update Bid

<b>Use Case ID:</b>	16.6		
<b>Use Case Name:</b>	Update Bid		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	A special service order needs to be updated		
<b>Trigger:</b>	A outside contact person has realized that a bid needs an update		
<b>Preconditions:</b>	A bid exists		
<b>Postconditions:</b>	The bid will be updated to be accurate		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view service orders”</li> <li>2. A list of bids appears, with various filters available</li> <li>3. Operator clicks “view detail” on the bid are interested in.</li> <li>4. Operator clicks “edit bid”</li> <li>5. Operator updates any needed fields.</li> <li>6. Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if the user fails to provide all necessary details of the service record.</p> <p>1. System highlights all missing fields that need data entered.</p> <p>2. User provides data for all missing fields</p> <p>3. Use case continues to step 6 of the normal flow</p> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <p>1. The system will state a no order found message, and encourage users to adjust filters.</p> <p>6a) the user decides not to save</p> <p>1. A prompt appears to confirm this decision</p>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <p>1. System will state that the service is not available, and to check back later</p>		
<b>Includes:</b>	Bid search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

### View Bid Detail

<b>Use Case ID:</b>	16.7		
<b>Use Case Name:</b>	View Bid detail		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative		
<b>Description:</b>	A outside contact representative may view details of a bid order		
<b>Trigger:</b>			
<b>Preconditions:</b>	A bid exists		
<b>Postconditions:</b>	The details of the bid will be displayed on screen		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view service orders”</li> <li>2. A list of bids appears, with various filters available</li> <li>3. Operator clicks “view detail” on the bid they are interested in.</li> <li>4. Operator closes window when they are done</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Bid search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## View All Bids

<b>Use Case ID:</b>	16.8		
<b>Use Case Name:</b>	View all bids		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	View all bids we have received for all service orders		
<b>Trigger:</b>	An operator wants to know how many bids we have outstanding		
<b>Preconditions:</b>	none		
<b>Postconditions:</b>	A list of bids will be displayed, no data will be edited.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view service orders”</li> <li>2. A list of bids appears, with various filters available</li> <li>3. Operator closes window when they are done</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	<p>2a. In step 2 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Bid search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Add Appraisal to Special Service Order

<b>Use Case ID:</b>	16.9		
<b>Use Case Name:</b>	Add appraisal to special service order		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	Add an appraisal to a special service order record		
<b>Trigger:</b>	An appraisal is received		
<b>Preconditions:</b>	A special service order exists		
<b>Postconditions:</b>	An appraisal record will be attached to the special service order		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Operator clicks on “view Special service orders”</li> <li>A list of Special service orders appears, with various filters available</li> <li>Operator clicks “view detail” on the service order they are interested in.</li> <li>Operator clicks “add appraisal”</li> <li>Operator adds any needed details</li> <li>Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if user fails to provide all necessary details of the appraisal record.</p> <ol style="list-style-type: none"> <li>System highlights all missing fields that need data entered.</li> <li>User provides data for all missing fields</li> <li>Use case continues to step 6 of the normal flow.</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>The system will state a no order found message, and encourage users to adjust filters.</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>System will state that the service is not available, and to check back later</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>A prompt appears to confirm this decision</li> </ol>		
<b>Includes:</b>	Work Order, Vehicle Search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Update Appraisal

<b>Use Case ID:</b>	16.10		
<b>Use Case Name:</b>	Update appraisal		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	A special service order appraisal needs to be updated		
<b>Trigger:</b>	A outside contact person has realized that an appraisal needs an update		
<b>Preconditions:</b>	An appraisal exists		
<b>Postconditions:</b>	The appraisal will be updated to be accurate		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view appraisals”</li> <li>2. A list of appraisals appears, with various filters available</li> <li>3. Operator clicks “view detail” on the appraisal are interested in.</li> <li>4. Operator clicks “edit appraisal”</li> <li>5. Operator updates any needed fields.</li> <li>6. Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if the user fails to provide all necessary details of the service record.</p> <ol style="list-style-type: none"> <li>1. System highlights all missing fields that need data entered.</li> <li>2. User provides data for all missing fields</li> <li>3. Use case continues to step 6 of the normal flow</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. The system will state a no order found message and encourage users to adjust filters.</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>1. A prompt appears to confirm this decision</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	appraisal search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			

<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

## View Appraisal Detail

<b>Use Case ID:</b>	16.11		
<b>Use Case Name:</b>	View appraisal detail		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative		
<b>Description:</b>	A outside contact representative may view details of an appraisal		
<b>Trigger:</b>			
<b>Preconditions:</b>	An appraisal exists		
<b>Postconditions:</b>	The details of the appraisal will be displayed on screen		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view service orders”</li> <li>2. A list of appraisals appears, with various filters available</li> <li>3. Operator clicks “view detail” on the appraisal they are interested in.</li> <li>4. Operator closes window when they are done</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	appraisal search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## View All Appraisals

<b>Use Case ID:</b>	16.12		
<b>Use Case Name:</b>	View all appraisals		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	View all appraisals we have one special service orders		
<b>Trigger:</b>	An operator wants to view a list of our internal appraisals		
<b>Preconditions:</b>	none		
<b>Postconditions:</b>	A list of appraisals will be displayed, no data will be edited.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view service orders”</li> <li>2. A list of appraisals appears, with various filters available</li> <li>3. Operator closes window when they are done</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	<p>2a. In step 2 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	appraisal search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Add Outside Work Order to Special Service Order

<b>Use Case ID:</b>	16.13		
<b>Use Case Name:</b>	Add outside work order to special service order		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	Add an outside service order to a special service order record		
<b>Trigger:</b>	A bid has been accepted		
<b>Preconditions:</b>	A special service order exists		
<b>Postconditions:</b>	An outside work order record will be attached to the special service order		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view Special service orders”</li> <li>2. A list of Special service orders appears, with various filters available</li> <li>3. Operator clicks “view detail” on the service order they are interested in.</li> <li>4. Operator clicks “add outside work order”</li> <li>5. Operator adds any needed details</li> <li>6. Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if user fails to provide all necessary details of the work order record.</p> <ol style="list-style-type: none"> <li>1. System highlights all missing fields that need data entered.</li> <li>2. User provides data for all missing fields</li> <li>3. Use case continues to step 6 of the normal flow.</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. System will state a no order found message and encourage users to adjust filters.</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>1. A prompt appears to confirm this decision</li> </ol>		
<b>Includes:</b>	Work Order, Vehicle Search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Update Outside Work Order

<b>Use Case ID:</b>	16.14		
<b>Use Case Name:</b>	Update outside work order		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	An outside work order needs to be updated		
<b>Trigger:</b>	A outside contact person has realized that an outside work order needs an update		
<b>Preconditions:</b>	An outside work order exists		
<b>Postconditions:</b>	The outside work order will be updated to be accurate		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view outside work orders”</li> <li>2. A list of outside work orders appears, with various filters available</li> <li>3. Operator clicks “view detail” on the outside work order are interested in.</li> <li>4. Operator clicks “edit outside work order”</li> <li>5. Operator updates any needed fields.</li> <li>6. Operator clicks save</li> </ol>		
<b>Alternative Flows:</b>	<p>5a. In step 5 of the normal flow if the user fails to provide all necessary details of the service record.</p> <ol style="list-style-type: none"> <li>1. System highlights all missing fields that need data entered.</li> <li>2. User provides data for all missing fields</li> <li>3. Use case continues to step 6 of the normal flow</li> </ol> <p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. The system will state a no order found message and encourage users to adjust filters.</li> </ol> <p>6a) the user decides not to save</p> <ol style="list-style-type: none"> <li>1. A prompt appears to confirm this decision</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	work order search		
<b>Frequency of Use:</b>	Rarely		

## View Outside Work Order

<b>Use Case ID:</b>	16.15		
<b>Use Case Name:</b>	View outside work order		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	View an outside work order		
<b>Trigger:</b>	A outside contact person wants to view details of an outside work order		
<b>Preconditions:</b>	An outside work order exists		
<b>Postconditions:</b>	The outside work order will be displayed on screen, no data will be changed.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view outside work orders”</li> <li>2. A list of outside work orders appears, with various filters available</li> <li>3. Operator clicks “view detail” on the outside work order are interested in.</li> <li>4. Operator closes the window when complete.</li> </ol>		
<b>Alternative Flows:</b>	<p>2a. In step 2 of the normal flow, if no orders match the filters</p> <ol style="list-style-type: none"> <li>1. The system will state a no order found message and encourage users to adjust filters.</li> </ol>		
<b>Exceptions:</b>	<p>2a. In step 2 or 3 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Outside work order search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## View All Outside Work Orders

<b>Use Case ID:</b>	16.16		
<b>Use Case Name:</b>	View all outside work orders		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	View all work order		
<b>Trigger:</b>	An operator wants to view a list of our external work orders		
<b>Preconditions:</b>	A bid has been accepted.		
<b>Postconditions:</b>	A list of work orders will be displayed, no data will be edited.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Operator clicks on “view work orders”</li> <li>A list of work order appears, with various filters available</li> <li>Operator closes window when they are done</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	<p>2a. In step 2 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Work order search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

## Approve Bid

<b>Use Case ID:</b>	16.17		
<b>Use Case Name:</b>	Approve Bid		
<b>Created By:</b>	Jonathan Beck	<b>Last Updated By:</b>	James Williams
<b>Date Created:</b>	9/30/2023	<b>Last Revision Date:</b>	12/10/2023
<b>Actors:</b>	Outside Contact Representative, Mechanic		
<b>Description:</b>	Approve a bid we have received to authorize work		
<b>Trigger:</b>	A bid has been accepted by our management team		
<b>Preconditions:</b>	A bid exists and it has been approved to be worked on		
<b>Postconditions:</b>	A bid will be marked as “accepted”		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Operator clicks on “view special service orders”</li> <li>2. A list of special service orders appears, with various filters available</li> <li>3. Operator clicks “view bids”</li> <li>4. Operator clicks “accept” on the bid they want to accept.</li> <li>5. Operator closes window when they are done</li> </ol>		
<b>Alternative Flows:</b>			
<b>Exceptions:</b>	<p>2a. In step 2 of the normal flow, the database is unable to be reached</p> <ol style="list-style-type: none"> <li>1. System will state that the service is not available, and to check back later</li> </ol>		
<b>Includes:</b>	Bid search		
<b>Frequency of Use:</b>	Rarely		
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

# Stakeholders: User Interfaces

To interact with the application, the application must use a graphical interface which displays data to the user and provides input widgets and fields through which the user can pass data to and invoke functionality from the application. The User Interfaces are wireframes which give a basic picture of the graphical interfaces which the application will be provided to satisfy a certain Use Case. Each Use Case will have one or more User Interfaces attached.

Ui start

## UI Drawings for Admin

### Create Profile Desktop View(1)

The wireframe shows a desktop application window titled "Create Profile". The window has a header bar with tabs and a logo. On the right side, there is a circular profile icon and an "Account Name" field. The main content area contains fields for personal information: Given Name, Family Name, DOB, Email, Postal Code, City, Region, Address, and Phone Number, each with its own text box. A "Create" button is located at the bottom right of the form. The footer contains a copyright notice: "© 2023 Name".

Field	Type
Given Name	Text box
Family Name	Text box
DOB	Text box
Email	Text box
Postal Code	Text box
City	Text box
Region	Text box
Address	Text box
Phone Number	Text box

## Create Profile Desktop View

The screenshot shows a desktop application window titled "Create Profile". The window has a blue header bar with standard window controls (minimize, maximize, close) and a "Window" tab. In the top right corner, there is a blue circular button labeled "Profil" and a text field labeled "Account Name". The main content area is a dark grey box containing the following fields:

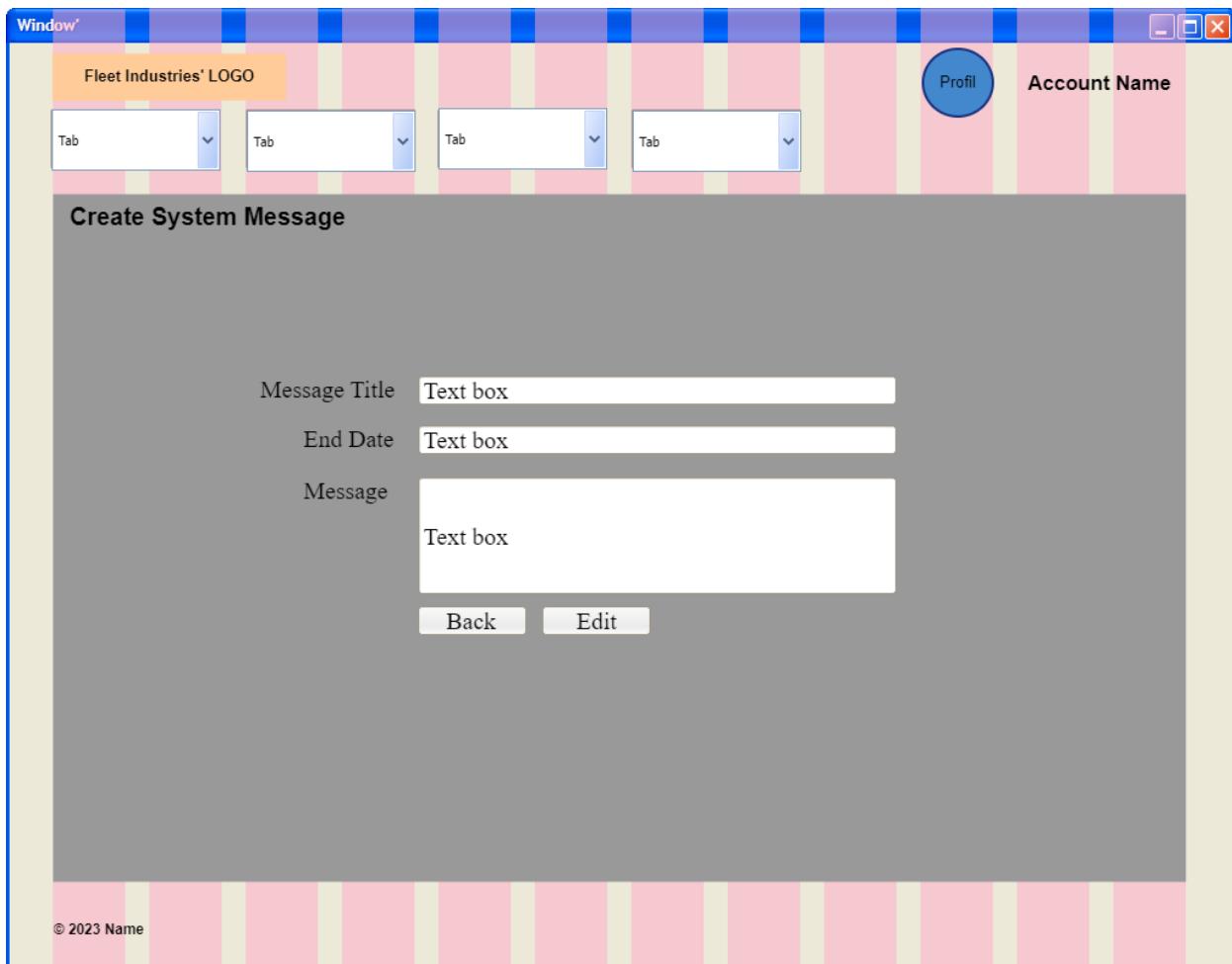
Given Name	Text box
Family Name	Text box
DOB	Text box
Email	Text box
Zip	Text box
Address	Text box
City	Text box
Phone Number	Text box
Roles	Combo Box

At the bottom right of the form is a "Create" button. The footer of the window contains the text "© 2023 Name".

## System Message - Create - Desktop View

The screenshot shows a desktop application window titled "Create System Message". The window has a blue header bar with standard window controls (minimize, maximize, close) on the right. In the top-left corner of the header, there is a placeholder text "Fleet Industries' LOGO". To the right of the logo is a blue circular button labeled "Profil". Next to the button is the text "Account Name". Below the header, there are four tabs labeled "Tab" with dropdown arrows. The main content area is a dark grey box containing three input fields: "Message Title" (text box), "End Date" (text box), and "Message" (text box). Below the "Message" text box is a "Create" button. At the bottom left of the main content area, there is a copyright notice: "© 2023 Name".

## System Message - Edit - Desktop View



## System Message - Edit\_Deactivate - Destkop View

The screenshot shows a desktop application window titled "System Message - Edit\_Deactivate - Destkop View". The window has a standard OS X-style title bar with "Window" and control buttons. In the top-left corner, there is a logo for "Fleet Industries' LOGO". To the right of the logo is a "Profil" button. The main content area is titled "Create System Message". It contains three text input fields: "Message Title" and "End Date" (both labeled "Text box"), and a larger "Message" area with its own "Text box". Below these fields are three buttons: "Back", "Edit", and "Deactivate". The bottom of the window features a decorative footer bar with alternating pink and yellow segments. A copyright notice "© 2023 Name" is visible at the bottom left.

## System Message - List - Desktop View

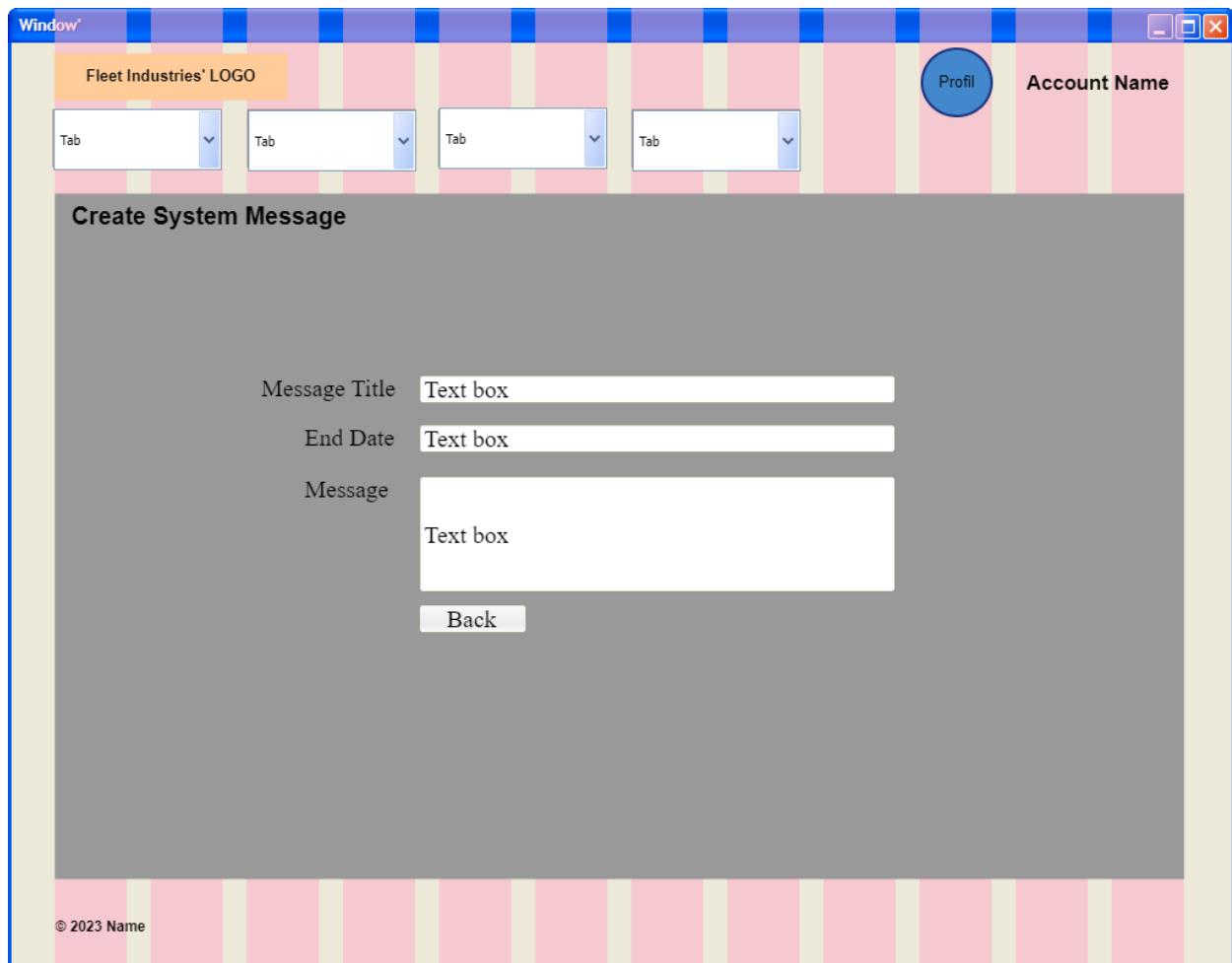
The screenshot shows a desktop application window titled "System Message - List - Desktop View". The window has a standard title bar with icons for minimize, maximize, and close. Inside, there's a header section with a logo labeled "Fleet Industries' LOGO", a "Profil" button, and an "Account Name" field. Below this is a tab bar with four tabs, each with a dropdown arrow. A "Name of Tab" label is positioned above the main content area.

The main content area features a heading "Heading level 2" and a search/filter section with a "Search" input, a magnifying glass icon, and a "Filter" dropdown. Below this is a table with the following data:

Message Title	Start Date	End Date	
Cell Content 1	Content	Content	<a href="#">Edit</a>
Cell content 2	Content	Content	<a href="#">Edit</a>
Cell content 3	Content	Content	<a href="#">Edit</a>
Cell content 4	Content	Content	<a href="#">Edit</a>
Cell content 5	Content	Content	<a href="#">Edit</a>
Cell content 6	Content	Content	<a href="#">Edit</a>
Cell content 7	Content	Content	<a href="#">Edit</a>

A "Back" button is located at the bottom left of the content area. The footer of the window contains the text "© 2023 Name".

## System Message - View - Desktop View



## View Employee Profile Desktop View

The screenshot shows a desktop application window titled "View Employee Profile". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a "Window" tab. In the top right corner, there is a blue circular button labeled "Profil" and the text "Account Name". The main content area is titled "Employee ID:" and contains six input fields:

Field	Type
Name	Input
Email	Input
Address	Input
DOB	Input
ZipCode	Input
Roles	Combo Box

Below the input fields is a "Back" button. On the right side of the main content area, there is a large empty rectangular box with a diagonal cross through it, containing the text "275 x 261". At the bottom left of the window, there is a copyright notice: "© 2023 Name".

## View Employee Profile With Edit Desktop View

The screenshot shows a desktop application window titled "View Employee Profile". The window has a blue header bar with tabs labeled "Tab", "Tab", "Tab", and "Tab". On the right side of the header, there is a blue circular button labeled "Profil" and the text "Account Name". The main content area contains a form for editing an employee profile. The form is titled "Employee ID:" and includes fields for Name (Input), Email (Input), Address (Input), DOB (Input), ZipCode (Input), and Roles (Combo Box). There is also a large empty rectangular area with a diagonal cross and the dimensions "275 x 261". A "Back" button is located at the bottom left of the form, and an "Edit" button is located on the right side. The footer of the window contains the text "© 2023 Name".

## View Employee Profile With Edit\_Deactivate Desktop View

The screenshot shows a desktop application window titled "View Employee Profile". The window has a blue header bar with tabs and a "Profile" button. The main content area contains fields for "Employee ID:" with labels and input boxes for Name, Email, Address, DOB, ZipCode, and Roles (a dropdown menu). There are two buttons at the bottom right: "Deactivate" and "Edit". A watermark "275 x 261" is visible in the background of the form area. The footer of the window includes a "Back" button and the text "© 2023 Name".

Employee ID:

Name

Email

Address

DOB

ZipCode

Roles

Deactivate

Back

© 2023 Name

## View Employee Profile With Reactivate Destkop View

Window

Fleet Industries' LOGO

Tab Tab Tab Tab Profil Account Name

**View Employee Profile**

**Employee ID:**

Name	<input type="text"/>
Email	<input type="text"/>
Address	<input type="text"/>
DOB	<input type="text"/>
ZipCode	<input type="text"/>
Roles	<input type="text"/> <input type="button" value="▼"/>

275 x 261

© 2023 Name

275 x 261

## View Employee Profile With Roles Desktop View

The screenshot shows a desktop application window titled "View Employee Profile". The window has a blue header bar with tabs labeled "Tab", "Tab", "Tab", "Tab", and "Tab". On the right side of the header is a circular button labeled "Profil" and an "Account Name" field. Below the header is a toolbar with icons for "Edit" and "Print". The main content area contains a form for viewing an employee profile. The form includes fields for "Name" (Input), "Email" (Input), "Address" (Input), "DOB" (Input), "ZipCode" (Input), and "Roles" (Combo Box). A "Back" button is located next to the Roles dropdown. A tooltip for the Roles dropdown lists four items: "Role1" followed by four checkboxes. To the right of the form is a large empty rectangular area with a "275 x 261" dimension indicator. At the bottom left of the window is a copyright notice: "© 2023 Name".

## View List of Users Desktop View(1)

The screenshot shows a desktop application window titled "View List of Users". The window has a blue header bar with standard window controls (minimize, maximize, close) on the right. In the top left corner, there is a placeholder text "Fleet Industries' LOGO". To the right of the logo are four tabs labeled "Tab", each accompanied by a small downward arrow icon. Further to the right is a blue circular button labeled "Profil". On the far right of the header, the text "Account Name" is displayed. The main content area has a dark grey header with the title "View List of Users" and a subtitle "Heading level 2". Below this is a search bar containing a text input field "Text box" with a magnifying glass icon, and a "Filter" dropdown menu labeled "Combo Box". The main body of the window contains a table with the following columns: "First Name", "Last Name", "Email", "Phone Number", and an empty column. The table has 7 rows, each labeled "Cell content 1" through "Cell content 7", with each row containing placeholder text "Content" in all columns except the last one, which contains the word "Details". At the bottom of the content area are two buttons: "Back" and "Add Item". The footer of the window has a light grey background with the copyright notice "© 2023 Name".

First Name	Last Name	Email	Phone Number	
Cell content 1	Content	Content	Content	Details
Cell content 2	Content	Content	Content	Details
Cell content 3	Content	Content	Content	Details
Cell content 4	Content	Content	Content	Details
Cell content 5	Content	Content	Content	Details
Cell content 6	Content	Content	Content	Details
Cell content 7	Content	Content	Content	Details

## View List of Users Desktop View

The screenshot shows a desktop application window titled "View List of Users". The window has a blue header bar with standard window controls (minimize, maximize, close) and a "Window" tab. In the top right corner, there is a blue circular button labeled "Profil" and an "Account Name" field. The main content area has a dark grey header with "View List of Users" and "Heading level 2". Below this is a search bar with a text box containing "Search" and a magnifying glass icon, and a filter dropdown labeled "Filter" with "Combo Box" selected. The main body contains a table with columns: First Name, Last Name, Email, and Phone Number. The rows are labeled "Cell Content 1" through "Cell content 7", each containing placeholder text "Content". At the bottom of the table are two buttons: "Back" and "Add Item". The footer of the window contains the copyright notice "© 2023 Name".

First Name	Last Name	Email	Phone Number	
Cell Content 1	Content	Content	Content	Details
Cell content 2	Content	Content	Content	Details
Cell content 3	Content	Content	Content	Details
Cell content 4	Content	Content	Content	Details
Cell content 5	Content	Content	Content	Details
Cell content 6	Content	Content	Content	Details
Cell content 7	Content	Content	Content	Details

## View Profile Desktop View

The screenshot shows a desktop application window titled "View Profile". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) on the right. In the top left corner of the header, there is a placeholder text "Fleet Industries' LOGO". To the right of the logo, there is a blue circular button labeled "Profil" and the text "Account Name". Below the header, there is a toolbar with four tabs, each labeled "Tab" and featuring a dropdown arrow icon. The main content area is titled "View Profile" and contains a heading "Heading level 2". Below the heading are five input fields, each with a label and an input box:

Label	Input Type
Name	Input
Email	Input
Address	Input
DOB	Input
ZipCode	Input

On the right side of the main content area, there is a large empty rectangular box with a diagonal cross through it, indicating it is a placeholder or a redacted area. The text "275 x 261" is visible near the bottom center of this box. At the bottom of the main content area, there is a "Back" button. The footer of the window contains the copyright notice "© 2023 Name".

## View Profile With Edit Button Destkop View

Window

Fleet Industries' LOGO

Tab Tab Tab Tab Profil Account Name

### View Profile

#### Heading level 2

Name	<input type="text" value="Input"/>
Email	<input type="text" value="Input"/>
Address	<input type="text" value="Input"/>
DOB	<input type="text" value="Input"/>
ZipCode	<input type="text" value="Input"/>

275 x 261

Edit

Back

© 2023 Name

## View Profile With Edit\_Deactivate Desktop View

Window' Minimize Maximize Close

Fleet Industries' LOGO Profil Account Name

Tab Tab Tab Tab

**View Profile**

**Heading level 2**

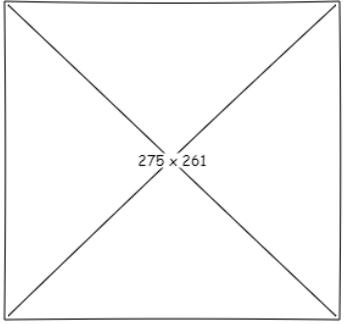
Name

Email

Address

DOB

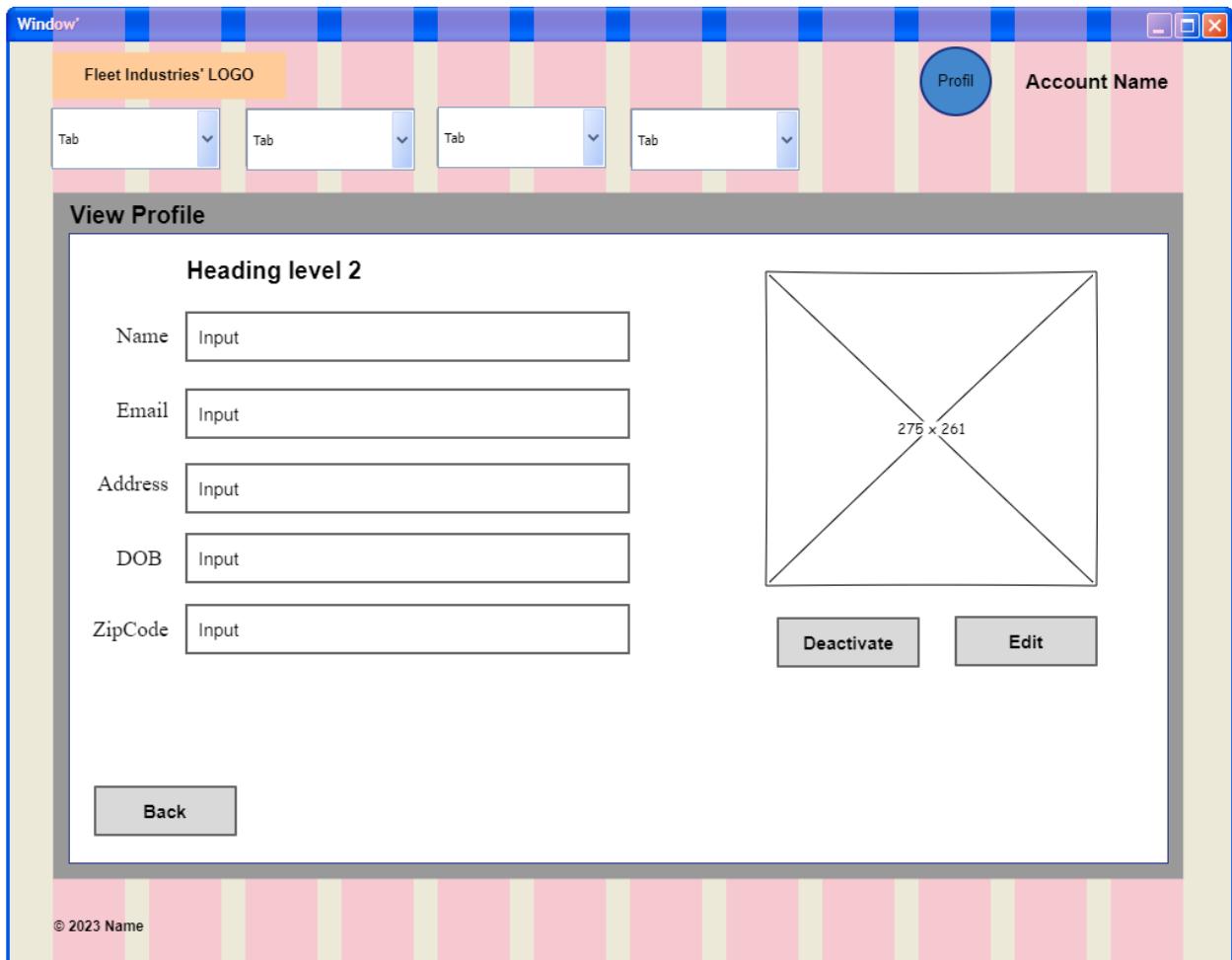
ZipCode



Deactivate Edit

Back

© 2023 Name

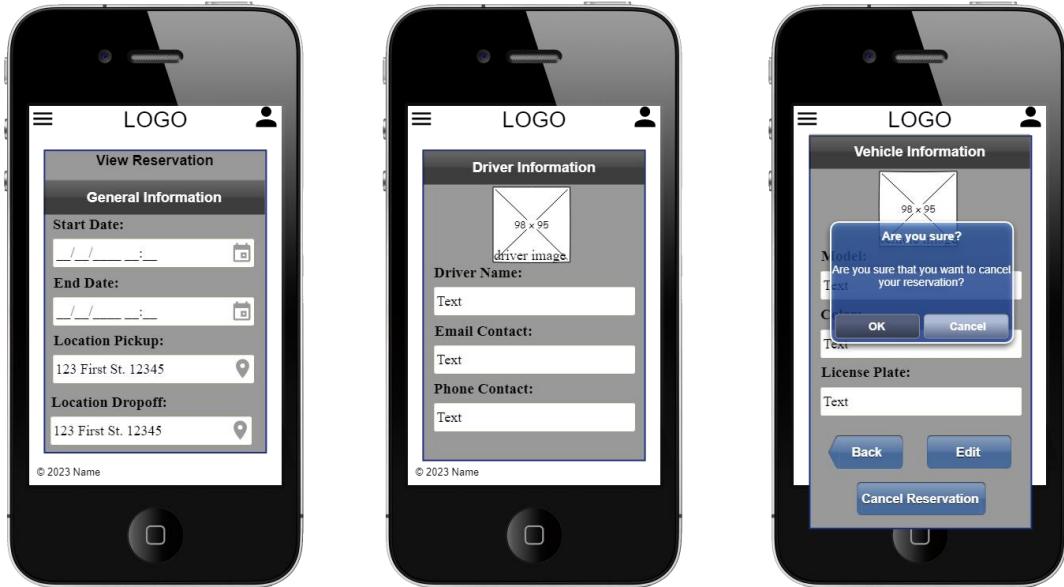


## View Profile With Reactivate Desktop View

The screenshot shows a desktop application window titled "View Profile". The window has a blue header bar with the title and several tabs. In the top right corner, there is a blue circular button labeled "Profil" and the text "Account Name". Below the header, there is a large rectangular input area with a gray border. Inside this area, there is a heading "Heading level 2" followed by five input fields: "Name", "Email", "Address", "DOB", and "ZipCode", each with an associated input field. To the right of these fields is a large empty rectangular box with a diagonal cross through it, labeled "275 x 261". At the bottom right of the input area are two buttons: "Reactivate" and "Edit". At the very bottom left of the window is a small button labeled "Back". The footer of the window contains the text "© 2023 Name".

# UI Drawings for Charter

## Cancel Reservation Mobile



## Cancel Reservation Web

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

View Reservation

General Information      Driver Information

Start Date: 2023/09/25 01:23

End Date:

Location Pickup: ?

Location Dropoff:

Are you sure you want to cancel your reservation?

Cancel      OK

Color: Text

License Plate: Text

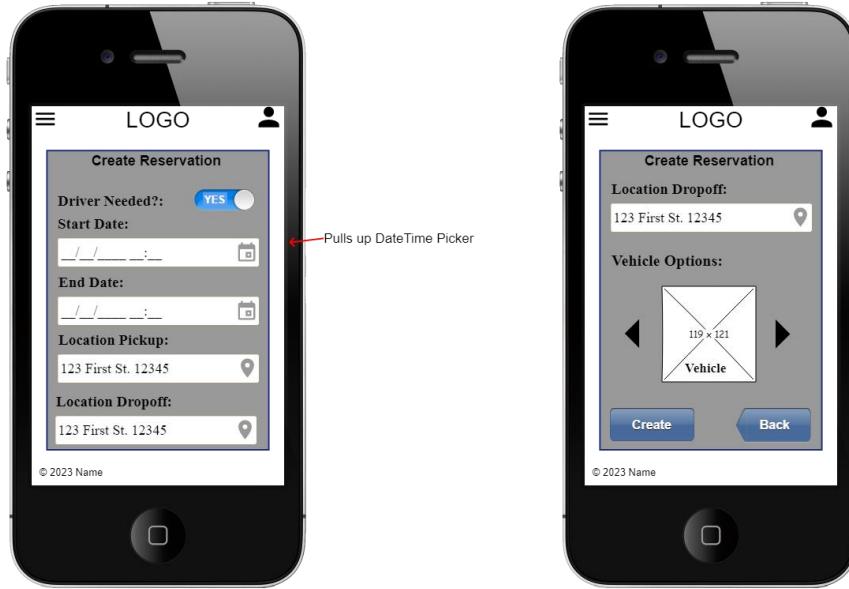
Back      Edit      Cancel Reservation

driver image  
100 x 135

vehicle image  
100 x 135

© 2023 Name

## Create Reservation mobile



## Create Reservation Web

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab Tab Tab Tab

### Create Reservation

**Driver Needed:**  Yes  No

**Requester Riding:**  Yes  No

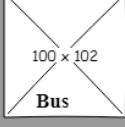
**Rider Quantity:** 1

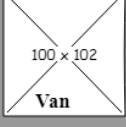
**Accommodations Needed:**  Yes  No

**Accommodation Type:** Name of accommodation

**Accommodation Description:** Enter description here...

**Preferred Vehicle Type:**

 100 x 102  
Bus

 100 x 102  
Van

 100 x 102  
Sedan

**Start Date:** \_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_ 

**End Date:** \_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_ 

**Location Pickup:** 123 First St. 12345 

**Location Dropoff:** 123 First St. 12345 

**Stop 1:** 123 First St. 12345 

**Duration:** 1 hour

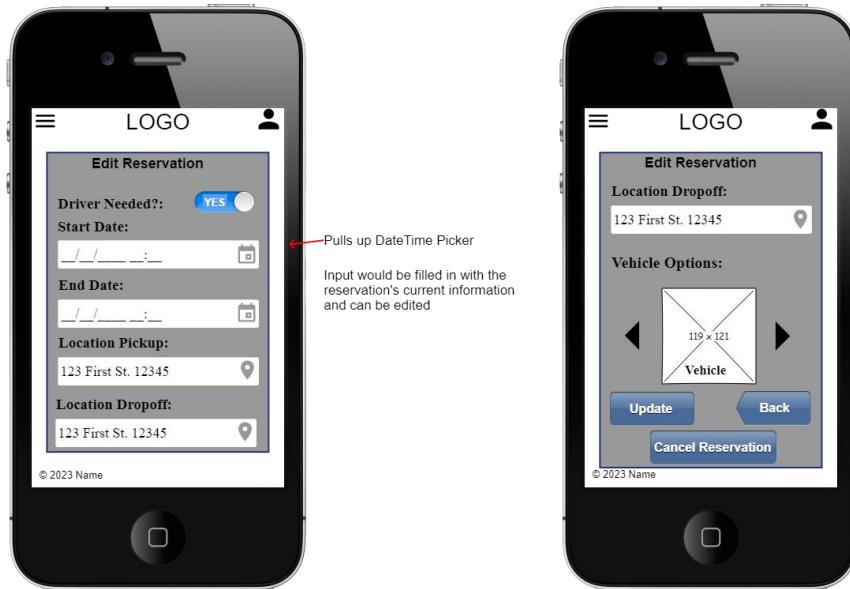
**Description:** Enter description here...

**Add Intermediate Stop** 

**Create** **Cancel**

© 2023 Name

## Update Reservation Mobile



## Update Reservation Web

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab Tab Tab Tab

### Edit Reservation

Driver Needed:  Yes  No

Requester Riding:  Yes  No

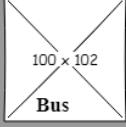
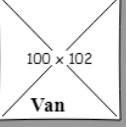
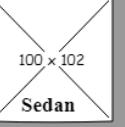
Rider Quantity:

Accommodations Needed:  Yes  No

Accommodation Type:

Accommodation Description:

Preferred Vehicle Type:

 Bus  
 Van  
 Sedan

Start Date:  

End Date:  

Location Pickup:  

Location Dropoff:  

Stop 1:  

Duration:

Description:

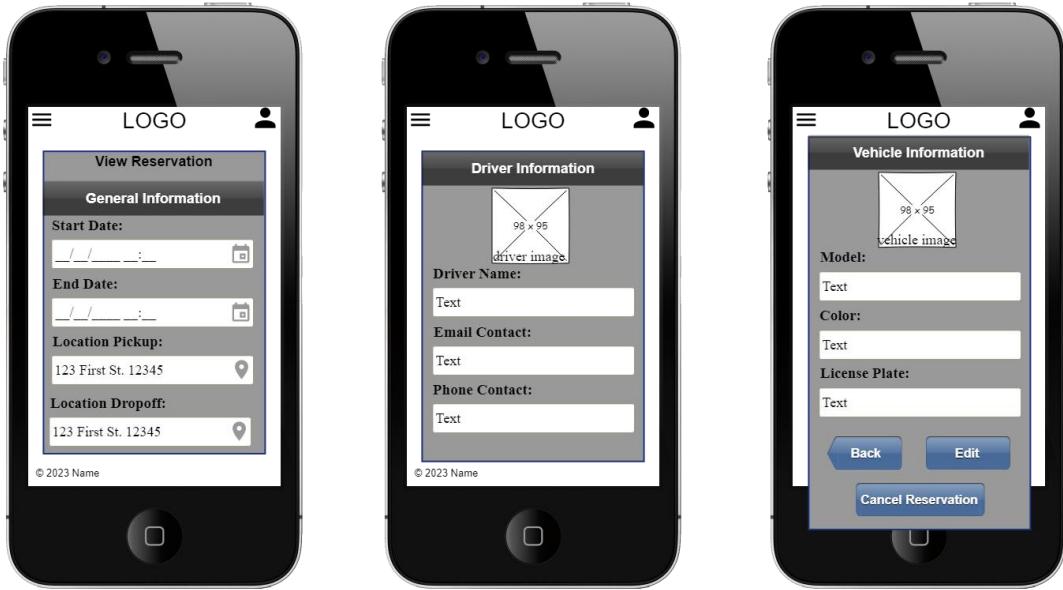
Add Intermediate Stop 

**Back**   **Update**   **Cancel Reservation**

All fields would be filled with the reservation's current information and can be edited

© 2023 Name

## View Reservation Mobile



## View Reservation Web

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab Tab Tab Tab

### View Reservation

**General Information**

**Driver Information**

**Vehicle Information**

Start Date: 2023/09/25 01:23

End Date: 2023/09/25 01:23

Rider Quantity: 1

Accommodation Type: Name of accommodation

Accommodation Description: Description here

Location Pickup: 123 First St. 12345

Location Dropoff: 123 First St. 12345

Stop 1: 123 First St. 12345

Duration: 1 hour

Description: Enter description here...

Driver Name: Text

Email Contact: Text

Phone Contact: Text

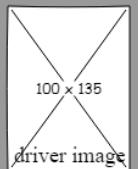
Vehicle Type: Text

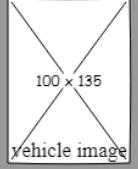
Model: Text

Color: Text

License Plate: Text

VIN Number: Text

 driver image

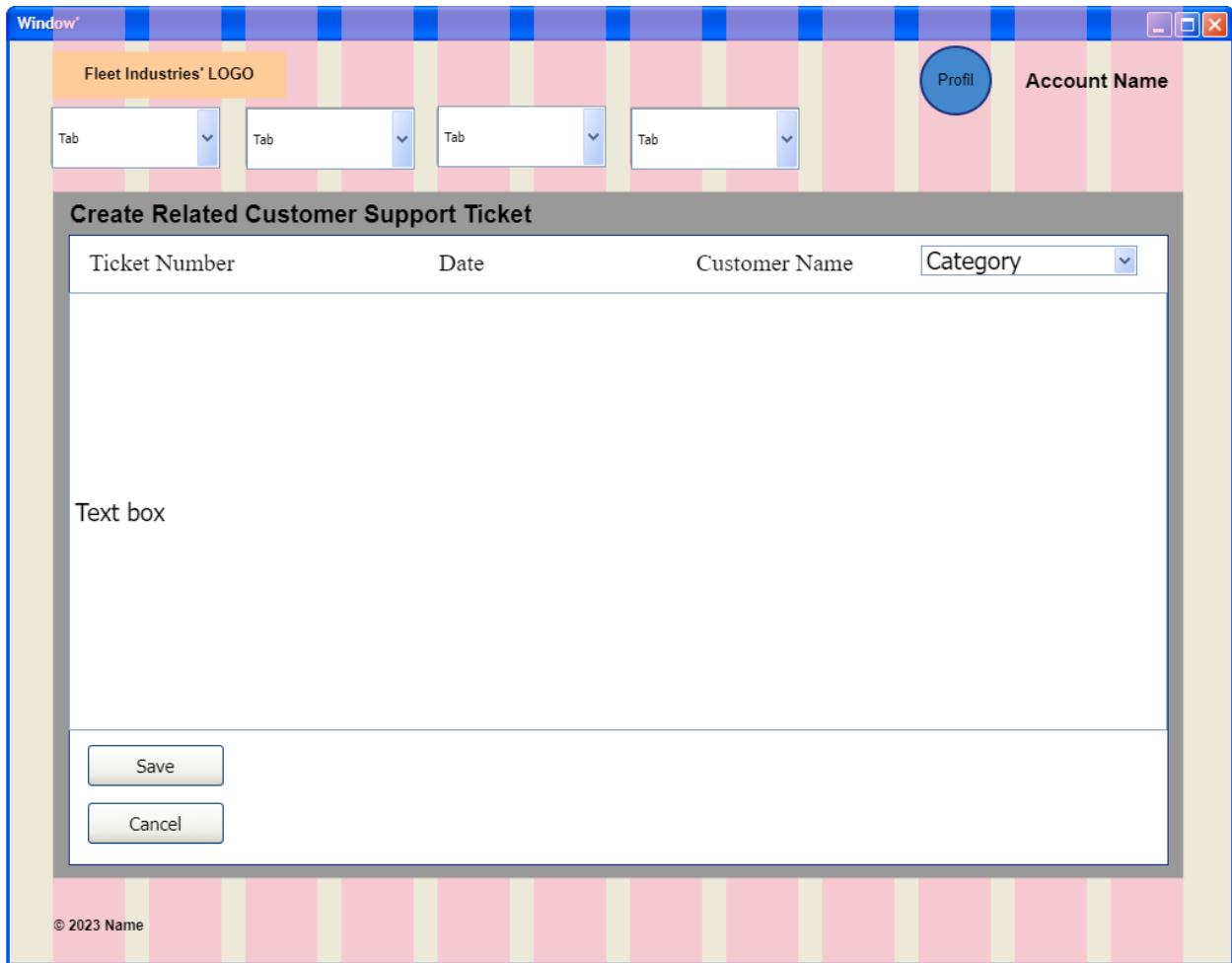
 vehicle image

Back Edit Cancel Reservation

© 2023 Name

# UI Drawings for Customer Support Representitive

## Create Support Ticket - Desktop



This UI drawing illustrates the 'Create Support Ticket' interface for a desktop application. The window title is 'Window'. The top bar includes a 'Fleet Industries' LOGO, four tabs, a 'Profil' button, and an 'Account Name' field. The main content area is titled 'Create Related Customer Support Ticket' and contains fields for 'Ticket Number', 'Date', 'Customer Name', and a dropdown menu for 'Category'. A large text box occupies the central area. At the bottom are 'Save' and 'Cancel' buttons, and a copyright notice '© 2023 Name'.

Window

Fleet Industries' LOGO

Tab Tab Tab Tab

Profil Account Name

Create Related Customer Support Ticket

Ticket Number Date Customer Name Category

Text box

Save Cancel

© 2023 Name

## Create Support Ticket - web

The screenshot shows a web browser window with a title bar labeled "Chrome". The main content area displays a web page for "Fleet Industries' Logo" under "Home > Products > Pencil". On the right side of the header, there are "Login" and "Register Now!" buttons. Below the header, there are four tabs labeled "Tab", "Tab", "Tab", and "Tab". The main content area has a dark grey header bar with the text "Create Related Customer Support Ticket". Below this, there is a form with fields for "Ticket Number", "Date", "Customer Name", and a dropdown menu for "Category". A large text area is labeled "Text box". At the bottom of the form, there are "Save" and "Cancel" buttons. The footer of the page contains the copyright notice "© 2023 Name".

Chrome

Fleet Industries' Logo Home > Products > Pencil

Login

Register Now!

Tab Tab Tab Tab

Create Related Customer Support Ticket

Ticket Number Date Customer Name Category

Text box

Save Cancel

© 2023 Name

## Update Support Ticket - Desktop

The screenshot shows a desktop application window titled "Update Customer Support Ticket". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a "Window" tab. In the top left corner, there is a placeholder for "Fleet Industries' LOGO". On the right side of the header, there is a circular "Profil" button and an "Account Name" field. Below the header, there is a toolbar with four tabs labeled "Tab" and a dropdown arrow. The main content area has a title "Update Customer Support Ticket" and a table with four columns: "Ticket Number", "Date", "Customer Name", and "Category". A large text area below the table is labeled "Support Ticket Details". At the bottom of the window, there are buttons for "Save", "Cancel", and "Create Related Ticket". A checkbox labeled "Resolved" is also present. The footer contains the copyright notice "© 2023 Name".

Ticket Number	Date	Customer Name	Category
Support Ticket Details			

Text box

Save  Resolved

Cancel Create Related Ticket

© 2023 Name

## Update Support Ticket - Web

The screenshot shows a web browser window with a pink and grey color scheme. At the top, there's a header bar with the text "Fleet Industries' Logo", "Home > Products > Pencil", and "Login / Register Now!". Below the header, there are four tabs labeled "Tab". The main content area has a dark grey header with the text "Update Customer Support Ticket". Below this, there's a table with columns: "Ticket Number", "Date", "Customer Name", and "Category". A large text area labeled "Support Ticket Details" follows. At the bottom, there are buttons for "Save", "Cancel", and "Create Related Ticket". A checkbox labeled "Resolved" is also present. The footer contains the text "© 2023 Name".

Ticket Number	Date	Customer Name	Category
Support Ticket Details			

Text box

Save       Resolved

Cancel      Create Related Ticket

© 2023 Name

## View All Support Tickets - Desktop

The screenshot shows a desktop application window titled "Customer Support Center". The window has a title bar with standard window controls (Minimize, Maximize, Close) and a logo for "Fleet Industries' LOGO". On the right side of the title bar, there is a blue circular button labeled "Profil" and a text field labeled "Account Name". The main content area is divided into sections: "Support Tickets" at the top, followed by a search/filter/sort section, and a table displaying ticket data. At the bottom of the content area is a "Back" button. The footer contains a copyright notice: "© 2023 Name".

**Customer Support Center**

**Support Tickets**

Ticket Number	Date Filed	Customer Name	Category	Status	Details
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>

Back

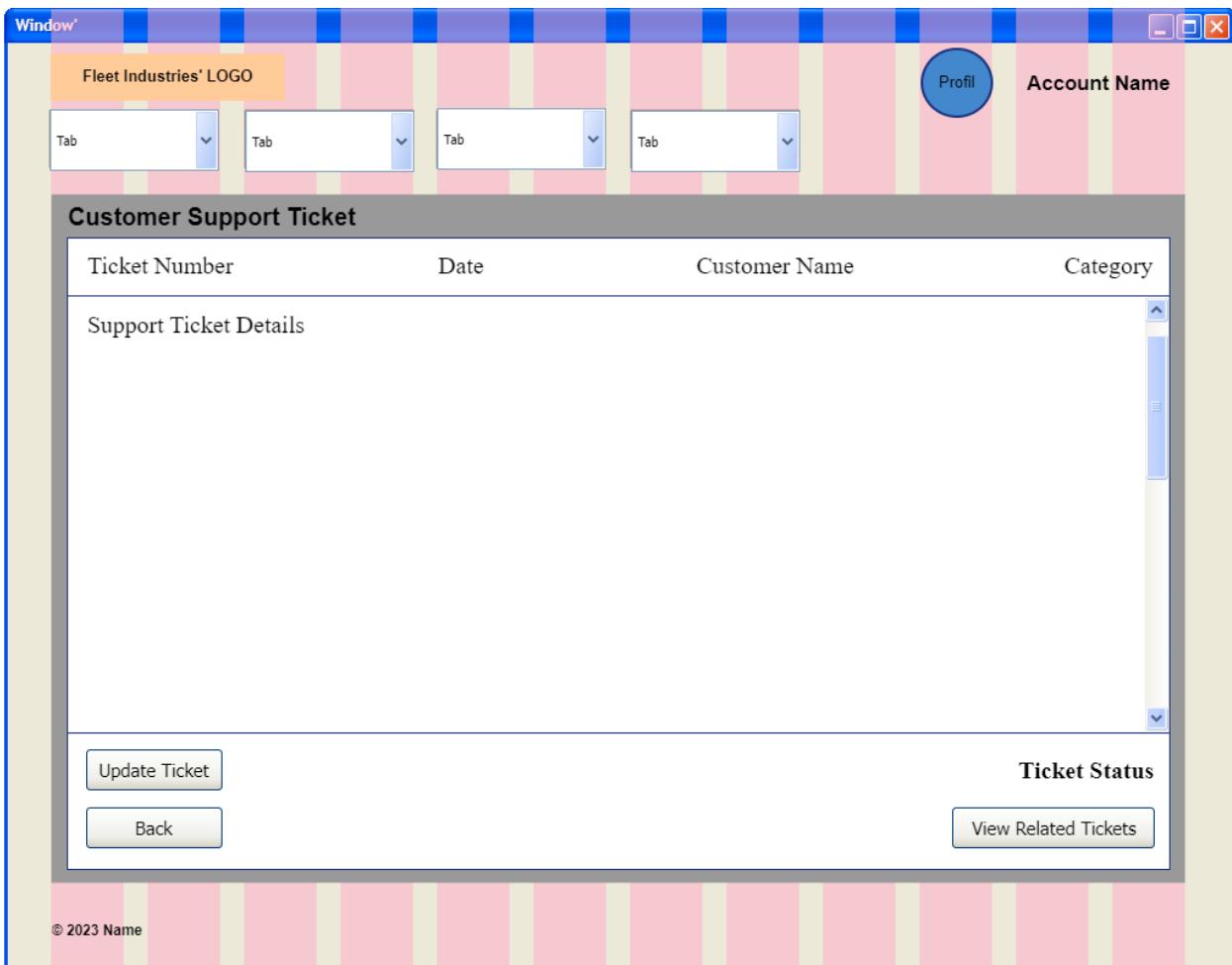
© 2023 Name

## View All Support Tickets - Web

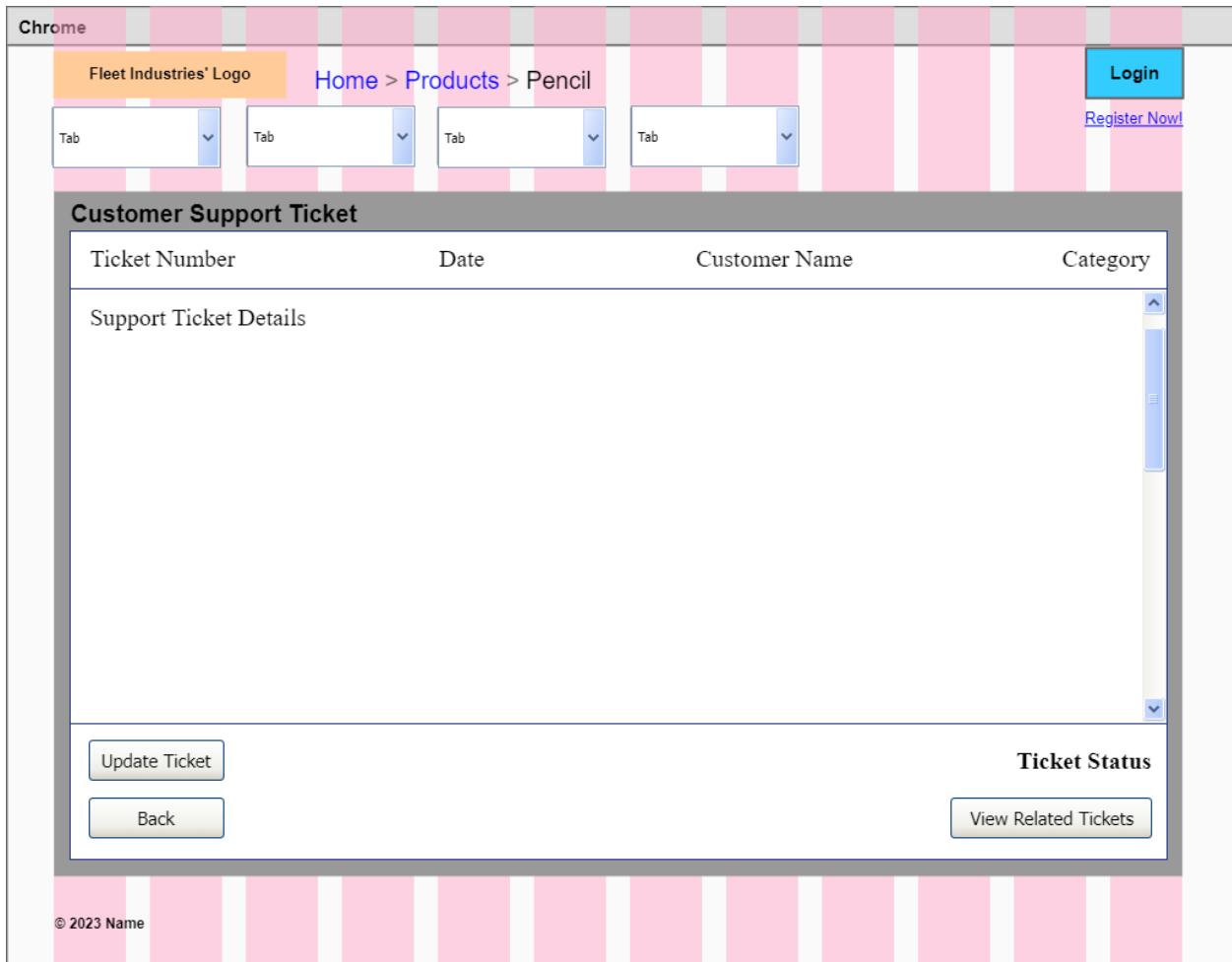
The screenshot shows a web browser window with a pink and grey color scheme. At the top, there's a header bar with tabs labeled 'Tab' and a navigation bar with 'Fleet Industries' Logo' and 'Home > Products > Pencil'. On the right side of the header are 'Login' and 'Register Now!' buttons. Below the header is a dark grey navigation bar with the text 'Customer Support Center'. Underneath is a white content area titled 'Support Tickets'. It features three input fields: 'Search' (text box), 'Filter' (combo box), and 'Sort' (combo box). A table follows, with columns: Ticket Number, Date Filed, Customer Name, Category, Status, and Details. Each row in the table contains placeholder text ('NUMBER', 'DATE', 'NAME', 'CATEGORY', 'STATUS', 'View'). At the bottom of the content area is a 'Back' button. The footer of the page includes the text '© 2023 Name'.

Ticket Number	Date Filed	Customer Name	Category	Status	Details
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>
NUMBER	DATE	NAME	CATEGORY	STATUS	<a href="#">View</a>

## View Support Ticket Details - Desktop

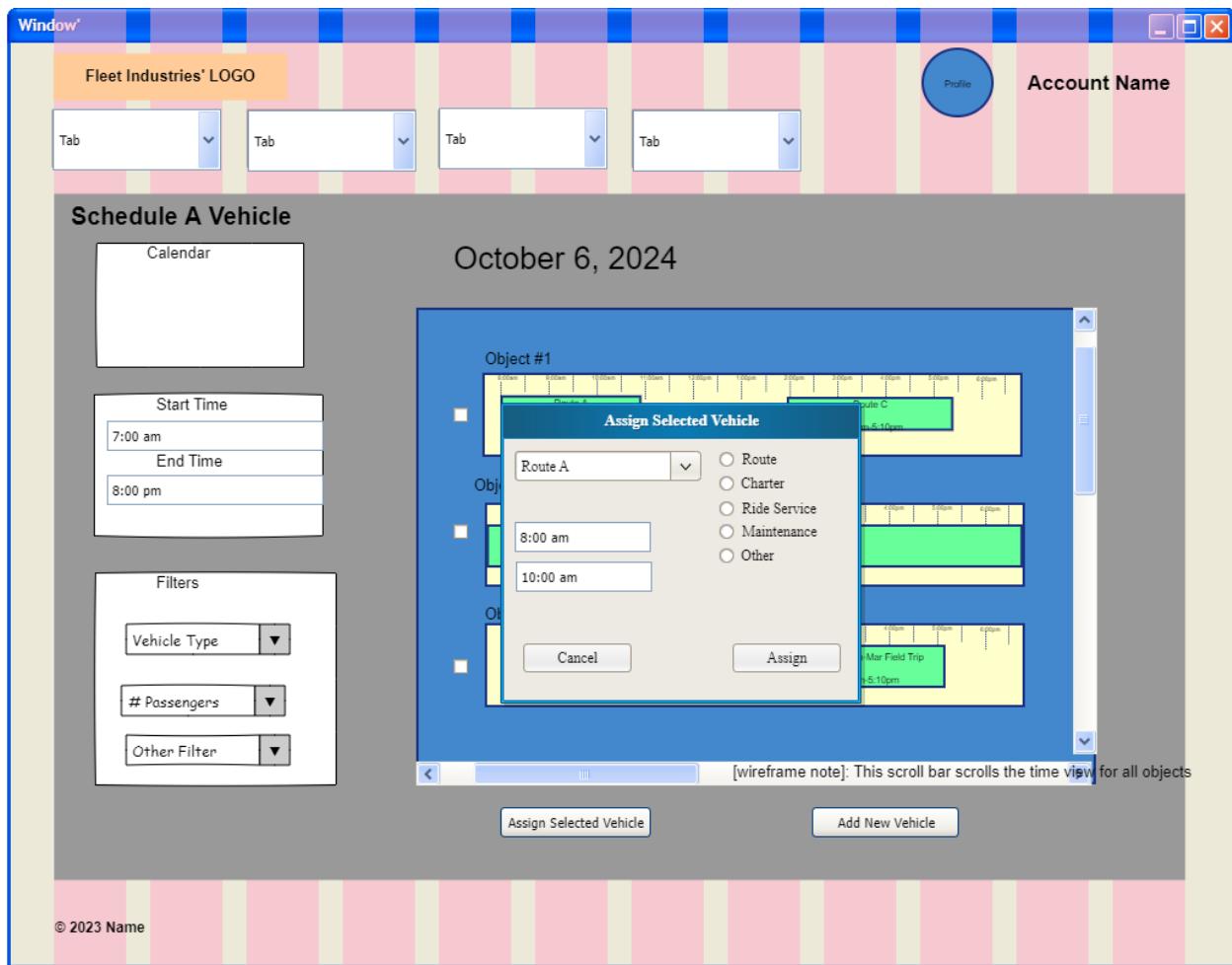


## View Support Ticket Details - Web

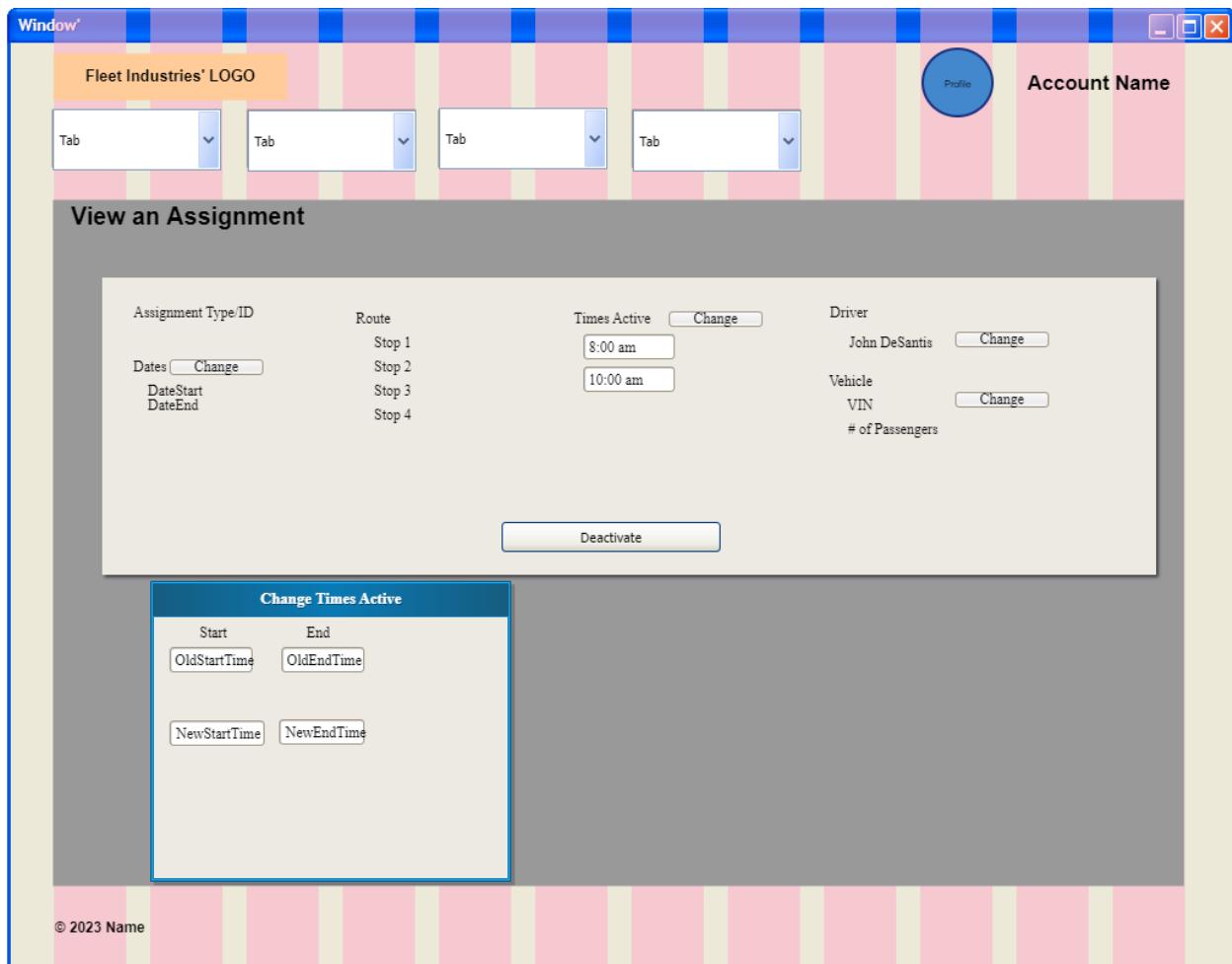


# UI Drawings for Dispatch

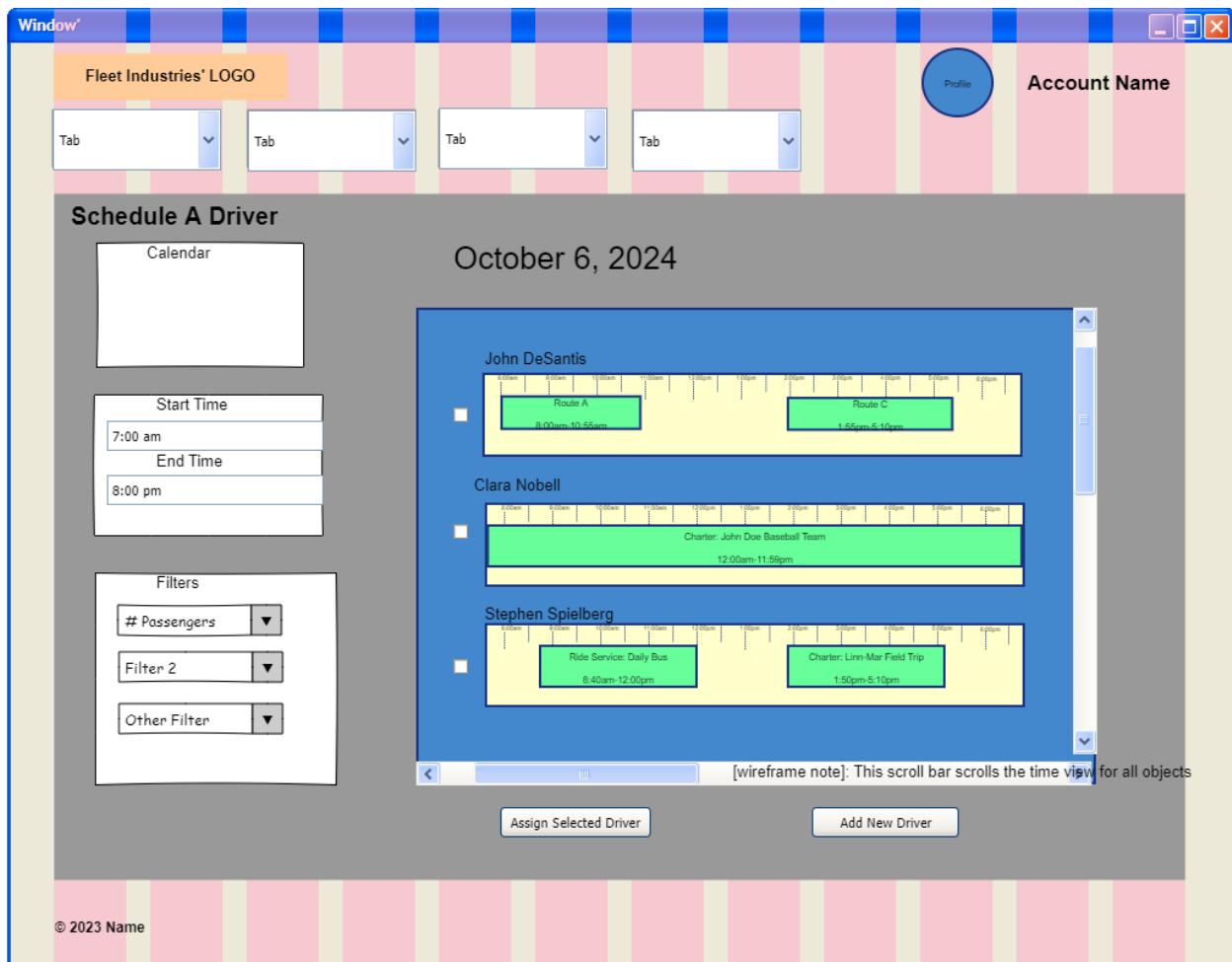
## Assign Selected Vehicle - Desktop View



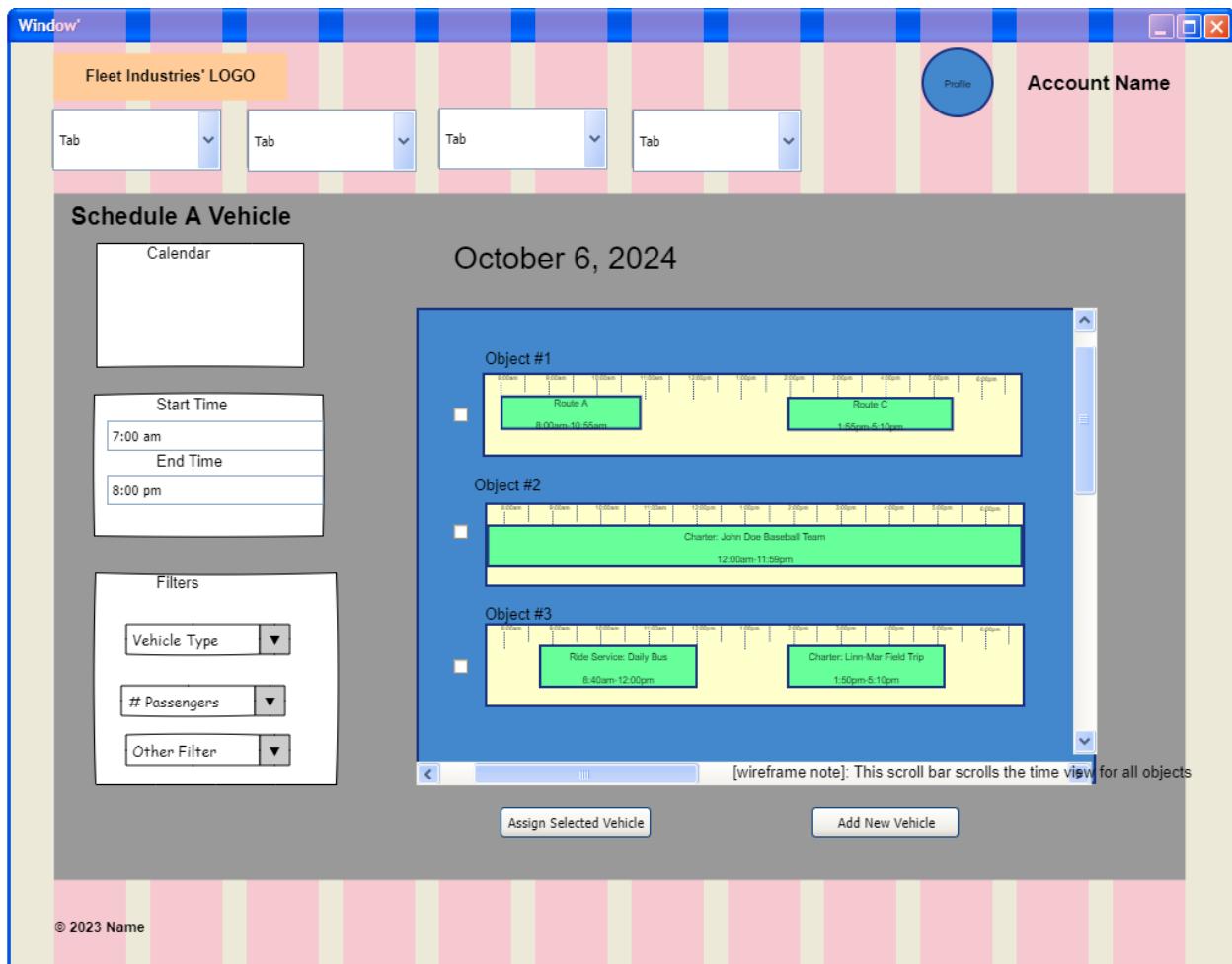
## Assignment View With Deactivate - Desktop View



## Schedule A Driver - Desktop View



## Schedule A Vehicle - Desktop View



# UI Drawings for Driver

## Driver Assigned Vehicle - Mobile



## Driver Assigned Vehicle - Web

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

My Assigned Vehicles

Image of selected vehicle  
223 x 162

List of Assigned Vehicles

- Vehicle 1 Model ►
- Vehicle 2 Model ►
- Vehicle ... Model ►

Vehicle ... Model

License Plate: Text box

Vin#: Text box

Mileage: Text box

Fuel Level: Text box

Vehicle History: History

Maintenance:  
Maintenance history

Checkout/Return:  
Checkout and return history

Vehicle Reports:  
recent trips, accidents, etc...

Additional Notes:

Text box

No Vehicles Available

You currently have no vehicles assigned at the moment!

Close

Confirm To Exit

No vehicle has been selected for checkout, continue to exit!

Confirm      Cancel

Vehicle Returned

Thank you for safely returning the vehicle!

Close

Checkout

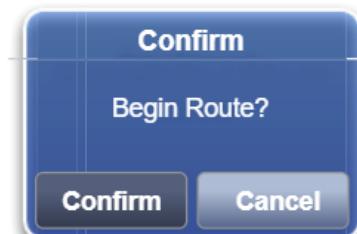
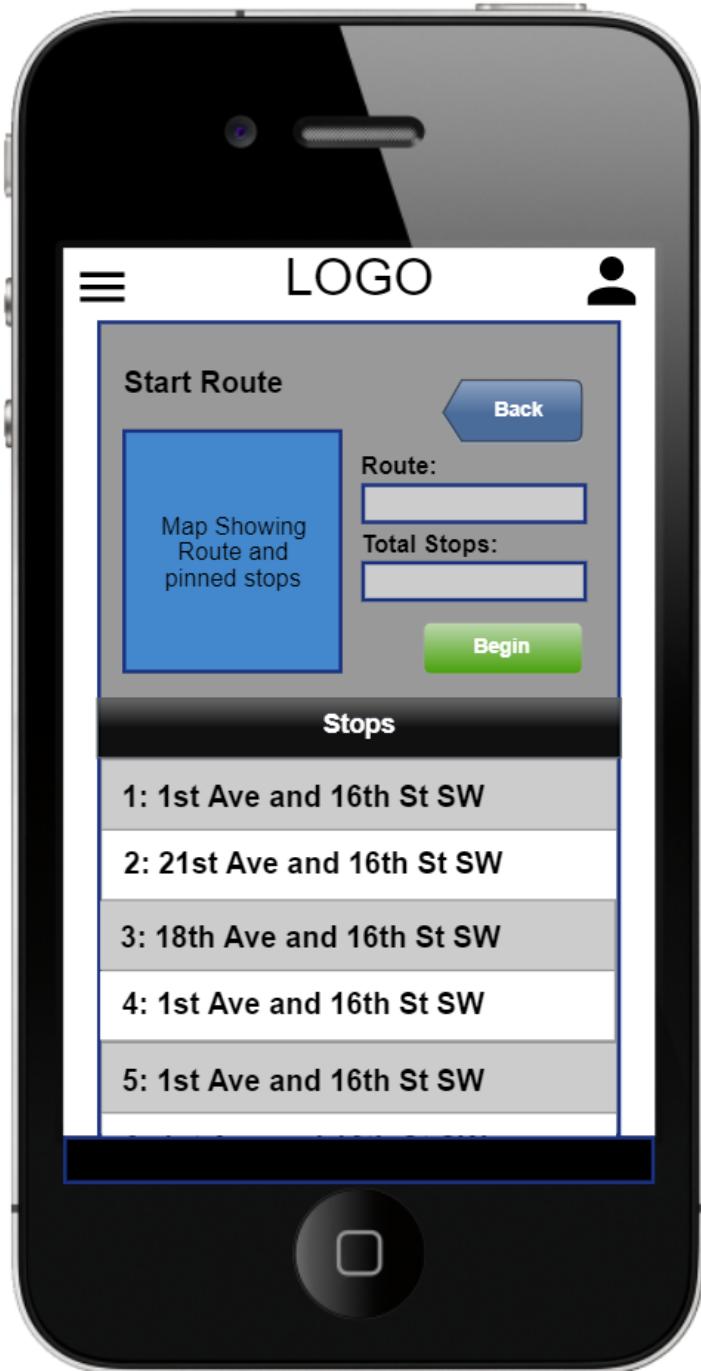
Your checkout has been confirmed, vehicle is ready for pick up!

Checkout      Return

© 2023 Name

## Driver Begin Route - Mobile

### DRV.01.10.mobile Start Route



## Driver Begin Route - Web

### DRV.01.10.web Start Route

Chrome

Fleet Industries' Logo    Home > Products > Pencil    Login    Register Now!

Start Route

Map of route with pinned stops  
285 x 227

Route Info

Route:

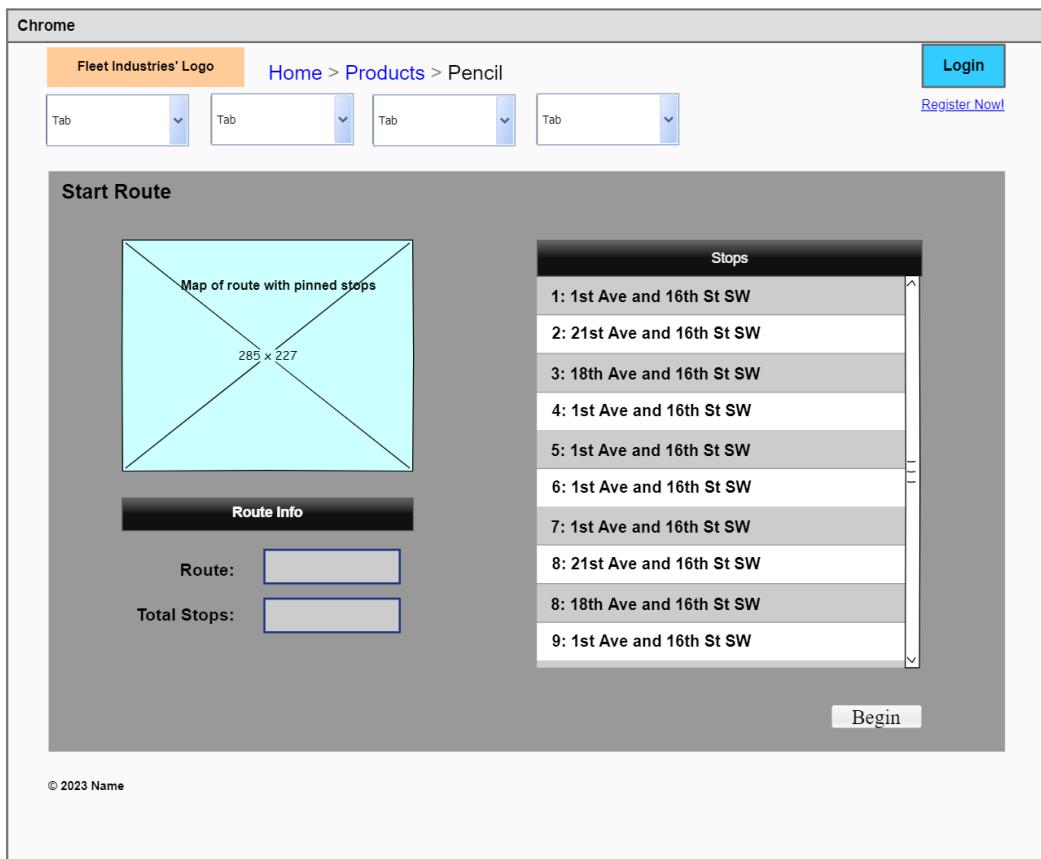
Total Stops:

Stops

1: 1st Ave and 16th St SW
2: 21st Ave and 16th St SW
3: 18th Ave and 16th St SW
4: 1st Ave and 16th St SW
5: 1st Ave and 16th St SW
6: 1st Ave and 16th St SW
7: 1st Ave and 16th St SW
8: 21st Ave and 16th St SW
9: 18th Ave and 16th St SW
9: 1st Ave and 16th St SW

Begin

© 2023 Name



## Driver Dropoff Passenger - Web

### DRV.01.09.web Passenger Dropoff

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Passenger Dropoff

Live Map  
285 x 227

Route Info

Route: [ ]  
Stop: [ ]

Passenger List

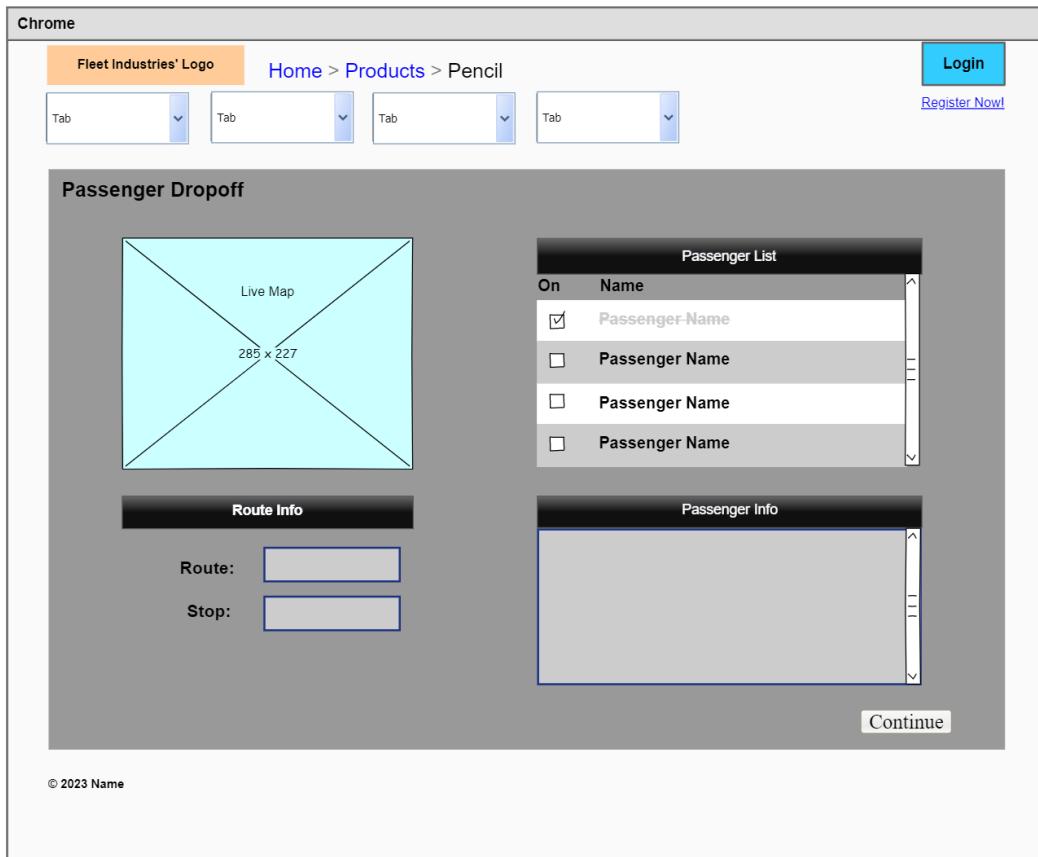
On	Name
<input checked="" type="checkbox"/>	Passenger Name
<input type="checkbox"/>	Passenger Name
<input type="checkbox"/>	Passenger Name
<input type="checkbox"/>	Passenger Name

Passenger Info

[Large Placeholder Area]

Continue

© 2023 Name



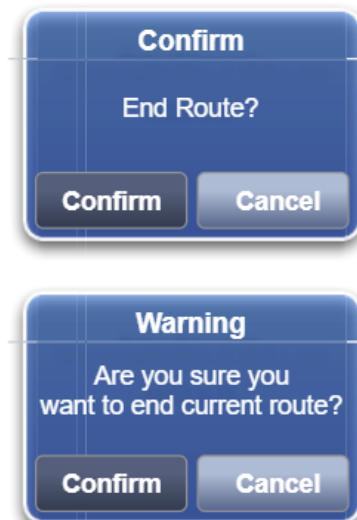
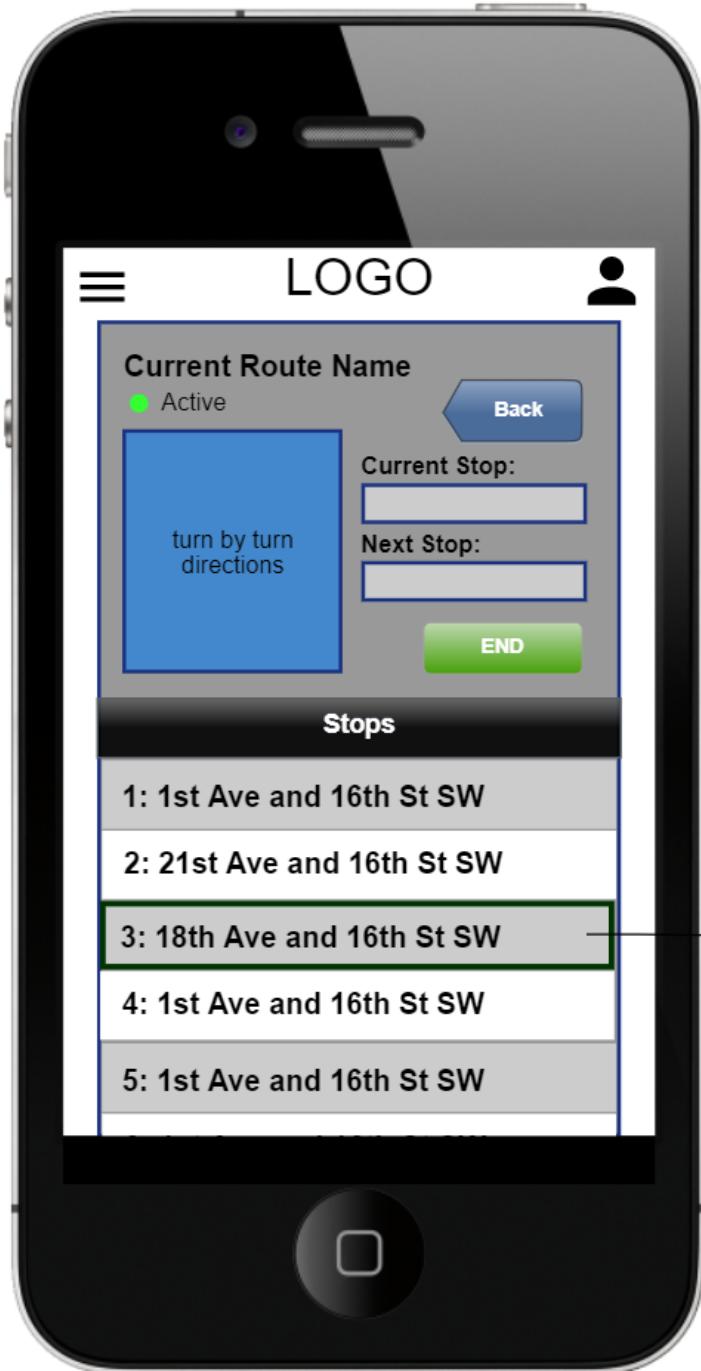
## Driver Dropoff Passenger -Mobile

### DRV.01.09.mobile Passenger Dropoff



## Driver End Route - Mobile

### DRV.01.11.mobile End Route



Border indicates current location on route

## Driver End Route - Web

### DRV.01.11.web End Route

The screenshot shows a web browser window for 'Fleet Industries' Logo' with the URL 'Home > Products > Pencil'. The browser tabs are labeled 'Tab', 'Tab', 'Tab', and 'Tab'. On the right, there are 'Login' and 'Register Now!' buttons.

**Route Name**  
Active

**Map of route with pinned stops**  
285 x 227

**Route Info**

Current Stop:

Next Stop:

**Stops**

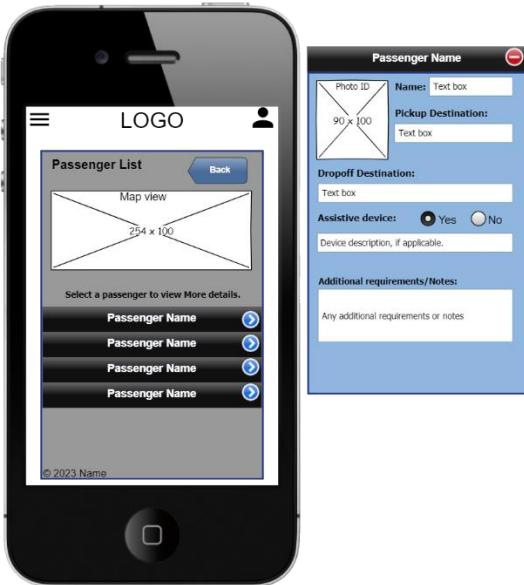
- 1: 1st Ave and 16th St SW
- 2: 21st Ave and 16th St SW
- 3: 18th Ave and 16th St SW
- 4: 1st Ave and 16th St SW**
- 5: 1st Ave and 16th St SW
- 6: 1st Ave and 16th St SW
- 7: 1st Ave and 16th St SW
- 8: 21st Ave and 16th St SW
- 9: 18th Ave and 16th St SW

**END**

© 2023 Name

## Driver Passenger List - Mobile

DRV.01.07.mobile



## Driver Passenger List - Web

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Passenger List

Map view with passengers location  
316 x 210

Passenger Name  
Photo ID  
Name: Text box

Pickup Destination: Text box

Dropoff Destination: Text box

Assistive Device:  Yes  No

Device description, if applicable.

Additional Requirements/Notes:

Any additional requirements or notes.

Back

© 2023 Name

The screenshot shows a web browser window for 'Fleet Industries' logo. The URL is 'Home > Products > Pencil'. There are four tabs labeled 'Tab' with dropdown arrows. On the right, there are 'Login' and 'Register Now!' buttons. The main content area is titled 'Passenger List'. It includes a placeholder for a map view ('Map view with passengers location') with dimensions 316 x 210. Below it is a 'Passenger Name' section with a 'Photo ID' field and a 'Name:' text box. Further down are fields for 'Pickup Destination' and 'Dropoff Destination', both as text boxes. An 'Assistive Device' section has radio buttons for 'Yes' and 'No', with 'No' being selected. A note field asks for 'Device description, if applicable.' Below that is a section for 'Additional Requirements/Notes' with a text area for 'Any additional requirements or notes.' At the bottom left is a copyright notice '© 2023 Name' and a 'Back' button at the bottom right.

## Driver Pickup Passenger - Mobile

### DRV.01.08.mobile Passenger Pickup



## Driver Pickup Passenger -Web

### DRV.01.08.web Passenger Pickup

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Passenger Pickup

Live Map  
285 x 227

Route Info

Route: [ ]  
Stop: [ ]

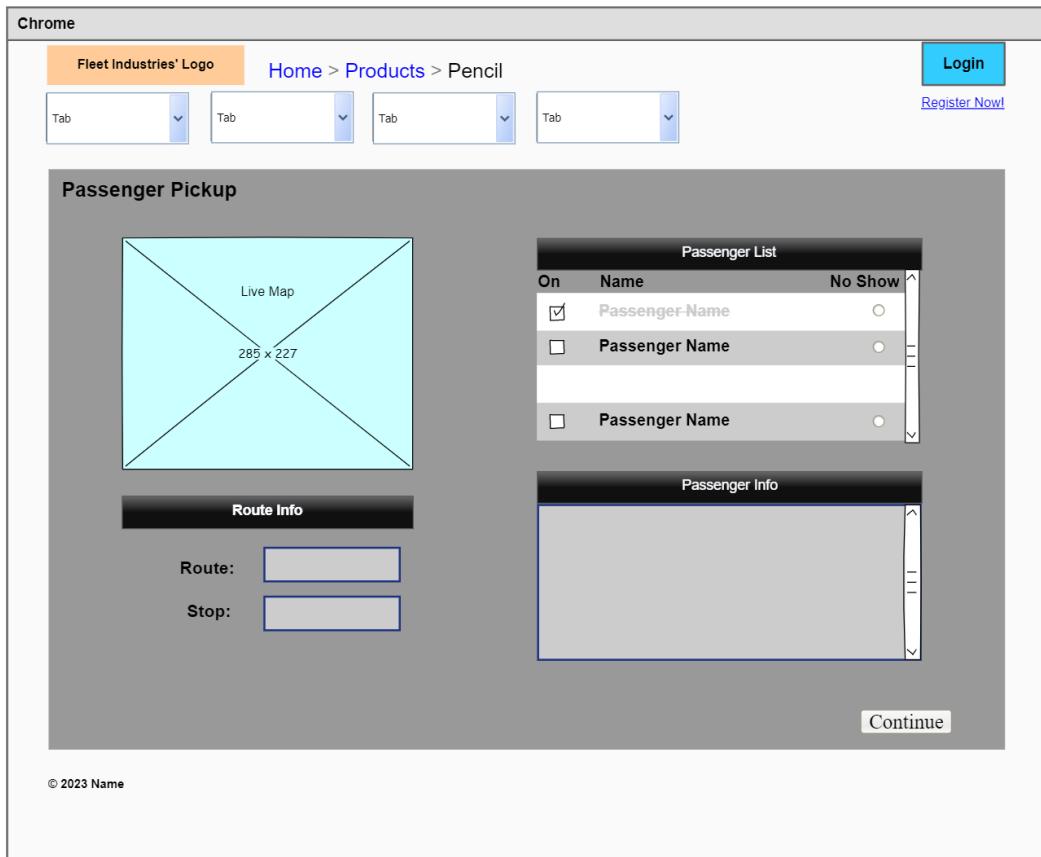
Passenger List

On	Name	No Show
<input checked="" type="checkbox"/>	Passenger Name	<input type="radio"/>
<input type="checkbox"/>	Passenger Name	<input type="radio"/>
<input type="checkbox"/>	Passenger Name	<input type="radio"/>

Passenger Info

Continue

© 2023 Name



## Driver Refuel Log - Mobile

DRV.01.04.mobile Refuel Log



**Purchase Log:** Fields for Mileage (HTML Text Field), PPG (\$0.00), Gallons, and Total (\$0.00). Includes a 'Back' button.

**Documentation:** Fields for Driver (HTML Text Field), Date (Date Picker), and Notes (HTML Text Field).

**Confirm Log:** Buttons for 'Submit Log', 'Confirm', and 'Cancel'. A success message 'Success! Log Submitted' and an 'Okay' button are shown below.

**Attention:** Message 'Errors in document. Please revise' with 'OK' and 'Submit Anyway' buttons.

## Driver Refuel Log - Web

### DRV.01.04.web Refuel Log

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Refuel Log

285 x 227

Purchase Log

Mileage	Text box
PPG	\$0.00
Gallons	Text box
Total	\$0.00

Notes

Text box

Vehicle Information

VIN:

Fuel Type:

Documentation

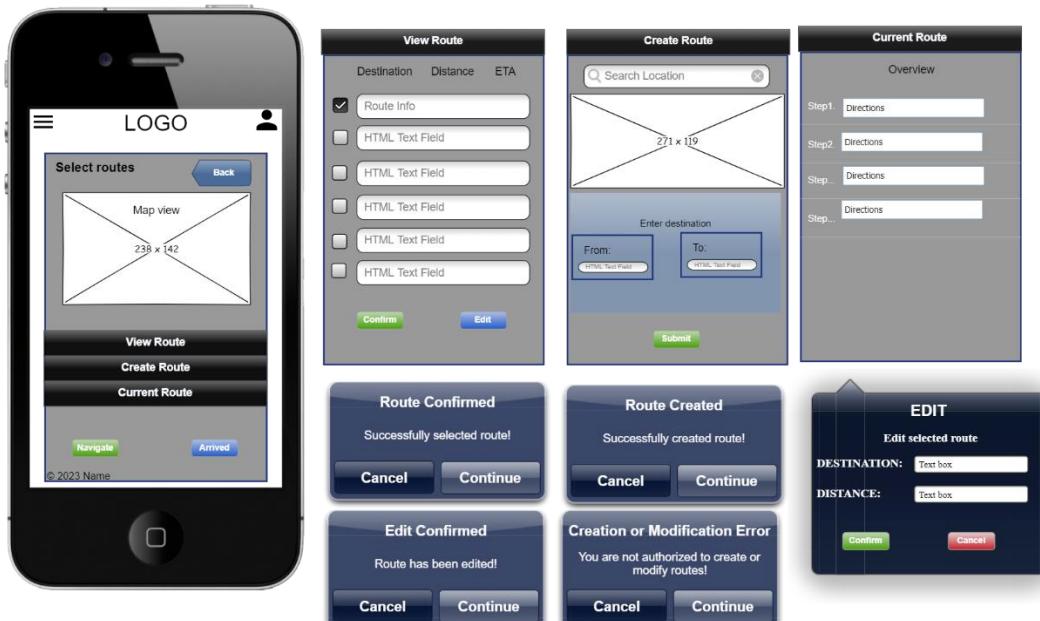
Name   
Date

Create      Cancel

© 2023 Name

## Driver Select Routes - Mobile

DRV.01.05.Mobile



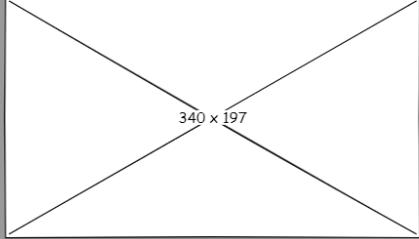
## Driver Select Route - Web

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab      Tab      Tab      Tab

**Select Routes**



**View Routes**

	DESTINATION	DISTANCE	ETA
<input checked="" type="checkbox"/>	Destination Info...	Distance Info...	ETA Info...
<input type="checkbox"/>	Destination Info...	Distance Info...	ETA Info...
<input type="checkbox"/>	Destination Info...	Distance Info...	ETA Info...

**Create Route**

FROM:  Text box

TO:  Text box

Submit

**Current Route**

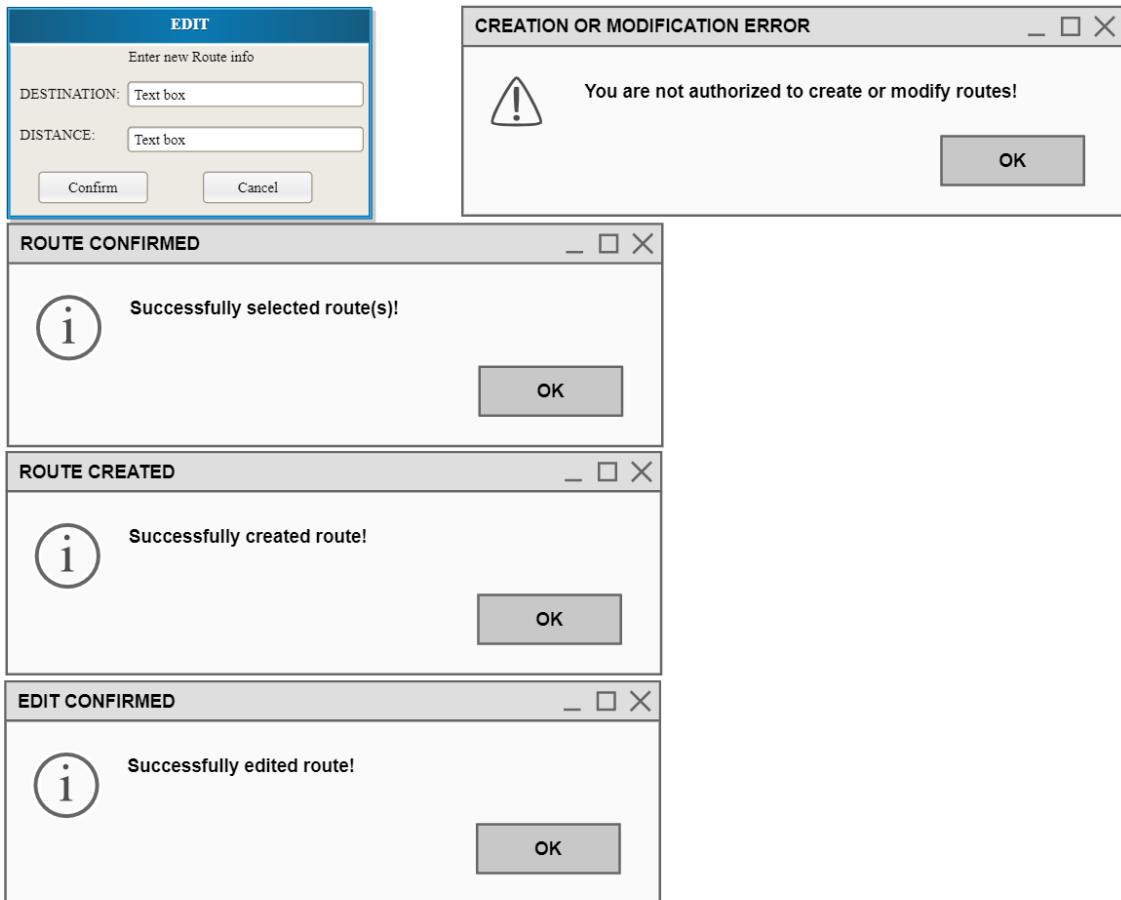
**STEP-BY-STEP DIRECTIONS**

Step 1	Directions...
Step 2	Directions...
Step 3	Directions...
Step ...	Directions...

Navigate       End Navigation

© 2023 Name

## Driver Select Routes Popups - Web



## Driver Submit Maintenance Report - Mobile

### DRV.01.13.mobile Submit Maintenance Report



## Driver Submit Maintenance Report - Web

### DRV.01.13.web Submit Maintenance Report

Chrome

Fleet Industries' Logo    Home > Products > Pencil    Login    Register Now!

Tab    Tab    Tab    Tab

Maintenance Report

EmployeeID:	Text box
VIN:	Text box
License Plate:	Text box
Severity:	Low
Description:	Text box

Submit    Cancel

© 2023 Name

Severity

Low
Moderate
High

## Driver Submit Safety Report - Mobile

### DRV.01.12.mobile Submit Safety Report



## Driver Submit Safety Report - Web

### DRV.01.12.web Submit Safety Report

Chrome

Fleet Industries' Logo    Home > Products > Pencil    Login    Register Now!

Tab Tab Tab Tab

**Safety Report**

**EmployeeID:**  Text box

**Date:**  Date Picker

**Time of Event:**  Text box

**Affected Party:**

**Description of the Event:**

Event Resulted in Injury

Affected Party Dropdown Options:

Yourself	▼
Co-Worker	▼
Passenger	▼
Civilian	▼

© 2023 Name

## Driver Vehicle Checkout - Mobile

DRV.01.01.mobile Vehicle Checkout



The mobile application interface for Driver Vehicle Checkout (DRV.01.01.mobile) displays the following sections:

- Vehicle Information:** Includes fields for License Plate and VIN.
- Interior:** Status for Clean, Steering, Pedals, AC/Heat, Dash Display, and Rear View Mirror, each with a switch labeled "ON".
- Exterior:** Status for Cosmetic, Tire Pressure, and Blinkers, each with two switches labeled DF: ON and PF: ON.
- Safety Equipment:** Status for Seatbelts, Airbags, First Aid Kit, Fire Extinguisher, Emergency Kit, and Notes.
- Mechanical:** Status for Mileage, Fuel Level, Breaks, Clutch, Accelerator, and Notes.
- Documentation:** Status for Driver, Date, and Notes.
- Notes:** HTML Text Fields for various sections.
- Confirm Checkout:** Buttons for Confirm, Cancel, Success!, Okay, and Maintenance/Return.
- Attention:** Message about errors and maintenance.

## Driver Vehicle Checkout - Web

### DRV.01.01.web Vehicle Checkout

Chrome

Fleet Industries' Logo    Home > Products > Pencil    Login    Register Now!

Vehicle Checkout

285 x 227

**Vehicle Information**

License Plate:     VIN:

**Documentation**

Name:     Date:

**Interior**

Clean	<input type="checkbox"/>	Steering	<input type="checkbox"/>
Pedals	<input type="checkbox"/>	AC/Heat	<input type="checkbox"/>
Dash	<input type="checkbox"/>	Mirror	<input type="checkbox"/>

**Exterior**

Cosmetic

Tire Pressure:

DF	<input type="Text box"/>	PF	<input type="Text box"/>
DR	<input type="Text box"/>	PR	<input type="Text box"/>

Belts     Fire Exting.   
Airbags     First Aid   
Emergency Kit

**Mechanical**

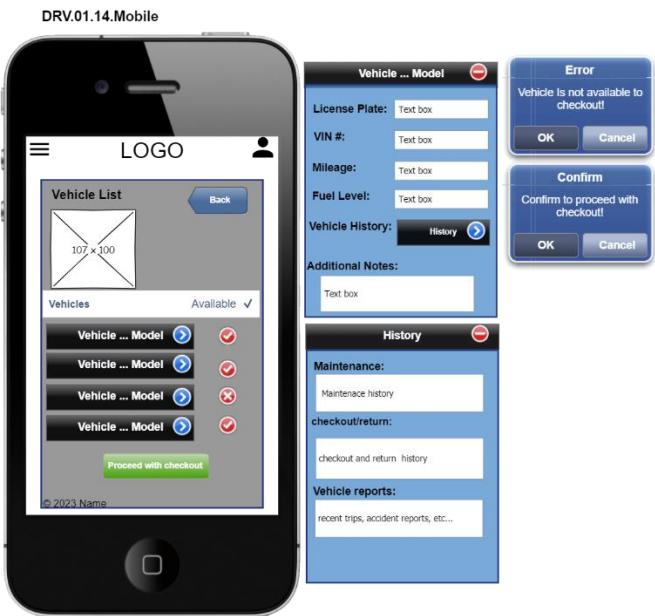
Mileage   
Fuel Level   
Breaks  Accelerator   
Clutch

**Notes:**

Create    Cancel

© 2023 Name

## Driver Vehicle List - Mobile



## Driver Vehicle List - Web

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Vehicle List

Image of selected vehicle  
213 x 176

Available	Vehicles
<input checked="" type="checkbox"/>	Vehicle Model Name 
<input checked="" type="checkbox"/>	Vehicle Model Name 
<input checked="" type="checkbox"/>	Vehicle Model Name 
<input checked="" type="checkbox"/>	Vehicle Model Name 
<input checked="" type="checkbox"/>	Vehicle Model Name 
<input type="checkbox"/>	Vehicle Model Name 
<input checked="" type="checkbox"/>	Vehicle Model Name 
<input type="checkbox"/>	Vehicle Model Name 

Vehicle Model Name ▼

License Plate: Text box

Vin#: Text box

Mileage: Text box

Fuel Level: Text box

Vehicle History: History ▼

Maintenance: Maintenance history

Checkout/Return: Checkout and return history

Vehicle Reports: recent trips, accidents, etc....

Additional Notes:

Text box

CONFIRM

Vehicle is not available for checkout!

Close

Confirm to proceed with checkout!

Cancel      Confirm

Proceed With Checkout      Back

© 2023 Name

## Driver Vehicle Return - Mobile

DRV.01.03.mobile Vehicle Return

The image displays a mobile application interface for vehicle return. On the left, a smartphone screen shows a navigation menu with options like 'Vehicle Checkout', 'Vehicle Information', 'Interior', 'Exterior', 'Safety Equipment', 'Mechanical', 'Documentation', and a 'Submit' button. The main area is divided into three large sections: 'Vehicle Information', 'Exterior', and 'Safety Equipment'. Each section contains various vehicle status items with checkboxes and dropdown menus. To the right, a detailed view of the 'Vehicle Information' section is shown, listing items such as 'Clean', 'Steering', 'Pedals', 'AC/Heat', 'Dash Display', 'Rear View Mirror', and 'Notes'. Each item has a status indicator ('ON') and a checkbox. Below these are sections for 'Documentation' (Driver, Date, Notes) and 'Mechanical' (Mileage, Fuel Level, Breaks, Clutch, Accelerator, Notes). A separate 'Attention' box on the right side indicates errors in the vehicle return process.

Vehicle Information		Exterior		Safety Equipment	
VIN:	HTML Text Field	Cosmetic:	HTML Text Field	Seatbelts:	ON
Interior		Tire Pressure:	HTML Text Field	Airbags:	ON
Clean:	ON	DF: Text	PF: Text	First Aid Kit:	ON
Steering:	ON	DR: Text	PR: Text	Fire Extinguisher:	ON
Pedals:	ON	DR: ON	PR: ON	Emergency Kit:	ON
AC/Heat:	ON	DR: ON	PR: ON	Notes:	HTML Text Field
Dash Display:	ON	Break Lights:	ON	Mechanical	
Rear View Mirror:	ON	DF: ON	PF: ON	Mileage:	HTML Text Field
Notes:	HTML Text Field	DR: ON	PR: ON	Fuel Level:	HTML Text Field
Documentation		Head/Tail Lights:	ON	Breaks:	ON
Driver:	HTML Text Field	DF: ON	PF: ON	Clutch:	ON
Date:	HTML Text Field	DR: ON	PR: ON	Accelerator:	ON
Notes:	HTML Text Field	Wipers:	ON	Notes:	HTML Text Field
		DF: ON	PF: ON		
		Notes:	HTML Text Field		

**Confirm Checkout**

Confirm Vehicle Return?

**Success!**  
The vehicle has been returned.

**Attention**  
Vehicle Return is showing errors.  
Return to the form or submit a maintenance report?

**Maintenance**    **Return**

## Driver Vehicle Return - Web

### DRV.01.03.web Vehicle Return

Chrome

Fleet Industries' Logo    Home > Products > Pencil    Login    Register Now!

Vehicle Return

285 x 227

**Vehicle Information**

VIN:

**Documentation**

Name:   
Date:

**Interior**

Clean	<input type="checkbox"/>	Steering	<input type="checkbox"/>
Pedals	<input type="checkbox"/>	AC/Heat	<input type="checkbox"/>
Dash	<input type="checkbox"/>	Mirror	<input type="checkbox"/>

**Exterior**

Cosmetic	<input type="text"/>
Tire Pressure:	DF <input type="text"/> PF <input type="text"/>
DR	<input type="text"/> PR <input type="text"/>

**Safety Equipment**

Belts	<input type="checkbox"/>	Fire Exting.	<input type="checkbox"/>
Airbags	<input type="checkbox"/>	First Aid	<input type="checkbox"/>
Emergency Kit	<input type="checkbox"/>		

**Mechanical**

Mileage	<input type="text"/>
Fuel Level	<input type="text"/>
Breaks	<input type="checkbox"/> Accelerator <input type="checkbox"/>
Clutch	<input type="checkbox"/>

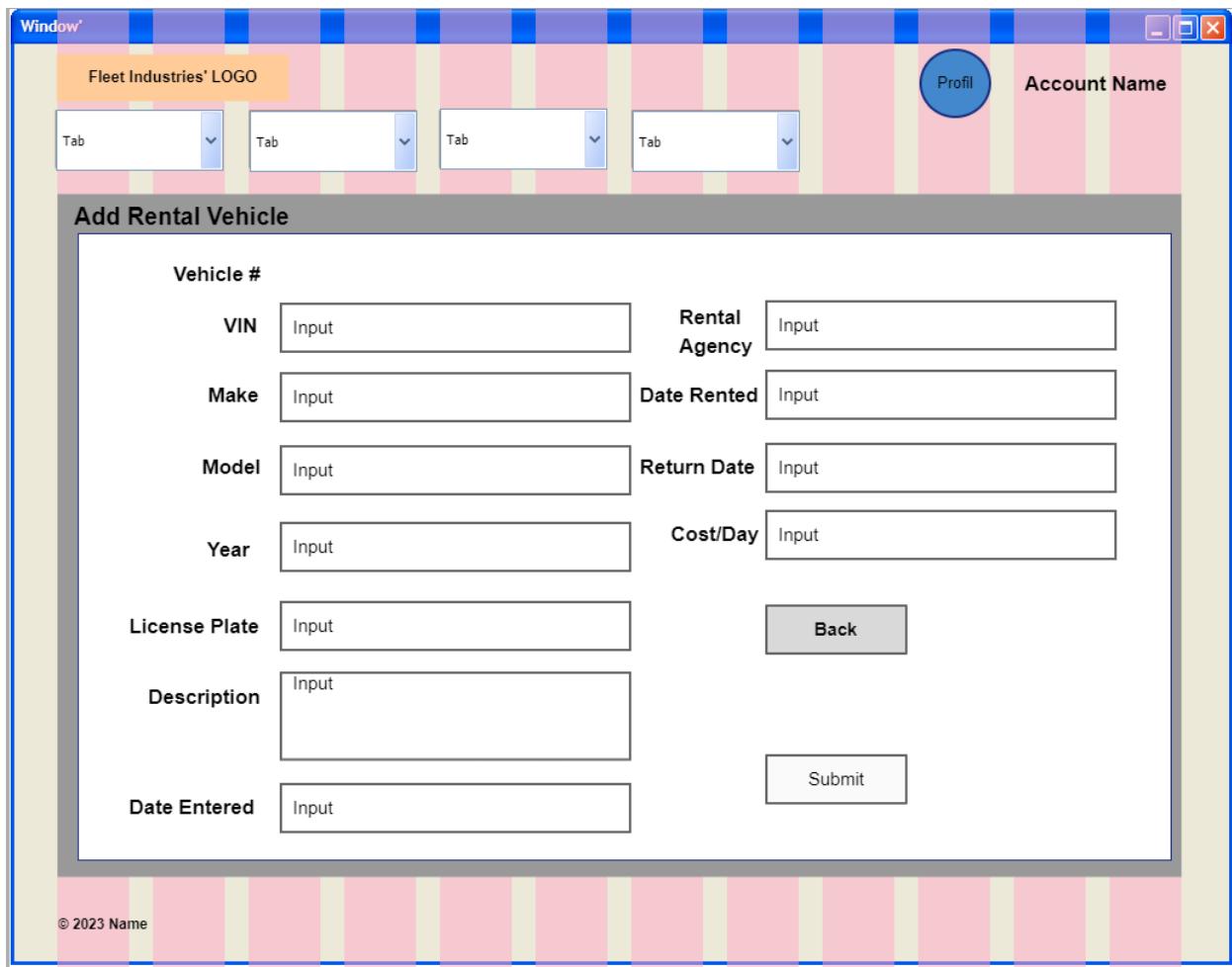
**Notes:**

Create    Cancel

© 2023 Name

# UI Drawings for Fleet Admin

## Add Rental Vehicle Desktop View



The UI drawing illustrates the desktop view for adding a rental vehicle. The window title is "Window". The top bar includes the "Fleet Industries' LOGO", "Account Name", and a "Profil" button. Below the title bar, there are four tabs labeled "Tab" with dropdown arrows. The main content area is titled "Add Rental Vehicle" and contains a form for "Vehicle #". The form fields are as follows:

Label	Input Type	Label	Input Type
VIN	Input	Rental Agency	Input
Make	Input	Date Rented	Input
Model	Input	Return Date	Input
Year	Input	Cost/Day	Input
License Plate	Input	Back	
Description	Input	Submit	
Date Entered	Input		

At the bottom left of the form, there is a copyright notice: "© 2023 Name".

## Add Vehicle Destkop View

The screenshot shows a desktop application window titled "Add Vehicle Info". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a logo for "Fleet Industries' LOGO". On the right side of the header, there is a circular button labeled "Profil" and the text "Account Name". Below the header, there are four tabs labeled "Tab" with dropdown arrows. The main content area is titled "Vehicle #". It contains eight input fields for vehicle information:

Field	Type
VIN	Input
Make	Input
Model	Input
Year	Input
License Plate	Input
Description	Input
Date Entered	Input
Seat Count	Input

On the right side of the form, there is a large empty rectangular area with a watermark-like diagonal cross and the text "275 x 261" in the center. Below this area are three buttons: "Back", "Maintenance", and "Submit". At the bottom left of the form, there is a copyright notice: "© 2023 Name".

## Change Vehicle Info Desktop View

The screenshot shows a desktop application window titled "Edit Vehicle Info". The window has a title bar with "Window" and other standard icons. In the top right corner, there is a blue circular button labeled "Profil" and an "Account Name" field. The main content area is titled "Edit Vehicle Info" and contains a form for entering vehicle information. The form fields are:

- Vehicle # <vehicle number>**
- VIN**: Input field
- Make**: Input field
- Model**: Input field
- Year**: Input field
- License Plate**: Input field
- Description**: Input field
- Date Entered**: Input field
- Seat Count**: Input field

On the right side of the form, there is a large square placeholder with a diagonal cross and the dimensions "275 x 261". Below the form, there are several buttons:

- Back**
- Maintenance**
- Track Vehicle**
- Submit**
- Deactivate**

At the bottom left of the window, there is a copyright notice: "© 2023 Name".

## Deactivate Vehicle Desktop View

The screenshot shows a desktop application window titled "Deactivate Vehicle". The window has a standard Windows-style title bar with "Window" and other icons. In the top right corner, there is a blue circular button labeled "Profil" and the text "Account Name". The main content area is titled "Deactivate Vehicle". It contains the following fields:

- Vehicle #**: vehicle number
- VIN**: VIN
- Make**: Make
- Model**: Model
- Year**: Year
- License Plate**: License Plate
- Reason**: Input field
- Date Deactivated**: Input field

On the right side of the form, there is a large square placeholder with a diagonal cross through it, labeled "275 x 261". Below the form are two buttons: "Back" and "Submit". At the bottom left of the window, there is a small copyright notice: "© 2023 Name".

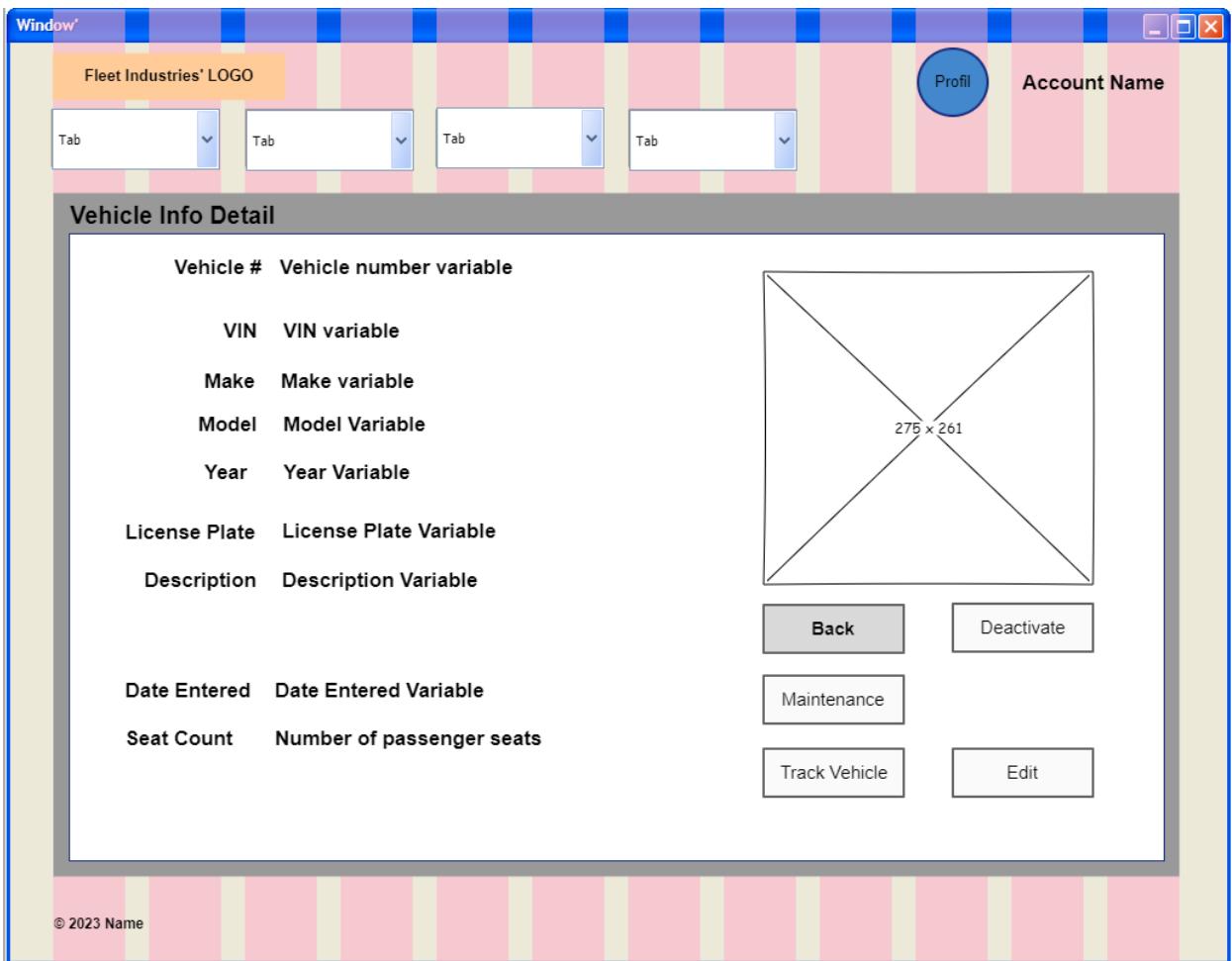
## Fuel Receipt List Destkop View



## Lookup Vehicle Desktop View



## Lookup Vehicle Info Desktop View



Lookup Vehicle Inspection Report Desktop View



## UI Drawings for Fleet Operations Manager

### Add Employee Alt

The screenshot shows a user interface for adding an employee. The form is contained within a dark gray rectangular area with a vertical scroll bar on the right side.

Hire Date	MM/DD/YYYY
Pay	\$ ___.__ /hr
Manager	Text box
Status	<input type="radio"/> Active <input checked="" type="radio"/> Non-Active
Termination Date	MM/DD/YYYY
<input type="button" value="Create"/>	

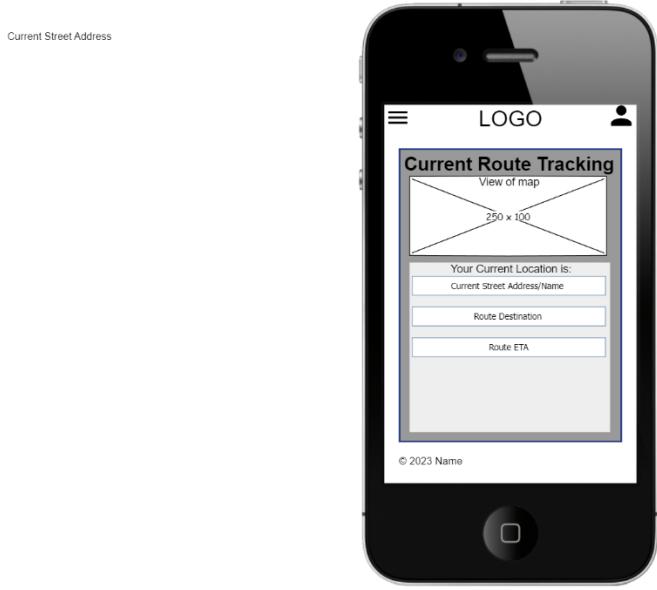
## Add Employee

The screenshot shows a Windows application window titled "Add Employee". The window has a title bar with "Windows", "Fleet Industries' LOGO", and "Account Name". There are four tabs at the top labeled "Tab", "Tab", "Tab", and "Tab". A "Print" button is located in the top right corner. The main area is titled "Add Employee" and contains the following fields:

First Name	Text box
Last Name	Text box
Address 1	Text box
Address 2	Text box
City	Text box
State	ANSI   ▾
ZIP Code	Text box
Phone	(____) - _____
Phone (Alternate)	(____) - _____
Emergency Contact	Text box
Contact Relation	Text box
Contact Phone	(____) - _____
Contact Email	Text box

At the bottom left of the window, there is a copyright notice: "© 2023 Name".

## Current Route Tracking



## Employee Detail Alt

<b>Emergency Contact Name</b>	<input type="text"/>
<b>Contact Phone</b>	<input type="text"/>
<b>Hire Date</b>	<input type="text"/>
<b>Pay</b>	<input type="text"/>
<b>Active</b>	<input type="text"/>
<b>Termination Date</b>	<input type="text"/>
<input type="button" value="Back"/>	



## Employee Detail

The screenshot shows a Windows application window titled "Employee Detail". The window has a title bar with standard window controls (Minimize, Maximize, Close) and a menu bar labeled "File". In the top right corner, there is a "Profile" button and an "Account Name" field. The main content area is titled "Employee Info" and contains a sub-section titled "Viewing Employee Details". This section includes input fields for "First Name", "Last Name", "Phone Number 1", "Phone Number 2", "Address 1", "Address 2", and "Email", each with an associated "Input" placeholder. To the right of these fields is a large, empty rectangular area containing a watermark or placeholder image of a diamond shape with the text "275 x 251". At the bottom left of the main content area, there is a copyright notice: "© 2023 Name".

## Employee Lookup

Window

Fleet Industries' LOGO

Tab Tab Tab Tab Profil Account Name

### Employee Lookup

#### Employee List

Search Employee List

Filter

Employee ID	First Name	Last Name	Job Title	Admin
154887	Jacob	Wendt	Fleet Operations	<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 2	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 3	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 4	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 5	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 6	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 7	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 8	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>

[Back](#)

© 2023 Name

## Update Deactive Employee Alt

Hire Date

Pay

Manager

Status  Active  Non-Active

Termination Date

## Update Deactive Employee

The screenshot shows a Windows application window titled "Update Employee". The window has a title bar with "Windows", "Fleet Industries' LOGO", and "Account Name". There are four tabs at the top labeled "Tab", "Tab", "Tab", and "Tab". A "Print" button is located in the top right corner. The main area is titled "Update Employee" and contains the following fields:

Field	Type
First Name	Text box
Last Name	Text box
Address 1	Text box
Address 2	Text box
City	Text box
State	ANSI   ▾
ZIP Code	Text box
Phone	(____) - _____
Phone (Alternate)	(____) - _____
Emergency Contact	Text box
Contact Relation	Text box
Contact Phone	(____) - _____
Contact Email	Text box

At the bottom left of the window, there is a copyright notice: "© 2023 Name".

## View All Route detail

**Easier Representation of the Table**

Routes	Arrival	Departure	Driver
+ Destination	11:00 AM	11:45 AM	John Doe
+ Destination	11:00 AM	11:45 AM	Jane Doe
+ Destination	11:00 AM	11:45 AM	Driver Name
+ Destination	11:00 AM	11:45 AM	Driver Name

## View Route Detail

The application interface consists of two main screens:

- Available Routes Screen:** Shows a table with columns: Routes, Arrival, Departure, and Driver. Each row contains a plus icon and a destination name. Buttons for Back and Edit are at the bottom.
- Route Detail Screen:** Shows a table with columns: Routes, Arrival, Departure, and Driver. A specific row is selected. Below the table, there are sections for Route Info, Driver Info, and Addition Info. At the bottom, there is a table representation of the data with columns: Routes, Arrival, Departure, and Driver.

**Easier Representation of the Table**

Routes	Arrival	Departure	Driver
+ Destination	11:00 AM	11:45 AM	Driver Name
+ Destination	11:00 AM	11:45 AM	Driver Name
+ Destination	11:00 AM	11:45 AM	Driver Name
+ Destination	11:00 AM	11:45 AM	Driver Name
+ Destination	11:00 AM	11:45 AM	Driver Name
+ Destination	11:00 AM	11:45 AM	Driver Name
+ Destination	11:00 AM	11:45 AM	Driver Name

# UI Drawings for Maintenance - Mechanic

## Maintenanace - Create Compatable Part - Desktop

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Create Part      View Part      Update Part      Delete Part

### Create Compatible Parts

Part Name      Description of Part      Part Number(Unique ID)

Compatibility Information

Vehicle(s)

Select vehicle(s) that are compatible With part

Vehicle Model...

Vehicle Model...

Vehicle Model...

Notes

Text box

Create      Cancel

Image of part (Optional)  
100 x 100

PopUps

**CONFIRM**  
Confirm to add part to list?  
Confirm      Cancel

**Saved**  
Part is saved!

**Cancel**  
Process canceled!

© 2023 Name

## Maintenance - Complete Vehicle Workorder - Desktop

Window

Fleet Industries' LOGO

Profil Account Name

### Complete Work Order

Product/s Used:

Category	Product	Quantity	Units
Oil / Lubricants	Castrol GTX 5W-20	12	Quarts
Tires	Cooper Discover ATX-3	4	Tires

Category: Oil / Lubricants

Product: Castrol GTX 5W-20

Quantity: 12 Unit: Quarts

Update Product List

Service Completed?  Yes  No

Service Type: Oil Change

Request Description: Change Oil and oil filter with OEM products

Maintenance Notes: This take 5 quarts of oil. The tires are replaced.

© 2023 Name

Confirm Completion

Above options are filled with existing information so user knows what they are changing

## Maintenance - Create Vehicle Workorder - Desktop

Window

Fleet Industries' LOGO

Profil Account Name

### Create Work Order Request

High Priority?  Yes  No

Service Type: Oil Change

Request Description: Change Oil and oil filter with OEM products

Requested Products:

Category: Oil / Lubricants

Product: Castrol GTX 5W-20

Quantity: 12 Unit: Quarts

© 2023 Name Above options are filled with existing information so user knows what they are changing

Above options are filled with existing information so user knows what they are changing

## Maintenance - Delete Compatable Parts - Desktop

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Create Part      View Part      Update Part      Delete Part

### Delete Compatible Part

Compatible parts list

	Select part from list	Description	Vehicle
<input checked="" type="radio"/>	Part name	Part Description	Model
<input type="radio"/>	Part name	Part Description	Model
<input type="radio"/>	Part name	Part Description	Model
<input type="radio"/>	Part name	Part Description	Model

**DELETE**

**Warning**  
Are you sure you want to save changes?  
Deletions can not be undone.  
**Cancel**      **Save**

**DELETED**  
Part has been deleted!  
**OK**

**ERROR**  
Unable to delete  
**OK**

© 2023 Name

## Maintenance - Edit Vehicle Details - Desktop

Window

Fleet Industries' LOGO

Profil Account Name

Vehicle Details Work Orders Maintenance Log

**Vehicle Details**

**(Vehicle) Information**

Make	Ford
Model	F-150
Trim	XLT
Year	2004
Engine	5.4-Liter Triton V8
VIN	1FTPW14584KA65763

275 x 261

Choose File C:\Pictures\ford.jpg

Cancel Save

© 2023 Name

## Maintenance - Edit Vehicle Workorder - Desktop

Window

Fleet Industries' LOGO

Profil Account Name

### Update/Append Work Order Request

High Priority?  Yes  No

Service Type: Oil Change

Request Description: Change Oil and oil filter with OEM products

Requested Products:

Category: Oil / Lubricants

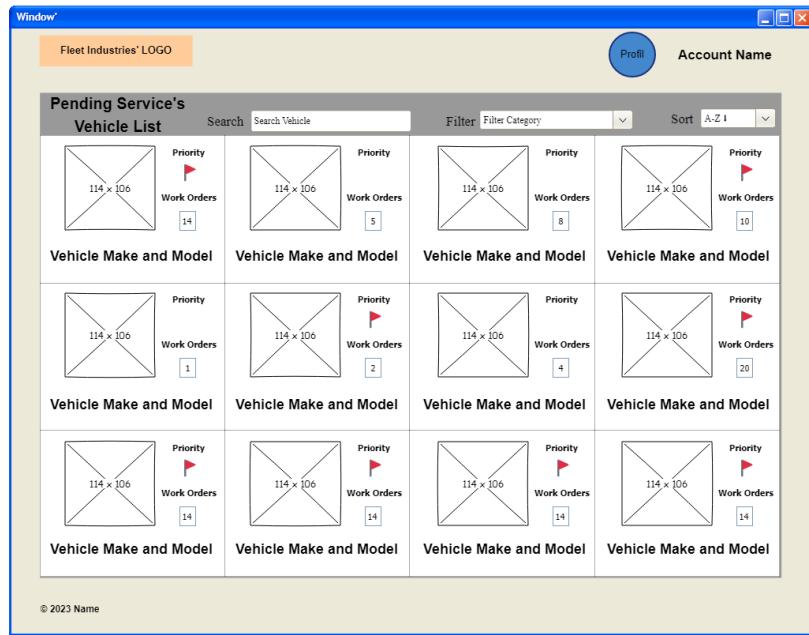
Product: Castrol GTX 5W-20

Quantity: 12 Unit: Quarts

Confirm Completion

© 2023 Name Above options are filled with existing information so user knows what they are changing

## Maintenance - Pending Services - Desktop



## Maintenance - Pending Vehicle Workorders - Desktop

The screenshot displays a desktop application window titled "Maintenance - Pending Vehicle Workorders - Desktop". The window has a blue header bar with standard window controls (minimize, maximize, close) and a "Window" title. In the top right corner, there is a circular profile icon labeled "Profil" and the text "Account Name". The main content area is titled "Work Orders" and contains a sub-section titled "Pending (Vehicle/s) Work Orders". At the top of this section are three search/filter controls: "Search" (with a placeholder "Search Service Type" and a magnifying glass icon), "Filter" (with a dropdown menu set to "Filter Category"), and "Sort" (with a dropdown menu set to "A-Z"). Below these controls is a table listing eight work orders. The table has columns: "High Priority" (indicated by a red triangle icon), "Service Type", "Description", "Update Work Order" (containing blue "Update" and "Complete" links), and "Complete Work Order" (also containing blue "Update" and "Complete" links). The rows of data are as follows:

High Priority	Service Type	Description	Update Work Order	Complete Work Order
▶	Oil Change	Change Oil and oil filter with OEM products	<a href="#">Update</a>	<a href="#">Complete</a>
▶	Cell content 2	Content	<a href="#">Update</a>	<a href="#">Complete</a>
	Cell content 3	Content	<a href="#">Update</a>	<a href="#">Complete</a>
	Cell content 4	Content	<a href="#">Update</a>	<a href="#">Complete</a>
▶	Cell content 5	Content	<a href="#">Update</a>	<a href="#">Complete</a>
▶	Cell content 6	Content	<a href="#">Update</a>	<a href="#">Complete</a>
	Cell content 7	Content	<a href="#">Update</a>	<a href="#">Complete</a>
▶	Cell content 8	Content	<a href="#">Update</a>	<a href="#">Complete</a>

At the bottom left of the content area is a "Close" button. The footer of the window contains the copyright notice "© 2023 Name".

## Maintenance - Update Compatable Parts - Desktop

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Create Part      View Part      Update Part      Delete Part

### Update Compatible Parts

Compatible Parts List

Parts	Select part from list	Description	Vehicle
<input checked="" type="radio"/>	Part name	Part Description	Model
<input type="radio"/>	Part name	Part Description	Model
<input type="radio"/>	Part name	Part Description	Model

Image of part (Optional)  
100 x 100  
PopUps

**CONFIRM**  
Confirm to update part to list?

Selected Part Name  
part name      part description      part ID #

Vehicle(s)

<input type="checkbox"/>	Select vehicle(s) that are compatible With part
<input checked="" type="checkbox"/>	Vehicle Model...
<input type="checkbox"/>	Vehicle Model...
<input checked="" type="checkbox"/>	Vehicle Model...

**Updated**     
Part Updated

**Cancel**     
Process canceled!

Update      Cancel

© 2023 Name

## Maintenance - Vehicle Details - Desktop

Window

Fleet Industries' LOGO

Profil Account Name

Vehicle Details Work Orders Maintenance Log

**Vehicle Details**

**(Vehicle) Information**

Make	Ford
Model	F-150
Trim	XLT
Year	2004
Engine	5.4-Liter Triton V8
VIN	1FTPW14584KA65763

275 x 261

Back Edit

© 2023 Name

## Maintenance - Vehicle Workorder Log - desktop

Details

Fleet Industries' LOGO

Profil Account Name

Vehicle Details Work Orders Maintenance Log

### Maintenance Log

#### (Vehicle)'s Completed Service Log

Date Completed	Service Type	Description	Products Category	Details	Update
2023-10-4	Oil Change	Change Oil and oil filter with OEM products	Oil/Lubricants	<a href="#">More</a>	<a href="#">Update</a>
2022-5-20	Tire Change	Replace the tires with OEM replacement	Tires	<a href="#">More</a>	<a href="#">Update</a>

Search

Filter  Sort

[Close](#)

© 2023 Name

## Maintenenace - View Compatable Parts - Desktop

Chrome

Fleet Industries' Logo    Home > Products > Pencil    Login    Register Now!

Create Part    View Part    Update Part    Delete Part

**View Compatible parts**    Search: Vehicle    Filter: Compatible Parts

**Selected Vehicle Model**

	Select part from list	Description
<input checked="" type="radio"/>	Part name	Part Description
<input type="radio"/>	Part name	Part Description
<input type="radio"/>	Part name	Part Description
<input type="radio"/>	Part name	Part Description
<input type="radio"/>	Part name	Part Description
<input type="radio"/>	Part name	Part Description

**Actions:** Create    Update    Delete

© 2023 Name

## UI\_MMTC\_04\_02

Window

Fleet Industries' LOGO

Profil Account Name

### Complete Work Order

Product/s Used:

Category	Product	Quantity	Units
Oil / Lubricants	Castrol GTX 5W-20	12	Quarts
Tires	Cooper Discover ATX-3	4	Tires

Category: Oil / Lubricants

Product: Castrol GTX 5W-20

Quantity: 12 Unit: Quarts

Update Product List

Service Completed?  Yes  No

Service Type: Oil Change

Request Description: Change Oil and oil filter with OEM products

Maintenance Notes: This take 5 quarts of oil. The tires are replaced.

© 2023 Name

Confirm Completion

Above options are filled with existing information so user knows what they are changing

## UI\_MMTC\_04\_03

Details

Fleet Industries' LOGO

Profil Account Name

Vehicle Details Work Orders Maintenance Log

### Maintenance Log

#### (Vehicle)'s Completed Service Log

High Priority	Service Type	Description	Category	Products Used	Quantity	Units
🚩	Oil Change	Change Oil and oil filter with OEM products	Oil/Lubricants	Castrol GTX 5W-20	12	Quarts
	Tire Change	Replace the tires with OEM replacement	Tires	Cooper Discover ATX-3	4	Tires

Search

Filter

Sort

© 2023 Name

## UI\_MMTC\_06\_01

Window

Fleet Industries' LOGO

Profil Account Name

### Complete Work Order

Product/s Used:

Category	Product	Quantity	Units
Oil / Lubricants	Castrol GTX 5W-20	12	Quarts
Tires	Cooper Discover ATX-3	4	Tires

Category: Oil / Lubricants

Product: Castrol GTX 5W-20

Quantity: 12 Unit: Quarts

Update Product List

Service Completed?  Yes  No

Service Type: Oil Change

Request Description: Change Oil and oil filter with OEM products

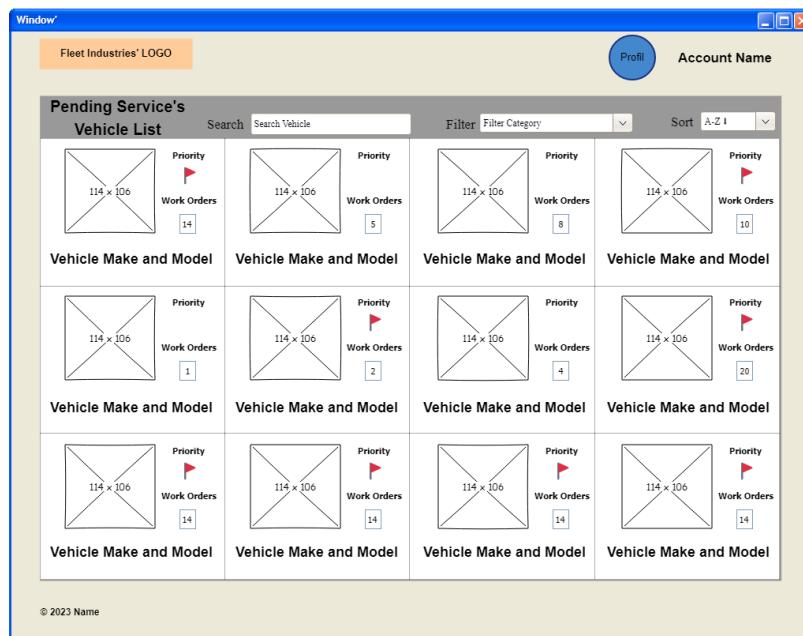
Maintenance Notes: This take 5 quarts of oil. The tires are replaced.

© 2023 Name

Confirm Completion

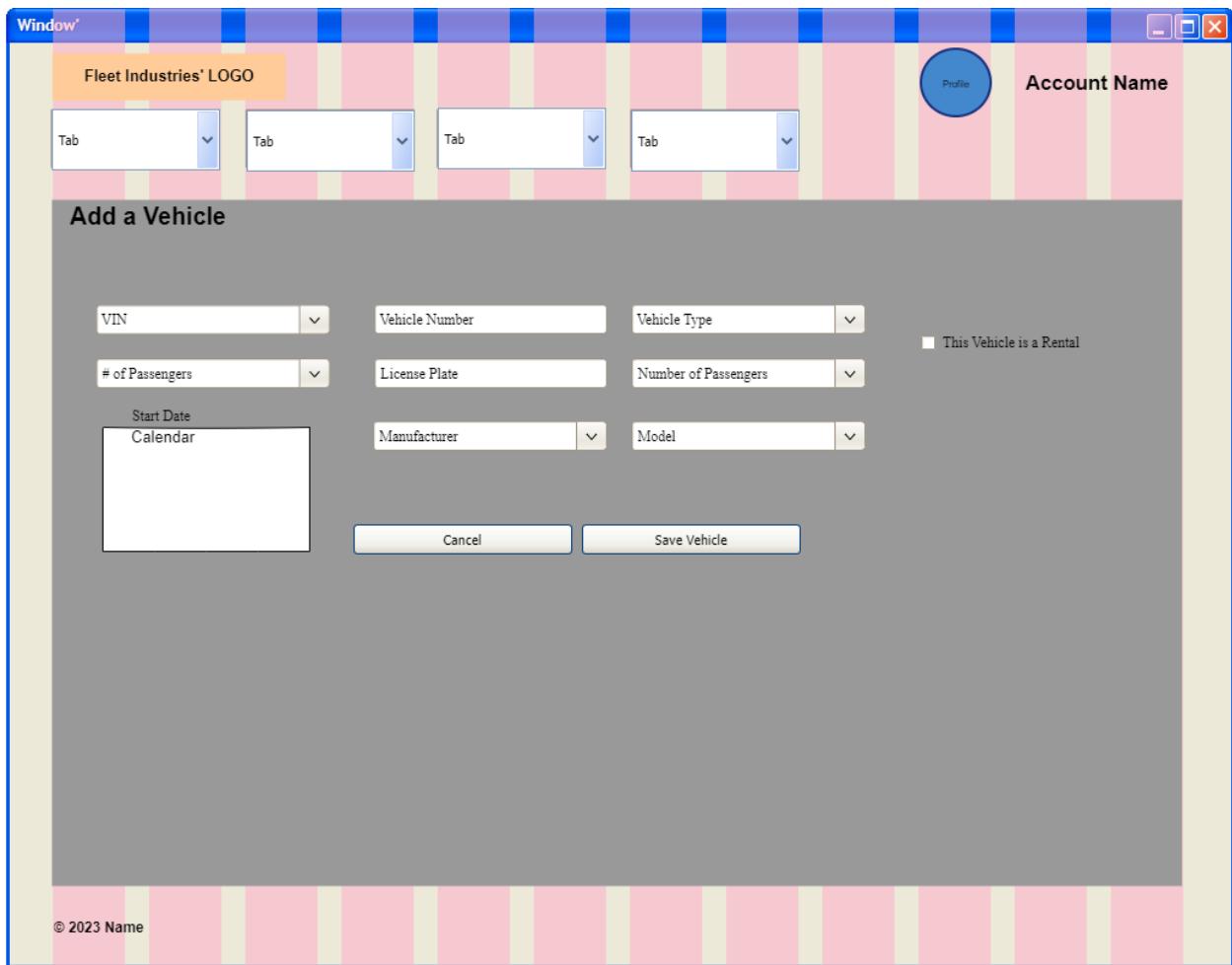
Above options are filled with existing information so user knows what they are changing

## UI\_MMTC\_07\_01



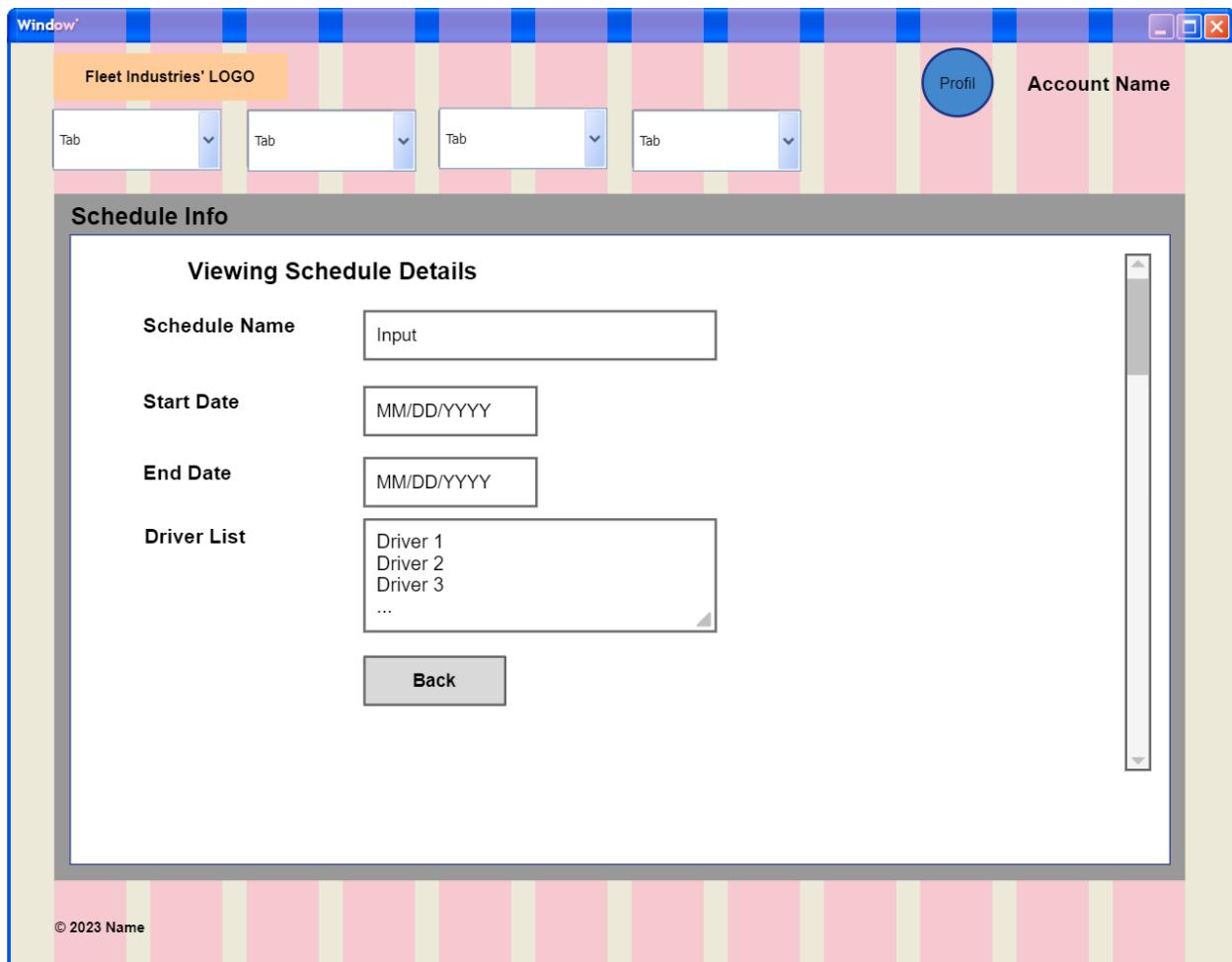
# UI Drawings for Operations Manager

## Operations Manager - Create Vehicle - Desktop

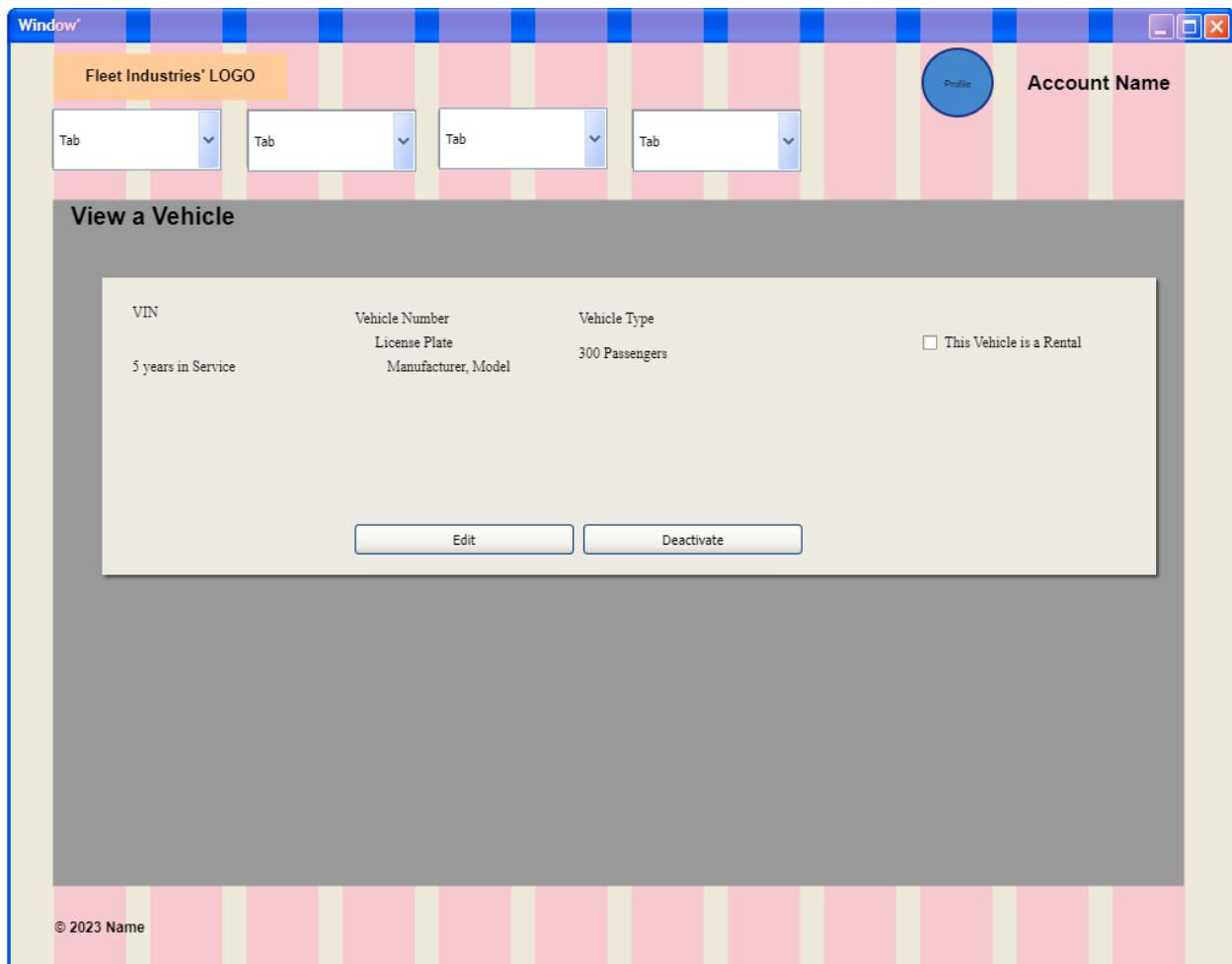


The image shows a wireframe diagram of a desktop application window titled "Operations Manager - Create Vehicle". The window has a standard OS X-style title bar with "Window", a close button, and a minimize button. In the top right corner, there is a blue circular profile icon and the text "Account Name". The main content area is titled "Add a Vehicle". It contains several input fields: "VIN" dropdown, "Vehicle Number" dropdown, "Vehicle Type" dropdown, "Start Date" calendar, "# of Passengers" dropdown, "License Plate" dropdown, "Number of Passengers" dropdown, "Manufacturer" dropdown, and "Model" dropdown. Below these fields are two buttons: "Cancel" and "Save Vehicle". At the bottom left of the window, there is a copyright notice: "© 2023 Name".

## Operations Manager - View Schedule Detail - Desktop



## Operations Manager - View Vehicle - Desktop



## Operations Manager Activate Bus Route - Desktop

The screenshot shows a desktop application window titled "Operations Manager Activate Bus Route - Desktop". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a "Window" tab. In the top right corner, there is a blue circular button labeled "Profil" and an "Account Name" field. The main content area is titled "Update Route". It contains several input fields: "Route Name:" with a text box "txtRouteName", "Route Start Time:" with dropdowns "cboStartHour" and "cboStartMinute", "Route Cycle:" with a text box "txtRouteCycle", and "Route End Time:" with dropdowns "cboEndHour" and "cboEndMinute". Below these is a checkbox "Active" and a "Days of Service" section with checkboxes for Sunday through Saturday. A "Route Stops" section follows, featuring a table with columns "StopNumber", "StartOffset", "Address", "ZIPCode", "Latitude", and "Longitude". Buttons "Add Stop" and "Delete Selected Stop" are located next to the table. At the bottom are "Update" and "Cancel" buttons. The footer of the window includes the text "© 2023 Name".

## Operations Manager Activate Stop - Desktop

The screenshot shows a desktop application window titled "Operations Manager Activate Stop - Desktop". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a logo for "Fleet Industries' LOGO". On the right side of the header, there is a circular button labeled "Profil" and the text "Account Name". The main content area is titled "Update Stop" and contains the following fields:

Street Address	txtStreetAddress
ZIP Code	txtZIPCode
Latitude	txtLatitude
Longitude	txtLongitude
<input type="checkbox"/> Active	

At the bottom of the window are two buttons: "Update" and "Cancel". In the bottom-left corner of the main content area, there is a small copyright notice: "© 2023 Name".

## Operations Manager Add or Edit Driver

The screenshot shows a software application window titled "Operations Manager Add or Edit Driver". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a "Profile" button. The main area is titled "Add a new Driver". On the left, there are three dropdown menus: "Employee Name", "License Class", and "Start Date" (with a "Calendar" button). In the center, there is a "Driver Name" section containing fields for "Name" (123 Main St, Suite 6, Fake City, CA, 12345), "Phone" (555-555-5555), "Email" (driver.name@email.com), and "Other Roles" (Maintenance, Fleet Administrator). At the bottom of this section are "Cancel" and "Save Driver" buttons. To the right, there are three boxes labeled "Schedule 1", "Schedule 2", and "Schedule 3", each with a red minus sign button. Below these is an "Add Schedule" button. The footer contains the copyright notice "© 2023 Name".

Fleet Industries' LOGO

Account Name

Add a new Driver

Employee Name

License Class

Start Date  
Calendar

Driver Name  
123 Main St  
Suite 6  
Fake City, CA, 12345  
555-555-5555  
driver.name@email.com  
Other Roles:  
Maintenance  
Fleet Administrator

Schedule 1

Schedule 2

Schedule 3

Add Schedule

Cancel

Save Driver

© 2023 Name

## Operations Manager Create New Bus Route - Desktop

The screenshot shows a desktop application window titled "Operations Manager Create New Bus Route - Desktop". The window has a standard Windows-style title bar with "Window", "Minimize", "Maximize", and "Close" buttons. In the top left corner, there is a placeholder for "Fleet Industries' LOGO". On the right side of the title bar, there is a blue circular button labeled "Profil" and an "Account Name" field. Below the title bar, there are four tabs labeled "Tab", "Tab", "Tab", and "Tab".

The main content area is titled "Create Route". It contains the following fields:

- Route Name:
- Route Start Time:  :
- Route Cycle:
- Route End Time:  :

Below these fields is a checkbox labeled "Active" and a "Days of Service" section containing checkboxes for Sunday through Saturday.

The "Route Stops" section features a table with columns: StopNumber, StartOffset, Address, ZIPCode, Latitude, and Longitude. There are "Add Stop" and "Delete Selected Stop" buttons next to the table.

At the bottom of the window are two buttons: "Create" and "Cancel".

In the bottom left corner of the window frame, there is a copyright notice: "© 2023 Name".

## Operations Manager Create Schedule - Desktop

The screenshot shows a desktop application window titled "Operations Manager Create Schedule - Desktop". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a logo for "Fleet Industries' LOGO". On the right side of the header, there is a blue circular button labeled "Profil" and a text field labeled "Account Name". The main content area is titled "Create Schedule". It contains several input fields: "Schedule Name" with a text box, "Schedule Start Date" and "Schedule End Date" both with MM/DD/YYYY date pickers, and a "Driver Selection" section. The "Driver Selection" section includes a "Driver List" dropdown menu showing four names: John Smith, Steve Miller, Paul Samson, and Doug Appleton. Below the dropdown are "Add Driver" and "Remove" buttons. At the bottom of the window is a "Create" button. The footer of the window displays the copyright notice "© 2023 Name".

Schedule Name

Schedule Start Date  MM/DD/YYYY

Schedule End Date  MM/DD/YYYY

Driver Selection

Driver List  Add Driver Remove

John Smith  
Steve Miller  
Paul Samson  
Doug Appleton

Create

© 2023 Name

## Operations Manager Create Stop - Desktop

The screenshot shows a desktop application window titled "Operations Manager Create Stop - Desktop". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a "Window" tab. In the top left corner, there is a placeholder for "Fleet Industries' LOGO". On the right side of the header, there is a blue circular button labeled "Profil" and a text field labeled "Account Name". The main content area is titled "Create Stop" and contains the following fields:

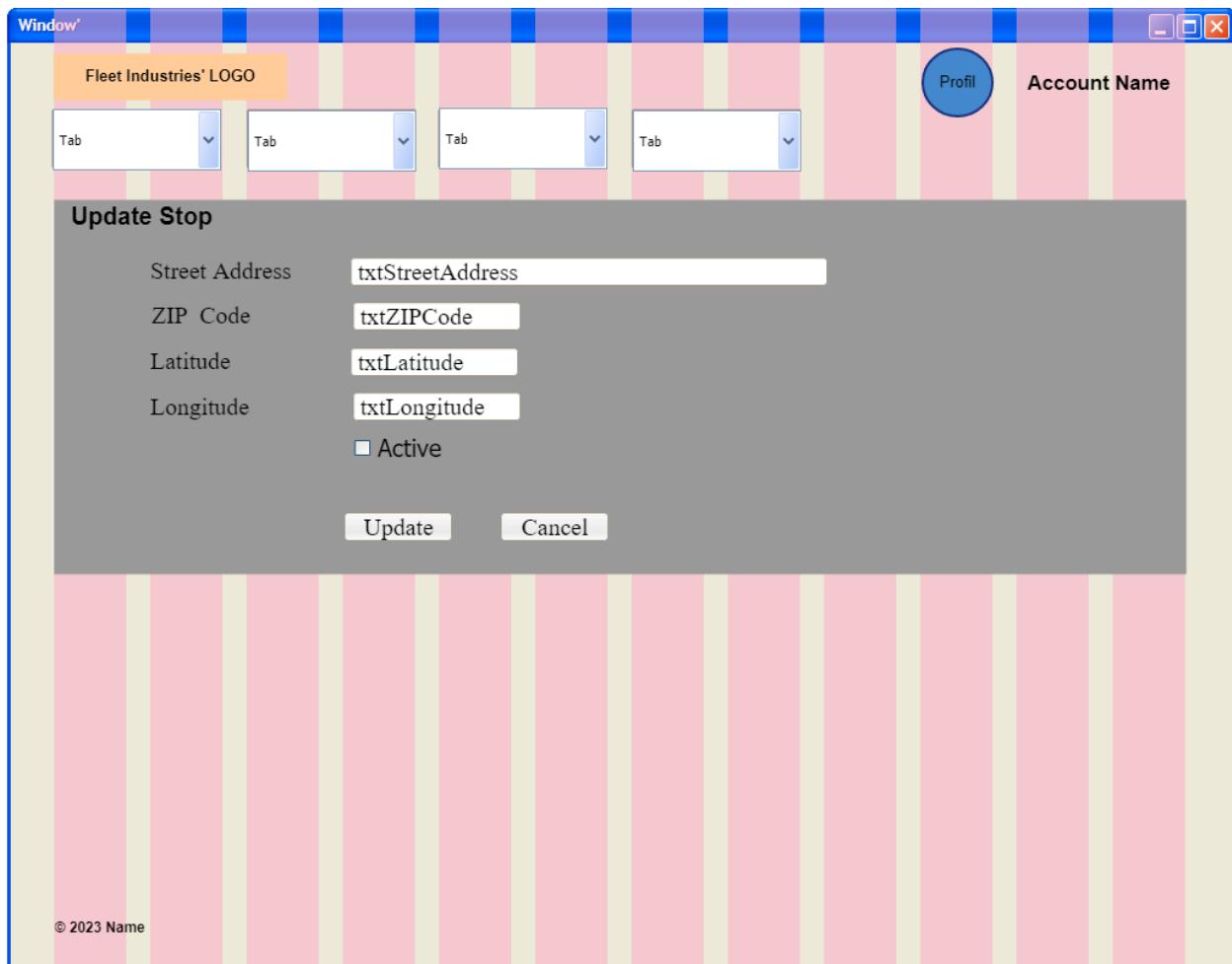
Street Address	txtStreetAddress
ZIP Code	txtZIPCode
Latitude	txtLatitude
Longitude	txtLongitude
<input type="checkbox"/> Active	

At the bottom of the window are two buttons: "Create" and "Cancel". A copyright notice "© 2023 Name" is visible at the bottom left of the window.

## Operations Manager Deactivate Bus Route - Desktop

The screenshot shows a desktop application window titled "Operations Manager Deactivate Bus Route - Desktop". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a "Window" tab. In the top right corner, there is a blue circular button labeled "Profil" and an "Account Name" field. The main content area is titled "Update Route". It contains several input fields: "Route Name:" with a text box "txtRouteName", "Route Start Time:" with a dropdown "cboStartHour" followed by a colon and another dropdown "cboStartMinute", "Route Cycle:" with a text box "txtRouteCycle", and "Route End Time:" with a dropdown "cboEndHour" followed by a colon and another dropdown "cboEndMinute". Below these fields is a checkbox "Active" and a "Days of Service" section containing checkboxes for Sunday through Saturday. A "Route Stops" section follows, featuring a table with columns: StopNumber, StartOffset, Address, ZIPCode, Latitude, and Longitude. The table has six rows. To the right of the table are two buttons: "Add Stop" and "Delete Selected Stop". At the bottom of the window are two buttons: "Update" and "Cancel". The footer of the window contains the text "© 2023 Name".

## Operations Manager Deactivate Stop - Desktop



## Operations Manager List Bus Routes - Desktop

The screenshot shows a desktop application window titled "Operations Manager List Bus Routes - Desktop". The window has a blue header bar with standard window controls (minimize, maximize, close) on the right. In the top left corner, there is a placeholder for "Fleet Industries' LOGO". On the right side of the header, there is a circular button labeled "Profil" and the text "Account Name". Below the header is a search bar with the title "Route List/Search". The search bar contains fields for "Name" (Text box), "Start Time" (Text box), "End Time" (Text box), "Day of Week" (Combo Box), "ZIP" (Text box), and buttons for "Search" and "Clear". Below the search bar is a table with columns: "Name", "Start Time", "End Time", "Days Of Week", "ZIP", an empty column, and an empty column. Each row in the table contains a "Content" entry in the first column and a "Cell Content 1" entry in the second column. The last two columns contain "Content" entries. Each row also includes a "Details" link and an "Edit" link. At the bottom of the search area is a "Back" button. The footer of the window displays the copyright notice "© 2023 Name".

Name	Start Time	End Time	Days Of Week	ZIP		
Content	Cell Content 1	Content	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>

[Back](#)

© 2023 Name

## Operations Manager Stop List - Desktop

The screenshot shows a desktop application window titled "Operations Manager Stop List - Desktop". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) on the right. In the top left corner, there is a placeholder for "Fleet Industries' LOGO". On the right side of the header, there is a blue circular button labeled "Profil" and a text field labeled "Account Name". The main content area is titled "Stop List/Search" and contains several input fields: "Address" (with a "Text box"), "ZIP" (with a "Text box"), "Start Time" (with a "Text box"), and "End Time" (with a "Text box"). Below these are two buttons: "Search" and "Clear". The main body of the window displays a table with columns: "Address", "ZIP", "Latitude", "Longitude", "Details", and "Edit". Each row in the table contains a "Content" entry in all columns except the first two, which have "Cell Content 1". There are also "Details" and "Edit" links in the last two columns. At the bottom left of the content area is a "Back" button. The footer of the window contains the copyright notice "© 2023 Name".

Address	ZIP	Latitude	Longitude	Details	Edit
Content	Cell Content 1	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>
Content	Cell Content 1	Content	Content	<a href="#">Details</a>	<a href="#">Edit</a>

## Operations Manager Update Bus Route - Desktop

The screenshot shows a Windows-style application window titled "Operations Manager Update Bus Route - Desktop". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a "Window" tab. In the top left corner, there is a placeholder for "Fleet Industries' LOGO". On the right side of the header, there is a circular button labeled "Profil" and a text field labeled "Account Name". The main content area is titled "Update Route". It contains several input fields: "Route Name:" with a text box "txtRouteName", "Route Start Time:" with a dropdown "cboStartHour" followed by a colon and another dropdown "cboStartMinute", "Route Cycle:" with a text box "txtRouteCycle", and "Route End Time:" with a dropdown "cboEndHour" followed by a colon and another dropdown "cboEndMinute". Below these fields is a checkbox "Active" and a "Days of Service" section containing checkboxes for Sunday through Saturday. A "Route Stops" section follows, featuring a table with columns: StopNumber, StartOffset, Address, ZIPCode, Latitude, and Longitude. The table has six rows. To the right of the table are two buttons: "Add Stop" and "Delete Selected Stop". At the bottom of the window are two buttons: "Update" and "Cancel". The footer of the window contains the copyright notice "© 2023 Name".

## Operations Manager Update Schedule - Desktop

The screenshot shows a desktop application window titled "Operations Manager Update Schedule - Desktop". The window has a blue header bar with standard window controls (Minimize, Maximize, Close) and a "Window" tab. In the top right corner, there is a blue circular button labeled "Profil" and an "Account Name" field. The main content area is titled "Update Schedule". It contains several input fields: "Schedule Name" (text box), "Schedule Start Date" (MM/DD/YYYY), "Schedule End Date" (MM/DD/YYYY), and a "Driver Selection" section. The "Driver Selection" section includes a dropdown menu labeled "Driver List" with a downward arrow, and buttons for "Add Driver" and "Remove". A list box displays four driver names: John Smith, Steve Miller, Paul Samson, and Doug Appleton. At the bottom of the window is a large "Update" button.

Fleet Industries' LOGO

Tab Tab Tab Tab

Profil Account Name

**Update Schedule**

Schedule Name

Schedule Start Date  MM/DD/YYYY

Schedule End Date  MM/DD/YYYY

Driver Selection

Driver List  Add Driver Remove

John Smith  
 Steve Miller  
 Paul Samson  
 Doug Appleton

Update

© 2023 Name

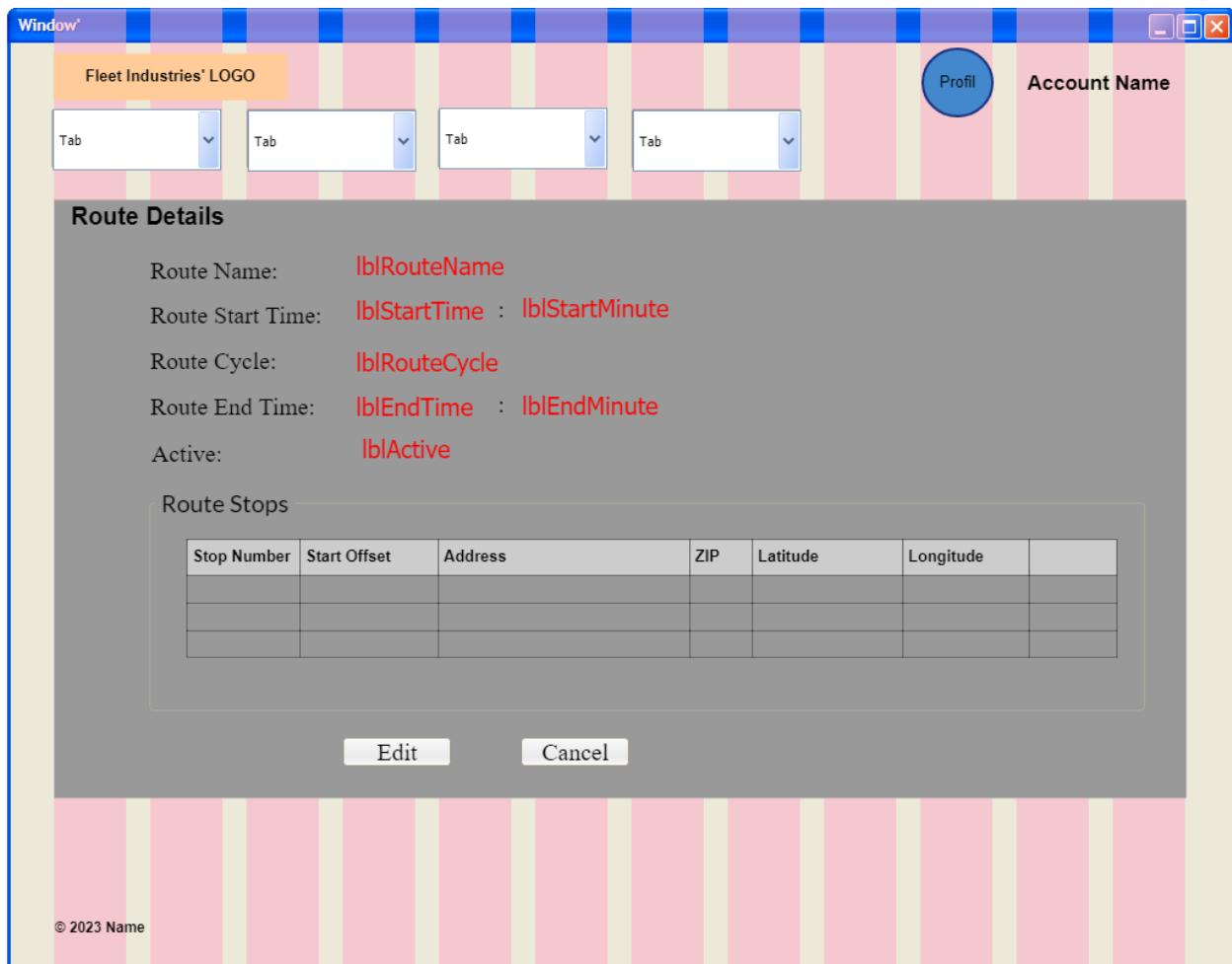
## Operations Manager Update Stop - Desktop

The screenshot shows a desktop application window titled "Operations Manager Update Stop - Desktop". The window has a blue header bar with standard window controls (minimize, maximize, close) and a logo for "Fleet Industries' LOGO". On the right side of the header, there is a blue circular button labeled "Profil" and a text field labeled "Account Name". The main content area is titled "Update Stop" and contains the following fields:

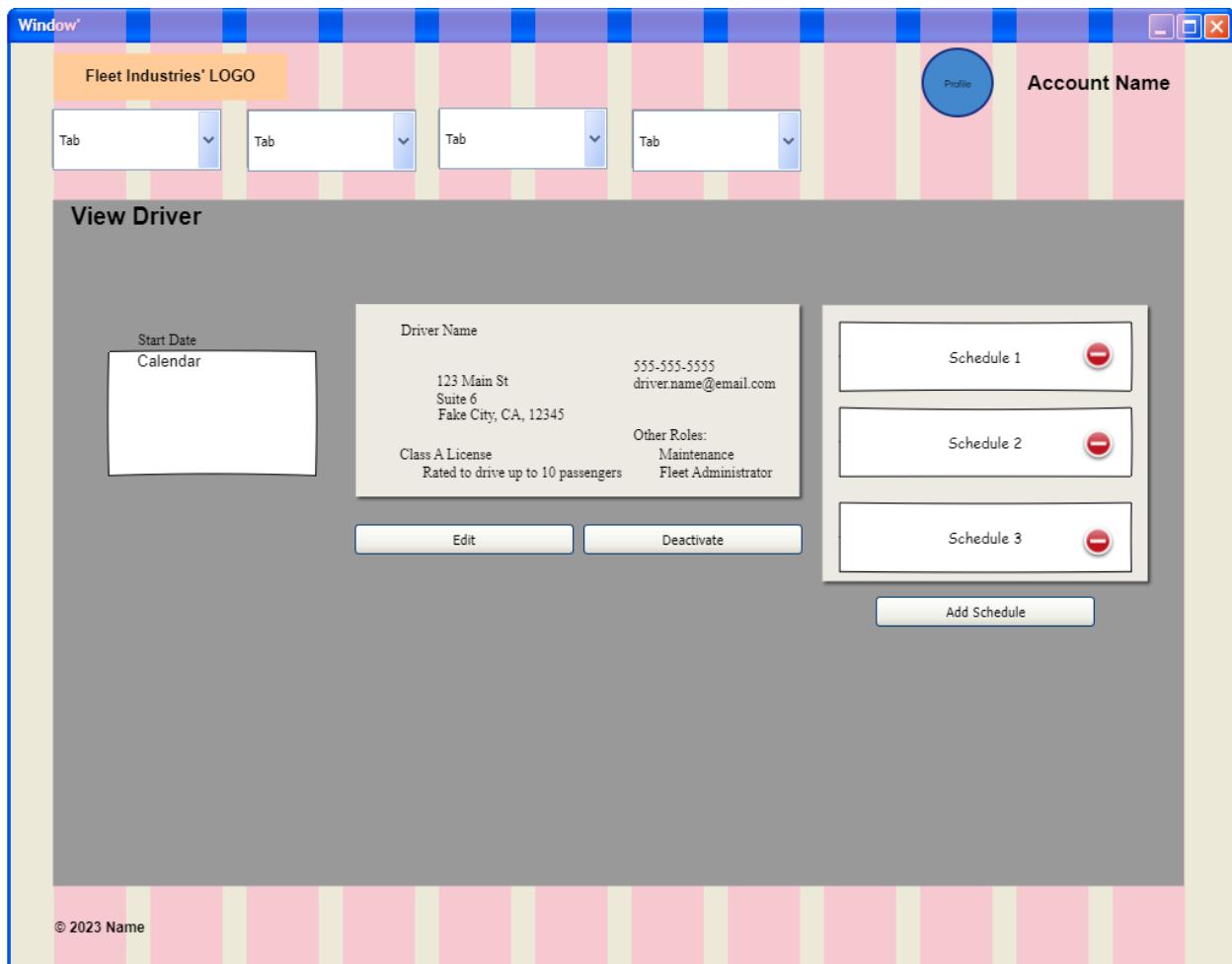
Street Address	txtStreetAddress
ZIP Code	txtZIPCode
Latitude	txtLatitude
Longitude	txtLongitude
<input type="checkbox"/> Active	

At the bottom of the window are two buttons: "Update" and "Cancel". In the bottom-left corner of the main content area, there is a small copyright notice: "© 2023 Name".

## Operations Manager View Bus Route Details - Desktop



## Operations Manager View or Deactivate Driver



## Operations Manager View Schedule List - Desktop

The screenshot shows a desktop application window titled "Operations Manager View Schedule List - Desktop". The window has a standard Windows-style title bar with icons for minimizing, maximizing, and closing. In the top right corner of the title bar, there is a blue circular button labeled "Profil". The main content area is divided into several sections:

- Fleet Industries' LOGO:** Located in the top left corner of the main content area.
- Account Name:** Located in the top right corner of the main content area.
- Schedule Lookup:** A header section containing the text "Schedule Lookup".
- Schedule List:** A sub-section containing the text "Schedule List".
- Search Schedule List:** A search bar with a placeholder "Text box" and a magnifying glass icon.
- Filter:** A dropdown menu labeled "Last Name".
- Data Table:** A table listing schedule details. The columns are: Schedule ID, Schedule Name, Start Date, End Date, and Admin. The rows contain the following data:

Schedule ID	Schedule Name	Start Date	End Date	Admin
154	Celebration Schedule	09/14/2023	09/30/2023	<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 2	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 3	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 4	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 5	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 6	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 7	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
Cell content 8	Content	Content		<a href="#">Edit</a> <a href="#">Remove</a>
- Back:** A button labeled "Back" located at the bottom left of the main content area.
- Footer:** A footer section at the bottom left containing the text "© 2023 Name".

# UI Drawings for Outside Contact

## Add Service Bid

The screenshot shows a Windows application window titled "Add Special Service Order Bid". The window has a blue header bar with standard window controls (minimize, maximize, close) and a "Window" tab. In the top right corner, there is a blue circular button labeled "Profil" and the text "Account Name". The main content area is a dark grey panel with white text and input fields. The fields include:

- Bid\_ID: #020502
- High Priority?:  Yes  No
- Vehicle Description: Yellow #7 bus
- Bid Description: Vehicle hit a deer going south on I-380
- Vendor: Mike's
- Estimate: Yellow #7 bus
- Service Date: 12/14/23

At the bottom left of the main panel, there is a "Create" button. At the very bottom left of the window, outside the main panel, is the text "© 2023 Name".

## Add Special Inspection

Window' X

Fleet Industries' LOGO Profil Account Name

**Add Speical Inspection**

Inspection ID	#234
High Priority?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Vehicle Description	Yellow #7 bus
Inspection Description	Vehicle hit a deer going south on I-380
Mechanic	Mike <span style="float: right;">▼</span>
Appraisal Date	12/14/23

**Create**

© 2023 Name

## Add Special Service Order

Window

Fleet Industries' LOGO

Profil Account Name

### Create Special Service Order Request

Special Service order # #2495

High Priority?  Yes  No

Vehicle Description Yellow #7 bus

Event Description Vehicle hit a deer going south on I-380

View Appraisals Previous Add

View Bids Mike's Auto Body Add

View Work Orders Mike's Auto Body Add

Create

© 2023 Name Above options are filled with existing information so user knows what they are changing

The screenshot shows a software interface for creating a special service order. At the top, there's a header bar with a logo for 'Fleet Industries' LOGO', a 'Profil' button, and an 'Account Name' field. Below the header, the main title is 'Create Special Service Order Request'. The form contains several input fields and dropdown menus. One dropdown menu for 'View Appraisals' has 'Previous' selected, and there's an 'Add' button next to it. Another dropdown for 'View Bids' has 'Mike's Auto Body' selected, with an 'Add' button. A third dropdown for 'View Work Orders' also has 'Mike's Auto Body' selected, with an 'Add' button. At the bottom left, there's a copyright notice: '© 2023 Name'. At the bottom right, a note says: 'Above options are filled with existing information so user knows what they are changing'.

## Add Special Work Order

Window

Fleet Industries' LOGO

Profil Account Name

Add Special work Order

Special Work Order ID #29324

Vehicle Description Yellow #7 bus

Event Description Alex will take the vehicle to Mike's at 7AM.

Vendor Mike's

Drop off Date 12/12/23

Pickup Date (est) 12/14/23

Create

© 2023 Name

The screenshot shows a software application window titled "Add Special Work Order". The window has a blue header bar with standard window controls (minimize, maximize, close) and a "Window" tab. In the top right corner, there is a blue circular button labeled "Profil" and a text field labeled "Account Name". The main content area is titled "Add Special work Order". It contains several input fields: "Special Work Order ID #29324", "Vehicle Description" (Yellow #7 bus), "Event Description" (Alex will take the vehicle to Mike's at 7AM.), "Vendor" (Mike's), "Drop off Date" (12/12/23), and "Pickup Date (est)" (12/14/23). At the bottom left is a "Create" button. The footer of the window says "© 2023 Name".

## Update or Deactivate Special Service Order

Window

Fleet Industries' LOGO

Profil Account Name

### Update Special Service Order Request

Special Service order # #2495

High Priority?  Yes  No

Vehicle Description Yellow #7 bus

Event Description Vehicle hit a deer going south on I-380

View Appraisals Previous Add

View Bids Mike's Auto Body Add

View Work Orders Mike's Auto Body Add

Update Deactivate

Above options are filled with existing information so user knows what they are changing

© 2023 Name

## Update Or Delete Service Bid

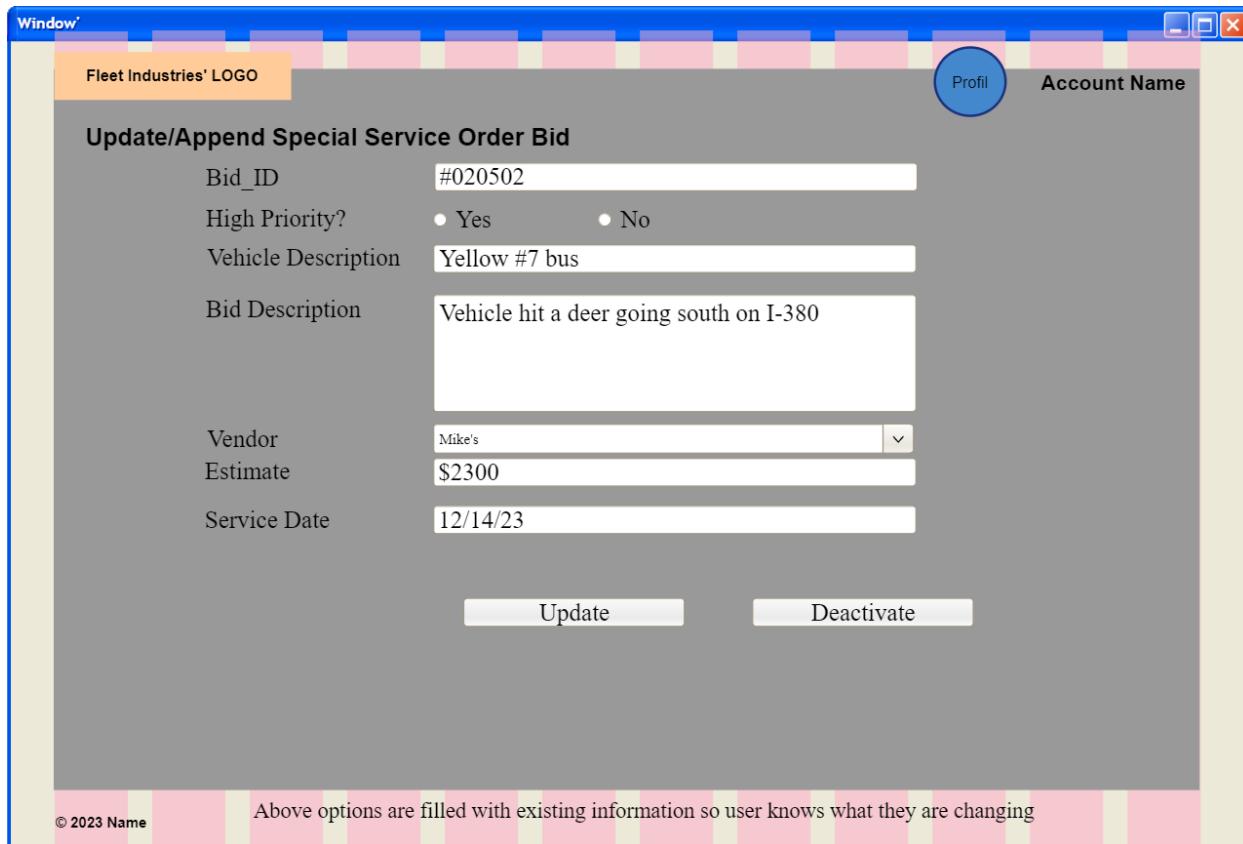
Window' X

Fleet Industries' LOGO Profil Account Name

**Update/Append Special Service Order Bid**

Bid_ID	#020502
High Priority?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Vehicle Description	Yellow #7 bus
Bid Description	Vehicle hit a deer going south on I-380
Vendor	Mike's
Estimate	\$2300
Service Date	12/14/23

© 2023 Name Above options are filled with existing information so user knows what they are changing



## Update or Delete Special Inspection

Window' X

Fleet Industries' LOGO Profil Account Name

**Update/Append Special Inspection**

Inspection ID	#234
High Priority?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Vehicle Description	Yellow #7 bus
Appraisal Description	Vehicle hit a deer going south on I-380
Mechanic	Mike <span style="float: right;">▼</span>
Appraisal Date	12/14/23

**Update**      **Deactivate**

Above options are filled with existing information so user knows what they are changing

© 2023 Name

## Update or Delete Speical Work Order

Window

Fleet Industries' LOGO

Profil Account Name

**Update/Append Outside Work Order**

Special Work Order ID #93552

Complete  Yes  No

Vehicle Description Yellow #7 bus

Event Description Alex will take the vehicle to Mike's at 7am

Vendor Mike's

Drop Off Date 12/12/23

Pick up date (est) 12/14/23

Update Deactivate

Above options are filled with existing information so user knows what they are changing

© 2023 Name

## View All Service Bid

The screenshot displays a software interface for managing service bids. At the top, there's a navigation bar with tabs for 'Vehicle Details', 'Work Orders', 'Maintenance Log', and 'Bids'. On the right side of the header, there are buttons for 'Profil' and 'Account Name'. Below the header, a section titled 'Work Orders' contains a sub-section 'Pending Bids'. This section includes search and filter tools: 'Search' (with a placeholder 'Search Service Type' and a magnifying glass icon), 'Filter' (with a dropdown menu 'Filter Category'), and 'Sort' (with a dropdown menu 'A-Z'). The main content area is a table with the following data:

High Priority	Bid_ID	Vehicle Description	Vendor	View Bid Order	Update Bid
▶	43	Bus #7	Mike's	<a href="#">View</a>	<a href="#">Update</a>
▶	44	Cell content 2	Content	<a href="#">View</a>	<a href="#">Update</a>
	45	Cell content 3	Content	<a href="#">View</a>	<a href="#">Update</a>
	46	Cell content 4	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	47	Cell content 5	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	48	Cell content 6	Content	<a href="#">View</a>	<a href="#">Update</a>
	49	Cell content 7	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	50	Cell content 8	Content	<a href="#">View</a>	<a href="#">Update</a>

At the bottom of the 'Pending Bids' section is a 'Close' button. The footer of the window contains the text '© 2023 Name'.

## View All Special Inspection

Fleet Industries' LOGO

Vehicle Details Work Orders Maintenance Log Special Service Profil Account Name

Work Orders

Completed Special Inspections

High Priority	Vehicle Description	Inspection Date	View Sp. Inspection	Update Sp. Inspection
▶	Bus #7	12/12/12	<a href="#">View</a>	<a href="#">Update</a>
▶	Cell content 2	Content	<a href="#">View</a>	<a href="#">Update</a>
	Cell content 3	Content	<a href="#">View</a>	<a href="#">Update</a>
	Cell content 4	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	Cell content 5	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	Cell content 6	Content	<a href="#">View</a>	<a href="#">Update</a>
	Cell content 7	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	Cell content 8	Content	<a href="#">View</a>	<a href="#">Update</a>

Close

© 2023 Name

## View All Special Service Order

The screenshot shows a software interface for managing service orders. At the top, there's a header bar with tabs for "Vehicle Details", "Work Orders", and "Maintenance Log". On the right side of the header, there's a blue circular icon labeled "Profil" and a text field labeled "Account Name". Below the header, the main area is titled "Work Orders" and contains a sub-section titled "Pending special service orders". This section includes search and filter tools: "Search" (with a placeholder "Search Service Type" and a magnifying glass icon), "Filter" (with a dropdown menu "Filter Category"), and "Sort" (with a dropdown menu "A-Z"). The main content is a table with the following columns: "High Priority", "Vehicle Description", "Description", "Update Service Order", and "Complete Service Order". The table contains 8 rows of data, each with a red warning icon in the first column. The "Vehicle Description" column lists various vehicles, and the "Description" column lists service details. The "Update Service Order" and "Complete Service Order" columns each contain two blue hyperlinks: "Update" and "Complete". At the bottom left of this section is a "Close" button.

High Priority	Vehicle Description	Description	Update Service Order	Complete Service Order
🚩	Bus #7	Body work after Deer accident	<a href="#">Update</a>	<a href="#">Complete</a>
🚩	Cell content 2	Content	<a href="#">Update</a>	<a href="#">Complete</a>
	Cell content 3	Content	<a href="#">Update</a>	<a href="#">Complete</a>
	Cell content 4	Content	<a href="#">Update</a>	<a href="#">Complete</a>
🚩	Cell content 5	Content	<a href="#">Update</a>	<a href="#">Complete</a>
🚩	Cell content 6	Content	<a href="#">Update</a>	<a href="#">Complete</a>
	Cell content 7	Content	<a href="#">Update</a>	<a href="#">Complete</a>
🚩	Cell content 8	Content	<a href="#">Update</a>	<a href="#">Complete</a>

© 2023 Name

## View All Special Work Order

The screenshot shows a software interface for managing work orders. At the top, there's a header bar with various buttons and a logo for "Fleet Industries' LOGO". To the right of the header is a blue circular button labeled "Profil" and the text "Account Name". Below the header, there are four main menu items: "Vehicle Details", "Work Orders", "Maintenance Log", and "Outside Work Orders". The "Outside Work Orders" item is highlighted.

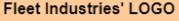
The main content area is titled "Outside Work Orders" and contains a sub-section titled "Pending Outside Work Order". This section includes a search bar with placeholder "Search Service Type" and a magnifying glass icon, a filter dropdown set to "Filter Category", and a sort dropdown set to "A-Z".

The data is presented in a table:

High Priority	Vehicle Description	Body Shop	View Work Order	Update Work Order
▶	Bus #7	Mike's	<a href="#">View</a>	<a href="#">Update</a>
▶	Cell content 2	Content	<a href="#">View</a>	<a href="#">Update</a>
	Cell content 3	Content	<a href="#">View</a>	<a href="#">Update</a>
	Cell content 4	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	Cell content 5	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	Cell content 6	Content	<a href="#">View</a>	<a href="#">Update</a>
	Cell content 7	Content	<a href="#">View</a>	<a href="#">Update</a>
▶	Cell content 8	Content	<a href="#">View</a>	<a href="#">Update</a>

At the bottom left of the content area is a "Close" button. The footer of the window contains the text "© 2023 Name".

## View Service Bid

Window'  Profil Account Name 

**View Special Service Order Bid**

Bid_ID	#020502
High Priority?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Vehicle Description	Yellow #7 bus
Bid Description	Vehicle hit a deer going south on I-380
Vendor	Mike's
Estimate	Yellow #7 bus
Service Date	12/14/23

© 2023 Name

## View Special Inspection

Window' X

Fleet Industries' LOGO Profil Account Name

**View Special Inspection**

Inspection ID	#234
High Priority?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Vehicle Description	Yellow #7 bus
Appraisal Description	Vehicle hit a deer going south on I-380
Mechanic	Mike
Appraisal Date	12/14/23

© 2023 Name

## View Special Service Order

Window' X

Fleet Industries' LOGO Profil Account Name

**View Special Service Order Request**

Special Service order # **#2495**

High Priority?  Yes  No

Vehicle Description **Yellow #7 bus**

Event Description **Vehicle hit a deer going south on I-380**

View Appraisals **Previous** Add

View Bids **Mike's Auto Body** Add

View Work Orders **Mike's Auto Body** Add

© 2023 Name Above options are filled with existing information so user knows what they are changing

## View Special Work Order

Window

Fleet Industries' LOGO

Profil Account Name

**View Outside Work Order**

Outside Work Order ID #932084

Vehicle Description Yellow #7 bus

Event Description Alex Will take the vehicle to Mike's at 7am

Vendor Mike's

Drop Off Date 12/12/23

Pickup Date (est) 12/14/23

© 2023 Name

Vehicle Description	Yellow #7 bus
Event Description	Alex Will take the vehicle to Mike's at 7am
Vendor	Mike's
Drop Off Date	12/12/23
Pickup Date (est)	12/14/23

# UI Drawings for Parent

## Add Bus Instructions - Web

The diagram illustrates the user interface for adding bus instructions. It features a top navigation bar with a 'Chrome' logo, a 'Fleet Industries' logo, and a breadcrumb trail 'Home > Users > Parent'. On the right side of the bar are 'Login' and 'Register Now!' buttons. Below the bar, there are four buttons labeled 'Home', 'Primary', 'Primary', and 'Primary'. The main content area is titled 'Child Bus Route Availability/Instructions'. It contains fields for 'Parent Name' (with a placeholder '<<Label>>'), 'Child Name' (placeholder '<<Label>>'), 'Current Bus Route' (placeholder '<<Label>>'), and 'Active' (with an unchecked checkbox). There is also a large text area for 'Instructions'. At the bottom of the form are 'Save' and 'Cancel' buttons. A copyright notice '© 2023 Name' is located at the bottom left.

Chrome	Fleet Industries' Logo	Home > Users > Parent	Login
Home	Primary	Primary	Primary
<b>Child Bus Route Availability/Instructions</b>			
Parent Name:	<<Label>>		
Child Name:	<<Label>>		
Current Bus Route:	<<Label>>		
Active:	<input type="checkbox"/>		
Instructions:			
Save Cancel			
© 2023 Name			

## Add Dependent

Chrome

Fleet Industries' Logo      Home > Users > Parent

Home Primary Primary Primary

Login Register Now!

**New Child**

Last Name:	Text box	DOB:	JQuery Date Picker
First Name:	Text box	Gender:	Combo Box
MI:	Text box	Height:	Text box
Street Address:	Text box	Weight:	Text box
City:	Text box	Hair Color:	Combo Box
State:	Combo Box	School:	Combo Box
ZIP:	Text box	Grade:	Combo Box

Create      Cancel

© 2023 Name

## Edit Bus Route Assignment - Web

Chrome

Fleet Industries' Logo      Home > Users > Parent

Home Primary Primary Primary

Login      Register Now!

**Child Bus Route Assignment**

Parent Name: <<Label>>

Child Name: <<Label>>

Current Bus Route: <<Label>>

New Bus Route:

Save      Cancel

© 2023 Name

## Edit Dependent

Chrome

Fleet Industries' Logo     Home > Users > Parent     Login     Register Now!

Home     Primary     Primary     Primary

Child Details

Last Name:	<<Label>>	DOB:	<<Label>>
First Name:	<<Label>>	Gender:	<<Label>>
MI:	<<Label>>	Height:	<<Label>>
Street Address:	<<Label>>	Weight:	<<Label>>
City:	<<Label>>	Hair Color:	<<Label>>
State:	<<Label>>	School:	<<Label>>
ZIP:	<<Label>>	Grade:	<<Label>>
		Bus Route:	<<Label>>     Edit

Edit     Cancel

© 2023 Name

## Set Bus Route Availability - Web

Chrome

Fleet Industries' Logo      Home > Users > Parent

Home Primary Primary Primary

Login Register Now!

**Set Child Bus Route Availability.**

Parent Name: <<Label>>

Child Name: <<Label>>

Current Bus Route: <<Label>>

Active:

Save Cancel

© 2023 Name

## View Bus History Report - Web

Chrome

Fleet Industries' Logo      Home > Users > Parent

Home Primary Primary Primary

Login Register Now!

**Child Bus Ride History Report**

Parent Name: <<Label>> Child Name: <<Label>> Current Bus Route: <<Label>>

Report Start: JQuery DatePicker Report End: JQuery DatePicker Run Report Cancel

Results: <<Date/Time On/Time Off>>

© 2023 Name

## View Current Status - Web

Chrome

Fleet Industries' Logo Home > Users > Parent

Home Primary Primary Primary

Login Register Now!

**Child Bus Ride Status/Location**

Parent Name: <<Label>> Child Name: <<Label>> Current Bus Route: <<Label>>

Estimated Location: <<Label>>

© 2023 Name

# UI Drawings for Parts Person

## Parts Add Part to Inventory - DESKTOP

Window

Fleet Industries' LOGO

Part Request    Inventory    Part Orders    Primary

Profil    Account Name

Add Part to Inventory

**Part #**

Packing Slip #

Part ID #

Quantity

**Parts Notes**

275 x 261

© 2023 Name

## Parts Confirm Parts Request - DESKTOP

Window

Fleet Industries' LOGO

Account Name

Profil

Part Request    Parts Order    Inventory    Receiving

**Part Inventory**

Part Name	Part Number	Vendor #	On Hand Qty	# Ordered	Stock Level	Audit
Content	(Hyperlink)	Content	Content	Content	Content	Link
Content	(Hyperlink)	Content	Content	Content	Content	Link
Content	(Hyperlink)	Content	Content	Content	Content	Link
Content	(Hyperlink)	Content	Content	Content	Content	Link
Content	(Hyperlink)	Content	Content	Content	Content	Link
Content	(Hyperlink)	Content	Content	Content	Content	Link
Content	(Hyperlink)	Content	Content	Content	Content	Link
Content	(Hyperlink)	Content	Content	Content	Content	Link

Search

Filter

Back

© 2023 Name

## Parts Create Change Order - DESKTOP

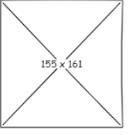
Window

Fleet Industries' LOGO

Part Requests Part Orders Inventory Receiving Profil Account Name Back

Part Order: #

Part to Change:

Part Name	<input type="text"/>	
Part Description	<input type="text"/> Input	
Current Quantity	<input type="text"/> Input	
New Quantity	<input type="text"/> Input	
Updated Price	<input type="text"/>	<input type="button"/> Back <input type="button"/> Submit

Request # Part Supplier Price total


Finalize Changes Cancel Change

© 2023 Name  
© 2023 Name

## Parts Part Inventory Audit - DESKTOP

Window

Fleet Industries' LOGO

Tab Tab Tab Tab

Profil Account Name

### Inventory Audit

**Part Number**

Vendor #

Stock Level

Description

Expected QoH  TextBox - on-hand currently in the system

Actual QoH  Input - the counted on-hand

275 x 261

Back Submit

© 2023 Name

## Parts Purchase Order Details - DESKTOP

Window

Fleet Industries' LOGO

Part Requests Part Orders Inventory Receiving Profil Account Name Back

Part Order: #

Part Request: #

Part Name

Quantity

Part Description

Vehicle

Price  Price for one item

155 x 161

Back Done

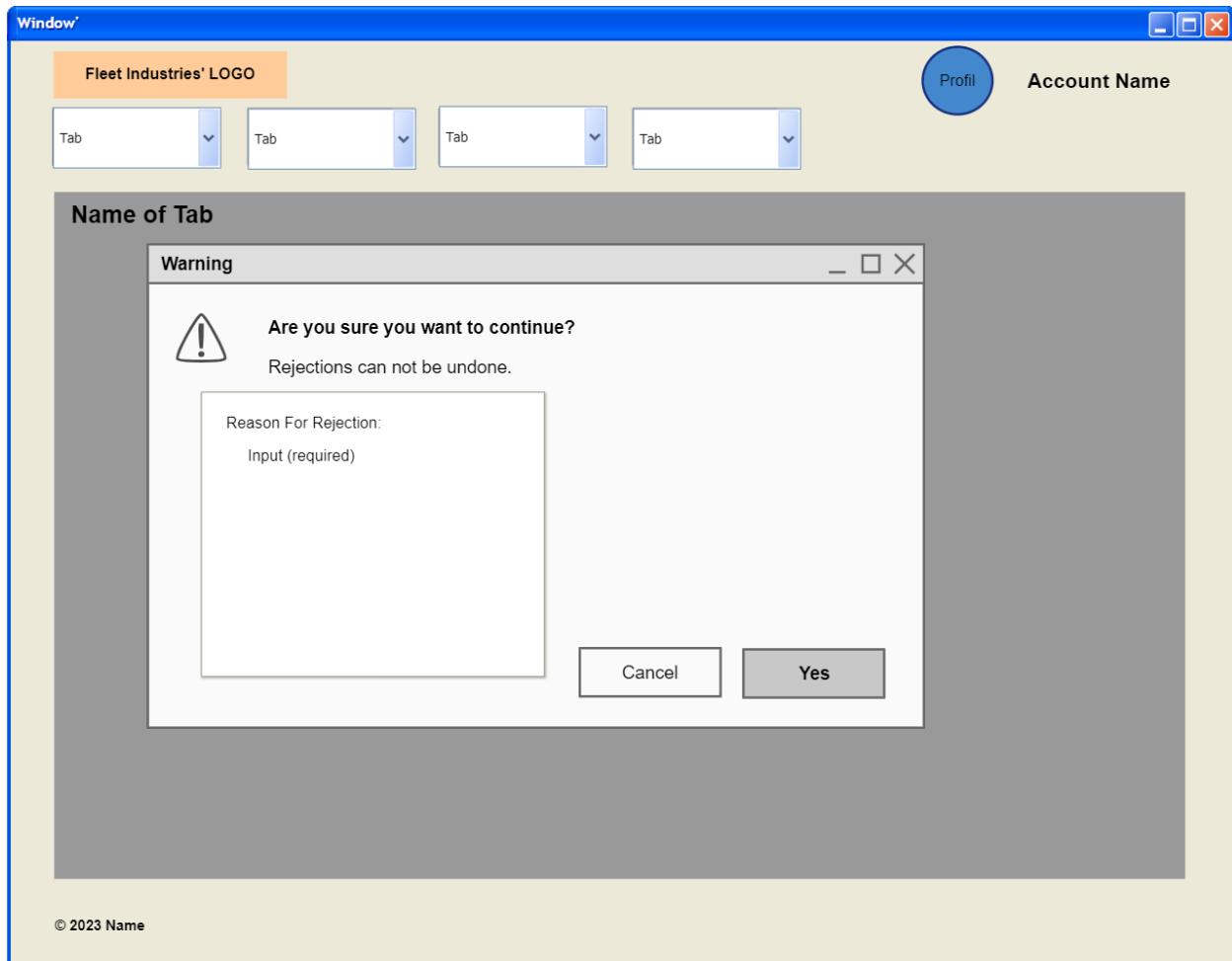
Request #	Part	Supplier	Price	total

View Changes  
Create Change  
Cancel Order

© 2023 Name  
© 2023 Name

460 | Page

## Parts Reject Part Request - DESKTOP



## Parts View Current PO Details - DESKTOP

Window

Fleet Industries' LOGO

Part Requests Part Orders Inventory Receiving Profil Account Name Back

Part Order: #

Part Request: #

Part Name

Quantity

Part Description

Vehicle

Price  Price for one item

Back Remove

195 x 161

Request #	Part	Supplier	Price	total

Cancel Order Send Order

© 2023 Name  
© 2023 Name

## Parts View Park Requests - DESKTOP

Window

Fleet Industries' LOGO

Account Name

Profil

Part Requests    Part Order    Inventory    Receiving

### Part Requests

There are \_\_ number of requests

Search

Filter

Request #	Part Requested	Quantity	Request Date
Cell Content 1(hyperlink)	Content	Content	Content
Cell content 2	Content	Content	Content
Cell content 3	Content	Content	Content
Cell content 4	Content	Content	Content
Cell content 5	Content	Content	Content
Cell content 6	Content	Content	Content
Cell content 7	Content	Content	Content
Cell content 8	Content	Content	Content

© 2023 Name

## Parts View Part Request Details - DESKTOP

Window

Fleet Industries' LOGO

Account Name

Profil

Part Requests    Part Orders    Inventory    Receiving

**Request Details**

**Request #**

**Part Requested**  Add to Order

**Quantity**

**Vehicle**

**Notes**   
with Service request ID with link to details

**Date Requested**

**Request From**

275 x 261

Back    Reject

© 2023 Name

## Parts View Purchase Orders - DESKTOP

Window

Fleet Industries' LOGO

Account Name

Part Requests    Part Order    Inventory    Receiving

Part Orders

There are \_\_ unfulfilled orders

PO #	Vendor #	Creation Date	Fulfilled	Recieve
Cell Content 1(hyperlink)	Content	Content	True	<a href="#">Recieve</a>
Cell content 2	Content	Content	False	<a href="#">Recieve</a>
Cell content 3	Content	Content	Content	<a href="#">Recieve</a>
Cell content 4	Content	Content	Content	<a href="#">Recieve</a>
Cell content 5	Content	Content	Content	<a href="#">Recieve</a>
Cell content 6	Content	Content	Content	<a href="#">Recieve</a>
Cell content 7	Content	Content	Content	<a href="#">Recieve</a>
Cell content 8	Content	Content	Content	<a href="#">Recieve</a>

Search

Filter

Back      Create Order

© 2023 Name

## Parts View Special PO Details - DESKTOP

Window

Fleet Industries' LOGO

Part Requests Part Orders Inventory Receiving Profil Account Name

Special Order: #

Part Description

Vehicle Name

Request #

Supplier

Price

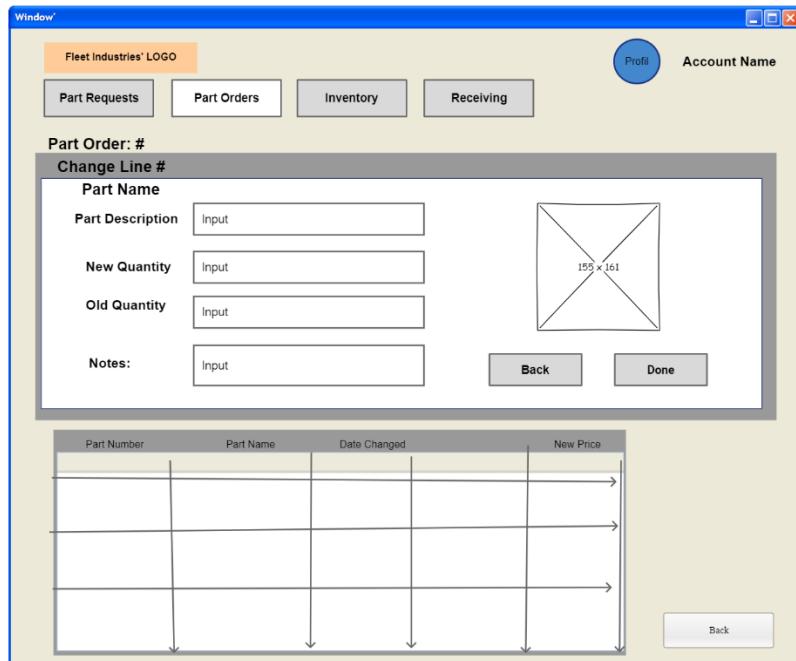
Approved By  Back Remove

176 x 161

Request #	Part	Vehicle	Supplier	Approved	Price

© 2023 Name

## UI\_Part.03.06/Desktop



## UI\_Part.03.07/Desktop

Window

Fleet Industries' LOGO

Profil Account Name

Part Requests Inventory Purchase Orders Recieving

**Receive PO: (po #)**

Part #	Part Name	Should Receive:	Receiving:
#	Name	Number	# ▾
#	Name	Number	# ▾

Notes

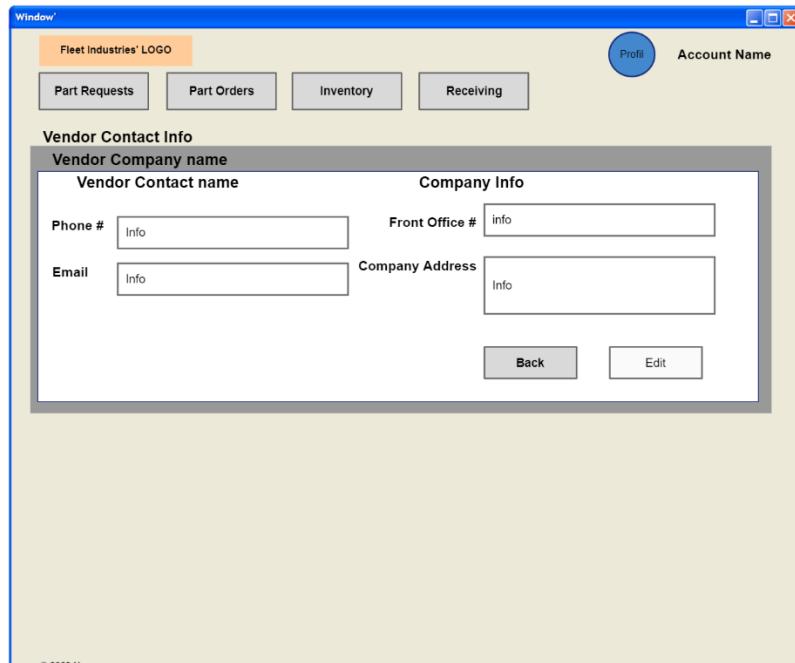
Text box

Submit

Part number and number received expand with more items on the po

© 2023 Name

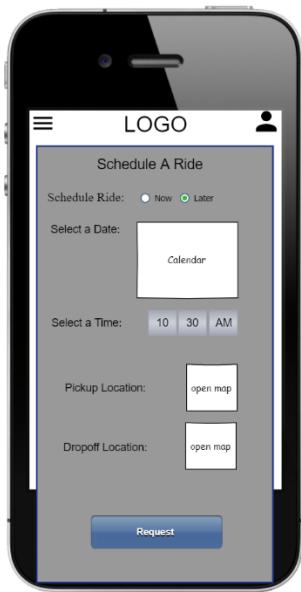
## UI\_Part.04.01/Desktop



# UI Drawings for Ride Request Later

## Ride Request Later Request Ride MOBILE

RRL.10.01.Mobile



# UI Drawings for Ride Request Now

## Details Of Active Services WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab Tab Tab Tab

**Services**

**SERVICE NAME**

Type	TYPE
Running	Yes
Area Serviced	AREA
Hours	DAYs / HOURS
# of Vehicles	NUM VEHICLES

275 x 261

Request Ride      Schedule Ride

© 2023 Name

## Immediate Ride Request WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab      Tab      Tab      Tab

**Services**

**SERVICE NAME**

Type  
Running

Estimated Pickup Time  
**00:00**

Estimated Dropoff Time  
**00:20**

Would you like to request this ride for the earliest time?

Confirm      Cancel

Area Serviced

Hours

# of Vehicles      NUM VEHICLES

275 x 261

Request Ride      Schedule Ride

© 2023 Name

## Next Available Ride Request WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab      Tab      Tab      Tab

**Scheduled Rides**      Search       Filter

Service	Pickup Time	Driver Name	Requested
REQUESTED SERVICE	~ESTIMATED PICKUP TIME	DRIVER NAME	
SCHEDULED SERVICE			
SCHEDULED SERVICE			

© 2023 Name

## View Active Transit Services WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab      Tab      Tab      Tab

**Services**      Search  Filter  Running      Start Location  End Location  Picture

Service Name      Service Type      Area      Running

NAME	TYPE	AREA	Running	Picture
				X
				X
				X

© 2023 Name

## View Services By Start And End Location WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

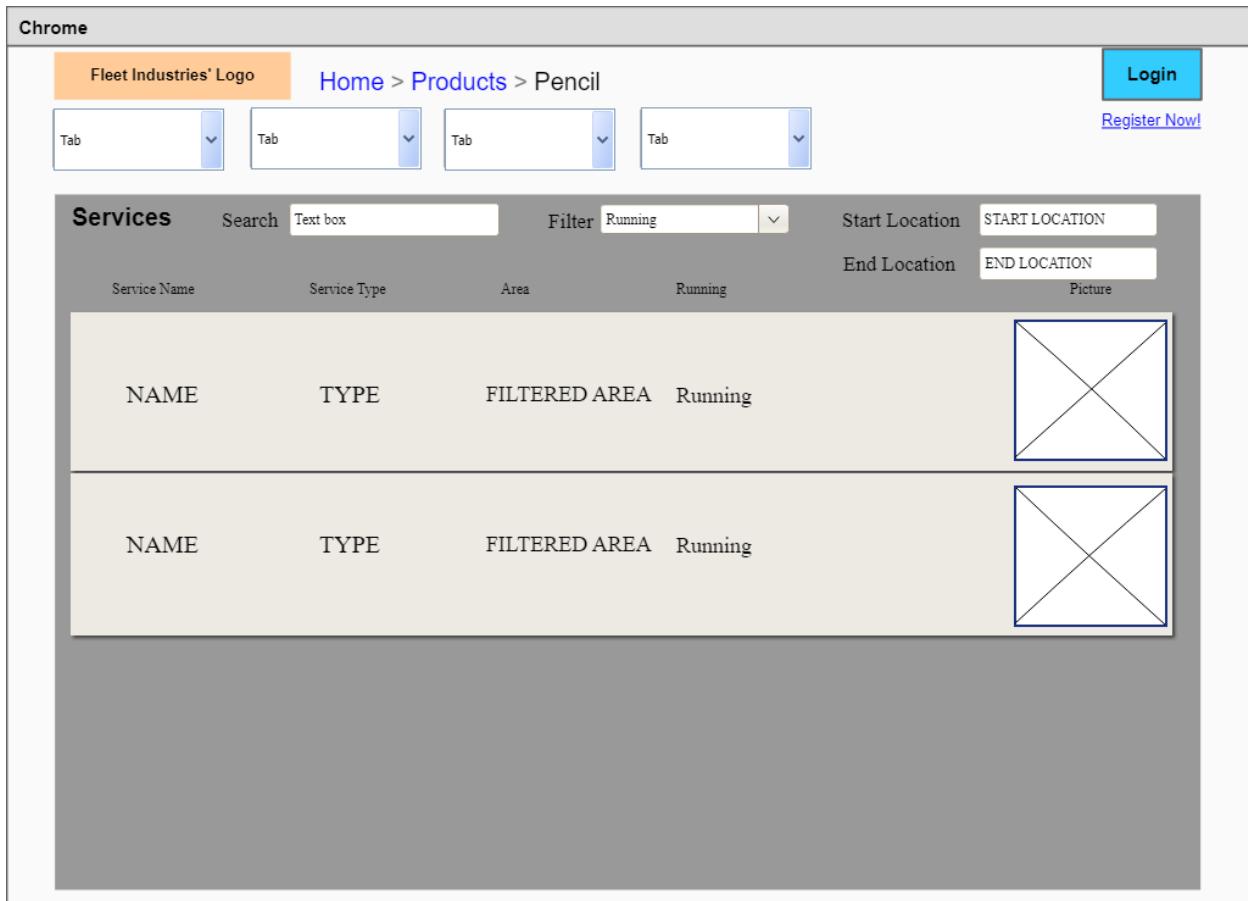
Tab      Tab      Tab      Tab

**Services**      Search  Filter  Start Location   
Service Name      Service Type      Area      Running      End Location   
Picture

NAME	TYPE	FILTERED AREA	Running

NAME      TYPE      FILTERED AREA      Running

© 2023 Name



# UI Drawings for Ride Schedule Later

## Time Table WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab Tab Tab Tab

Services

SERVICE NAME

Choose an Available Time

| September 12 Monday |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| 12:00               | 12:00               | 12:00               | 12:00               | 12:00               | 12:00               |
| 12:15               | 12:15               | 12:15               | 12:15               | 12:15               | 12:15               |
| 12:30               | 12:30               | 12:30               | 12:30               | 12:30               | 12:30               |
| 12:00               | 12:00               | 12:00               | 12:00               | 12:00               | 12:00               |
| 12:00               | 12:00               | 12:00               | 12:00               | 12:00               | 12:00               |
| 12:00               | 12:00               | 12:00               | 12:00               | 12:00               | 12:00               |
| 12:30               | 12:30               | 12:30               | 12:30               | 12:30               | 12:30               |

Date

←      Confirm      Cancel      →

Request Ride      Schedule Ride

© 2023 Name

## Location Entry And Active Services WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab      Tab      Tab      Tab

**Services**      Search       Filter       Start Location  START LOCATION  
Service Name      Service Type      Area      Running      End Location  END LOCATION      Picture

NAME      TYPE      FILTERED AREA      Running/Not Running

NAME      TYPE      FILTERED AREA      Running/Not Running

© 2023 Name

The screenshot shows a web browser window for the 'Fleet Industries' website. At the top, there's a header with the company logo, a breadcrumb trail ('Home > Products > Pencil'), and links for 'Login' and 'Register Now!'. Below the header, there are four tabs labeled 'Tab' with dropdown arrows. The main content area has a dark grey header with search and filter fields, and dropdowns for 'Start Location' and 'End Location'. It also includes buttons for 'Service Name', 'Service Type', 'Area', 'Running', 'Picture', and 'FILTERED AREA'. Below this header, there are two rows of data tables, each with columns for 'NAME', 'TYPE', 'FILTERED AREA', and 'Running/Not Running'. To the right of each row is a large blue-bordered square containing a black 'X'. At the bottom of the page, there's a copyright notice: '© 2023 Name'.

## Schedule Ride WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab      Tab      Tab      Tab

**Services**

**SERVICE NAME**

Type  
Running

Area Serviced

Hours

# of Vehicles

TYPE  
Estimated Pickup Time  
**00:00**

Estimated Dropoff Time  
**00:20**

Would you like to request this ride for the earliest time?

Confirm      Cancel

275 x 261

Request Ride      Schedule Ride

© 2023 Name

## View Active Transit Services WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab      Tab      Tab      Tab

**Services**      Search       Filter       Start Location       End Location   
Service Name      Service Type      Area      Running      Picture

NAME	TYPE	AREA	Running/Not Running	Picture
				X
				X
				X

© 2023 Name

## View Details Of Active Service WEB

Chrome

Fleet Industries' Logo      Home > Products > Pencil      Login      Register Now!

Tab      Tab      Tab      Tab

**Services**

**SERVICE NAME**

Type	TYPE
Running	[Yes / No]
Area Serviced	AREA
Hours	DAYs / HOURS
# of Vehicles	NUM VEHICLES

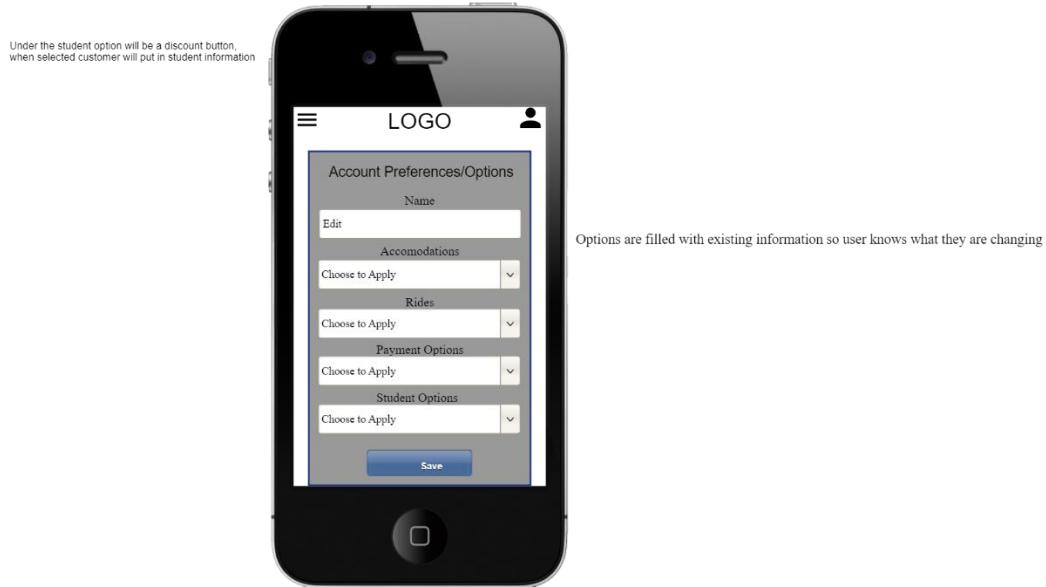
275 x 261

Request Ride      Schedule Ride

© 2023 Name

# UI Drawings for Student

## Edit Account Preferences - mobile



## Schedule Ride - mobile

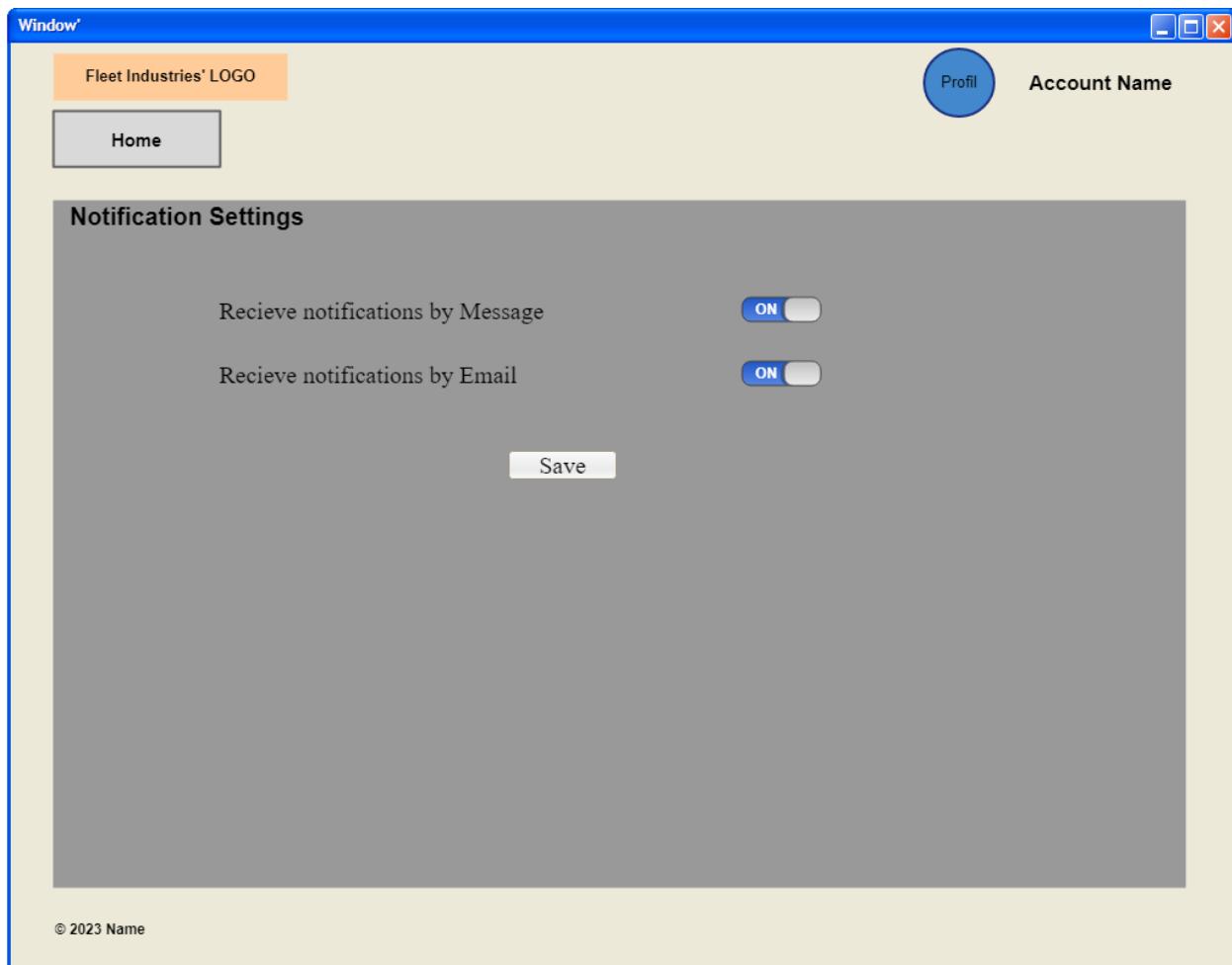


## View Available Rides - mobile

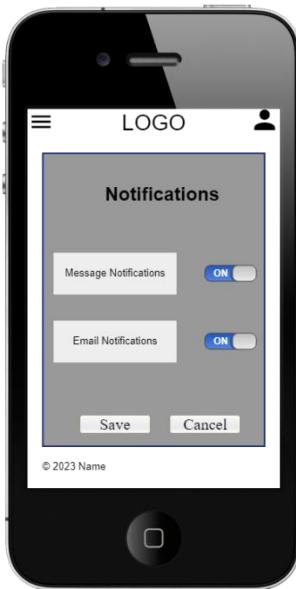


# UI Drawings for User

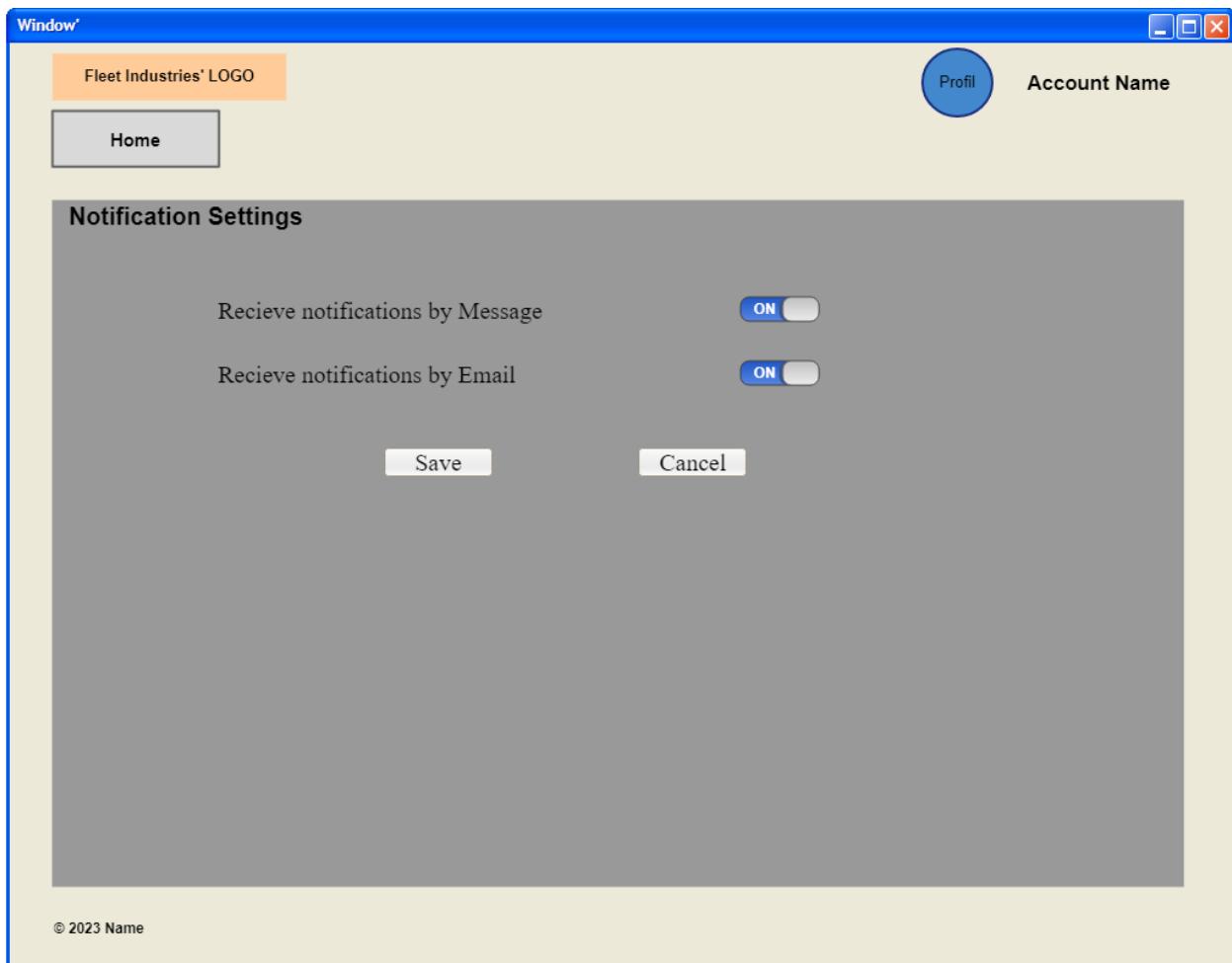
## Change Communications Settings Desktop



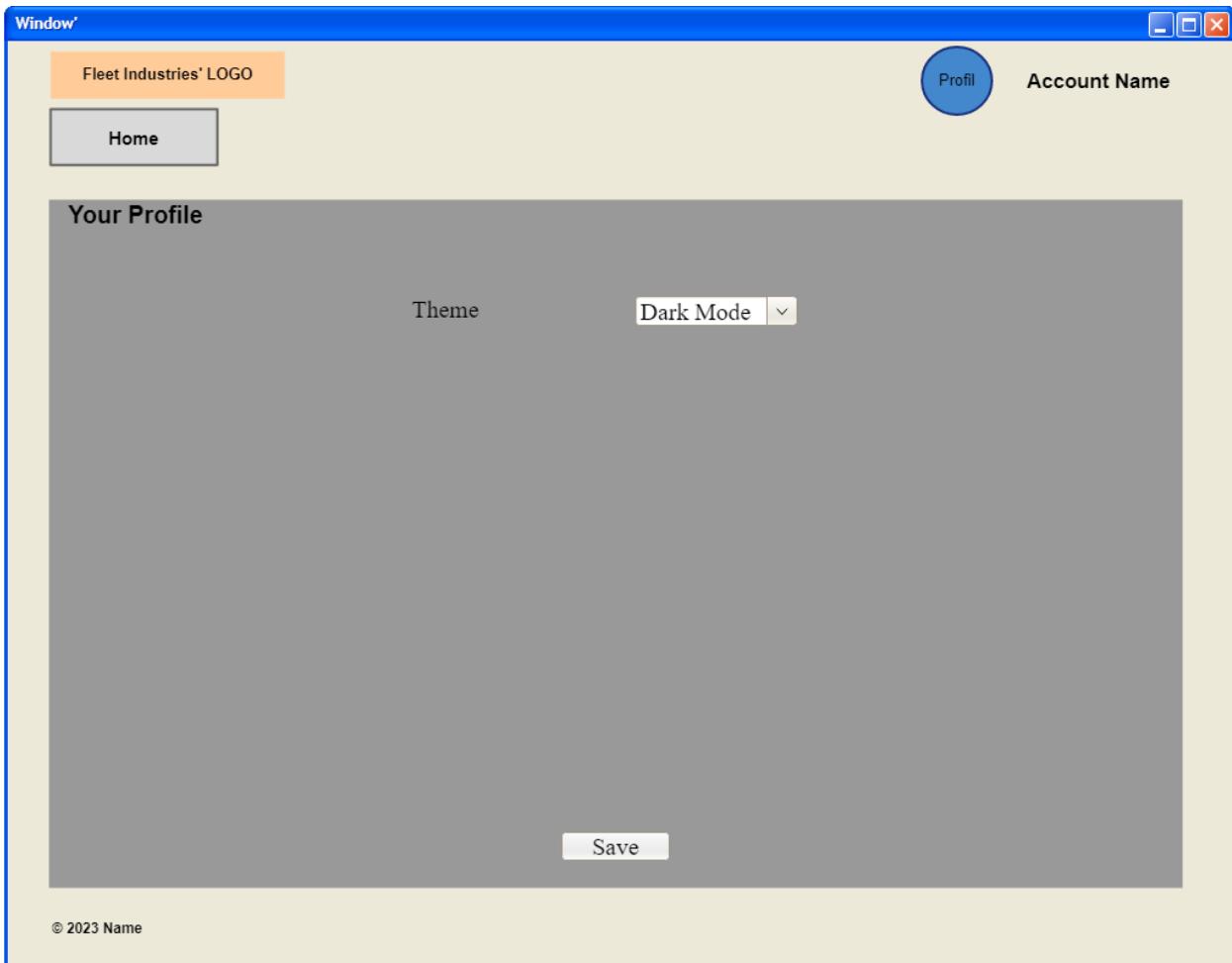
## Change Communications Settings Mobile



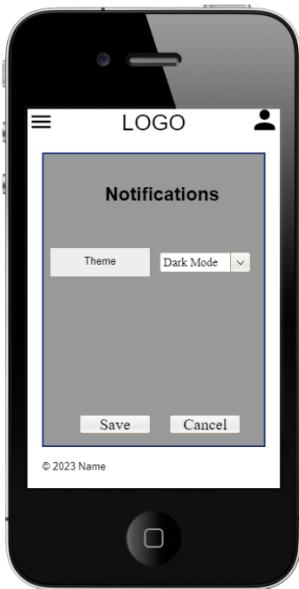
## Change Communications Settings Web



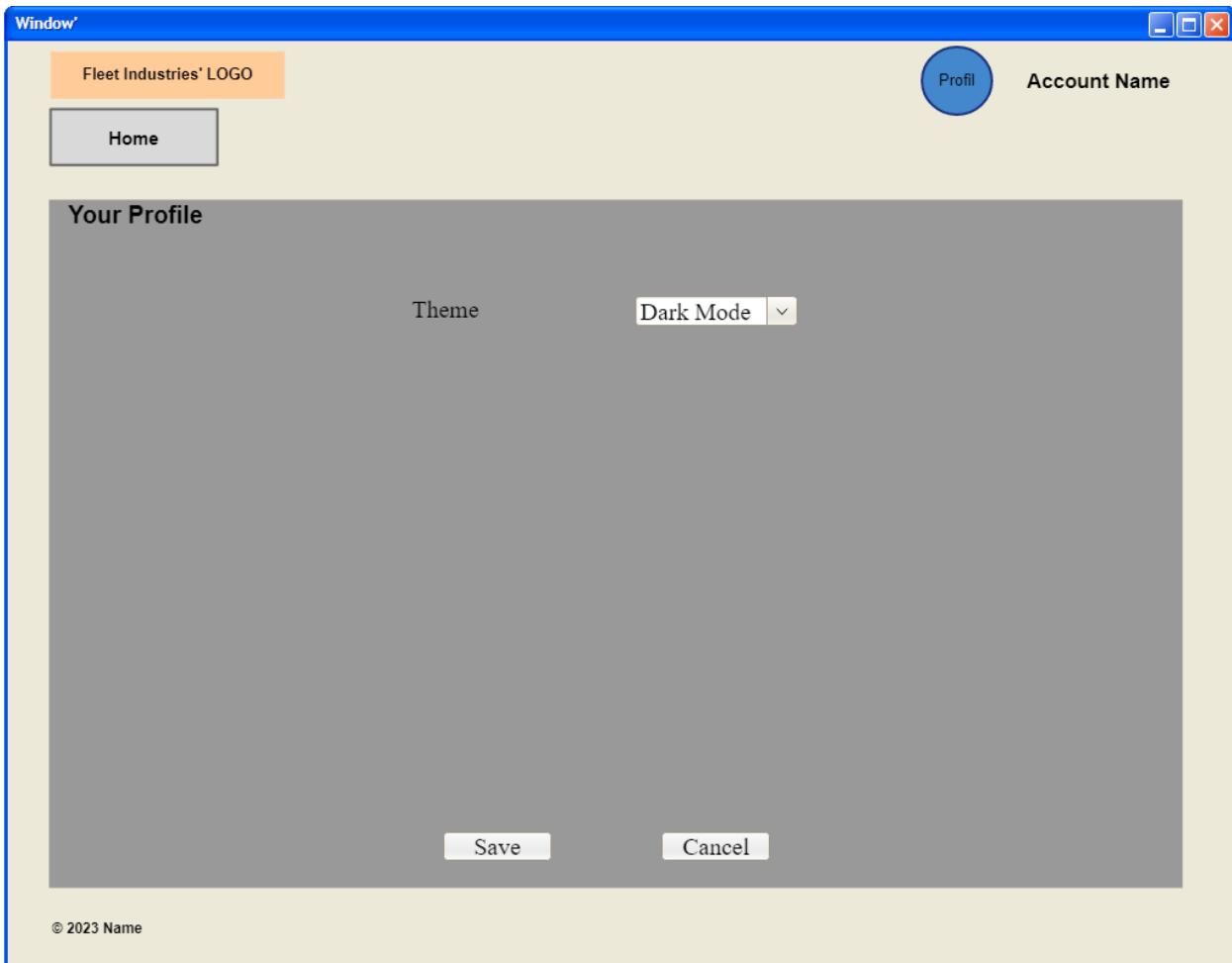
## Change UX Settings Desktop



## Change UX Settings Mobile



## Change UX Settings Web



## Create Account Desktop

Window

Fleet Industries' LOGO

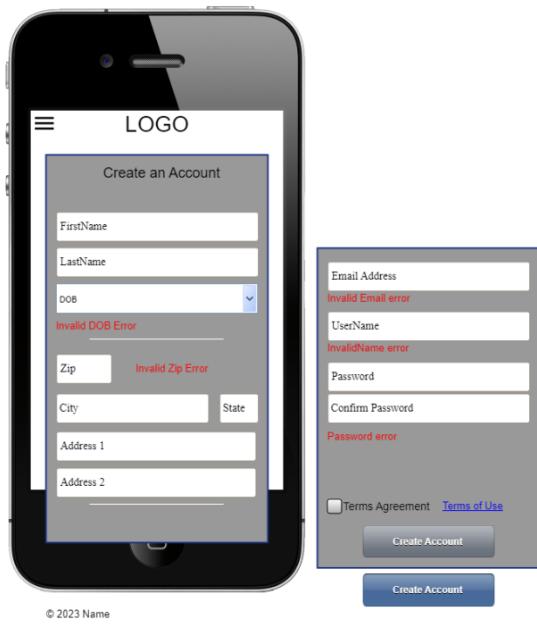
Home

### Create Account

First Name	Text box	
Last Name	Text box	
DOB	Combo Box	Invalid DOB Error
Zip	Text box	Invalid Zip Error
City	Text box	
State	Text box	
Address 1	Text box	
Address 1	Text box	
Email	Text box	Invalid Email error
User Name	Text box	InvalidName error
Password	Text box	
Confirm Password	Text box	Password error
<input type="checkbox"/> Terms Agreement <a href="#">Terms of Use</a>		
Create		

© 2023 Name

## Create Account Mobile



© 2023 Name

## Create Account Web

Chrome

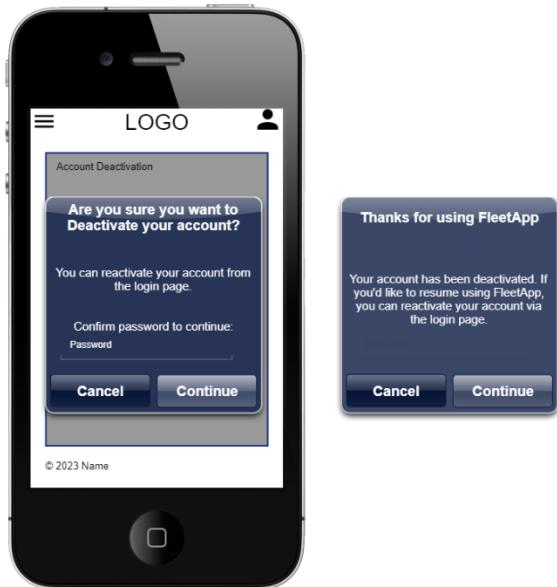
Fleet Industries' Logo

Home

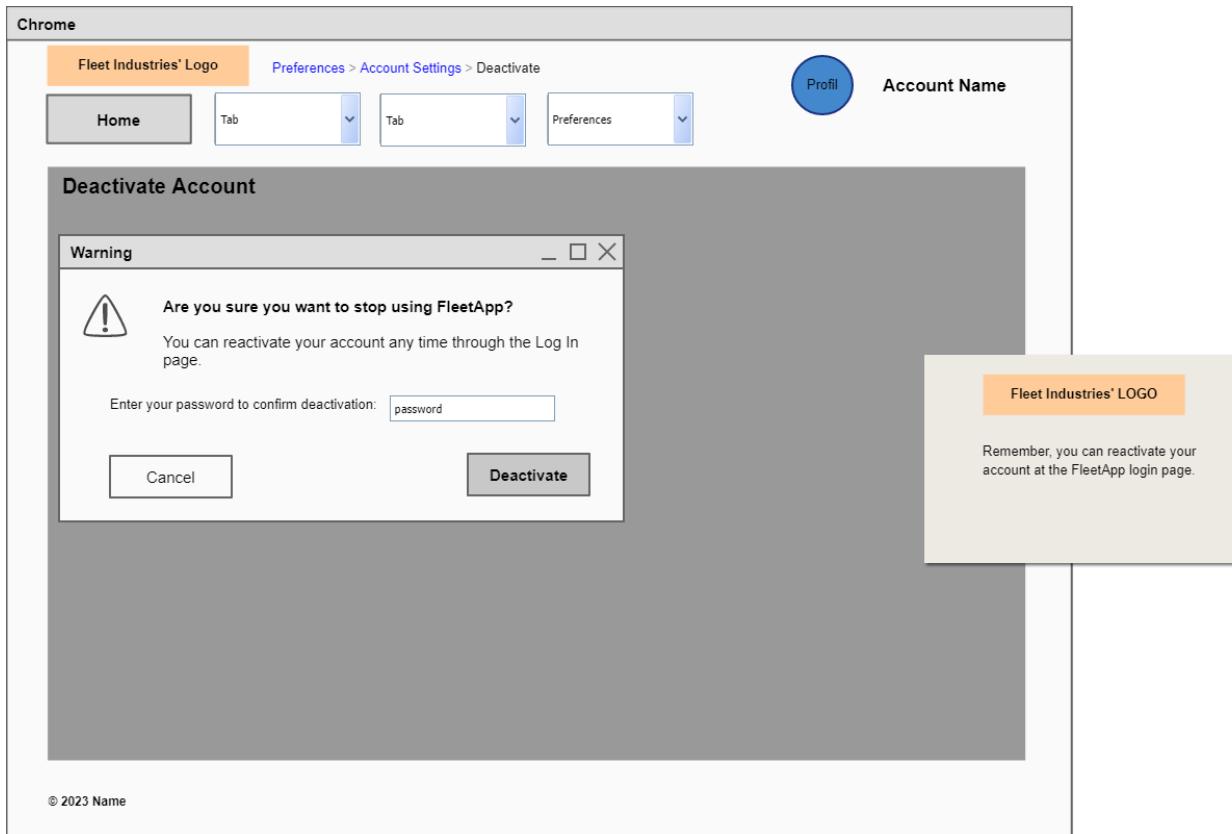
Last Name	Text box	
DOB	Combo Box	Invalid DOB Error
Zip	Text box	Invalid Zip Error
City	Text box	
State	Text box	
Address 1	Text box	
Address 1	Text box	
Email	Text box	Invalid Email error
User Name	Text box	InvalidName error
Password	Text box	
Confirm Password	Text box	Password error
<input type="checkbox"/> Terms Agreement <a href="#">Terms of Use</a>		
<a href="#">Create</a>		

© 2023 Name

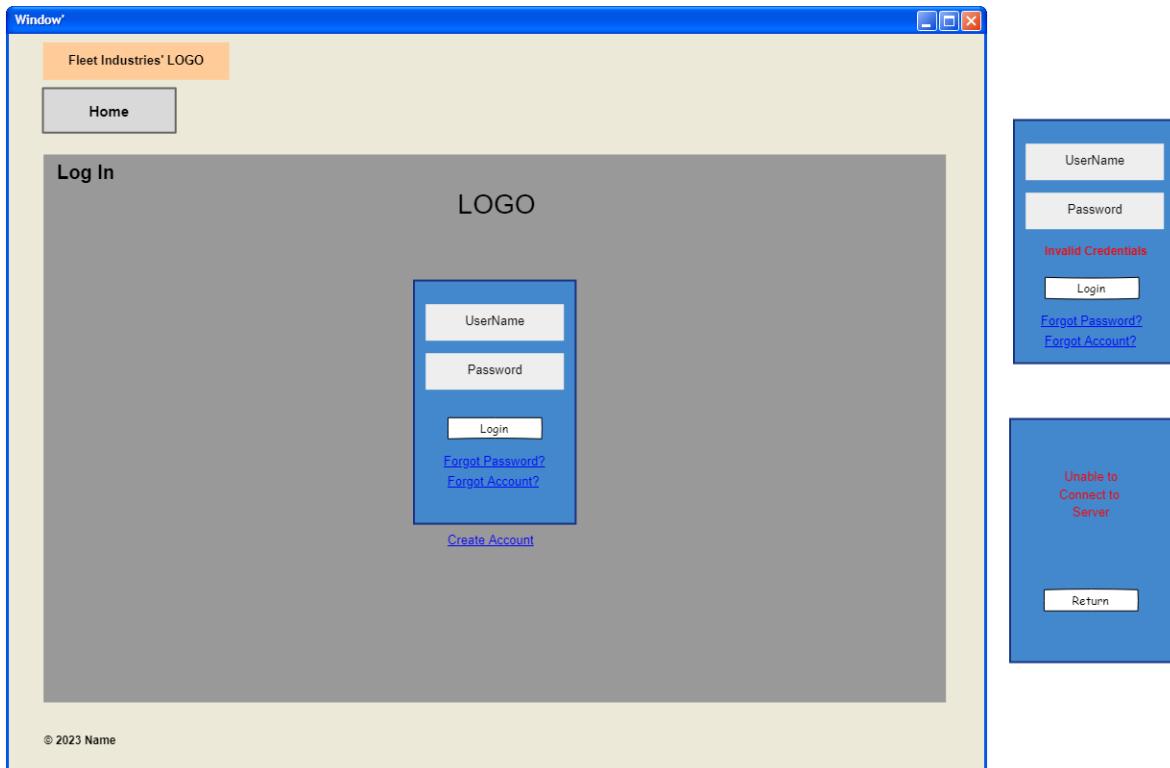
## Deactivate Account Mobile



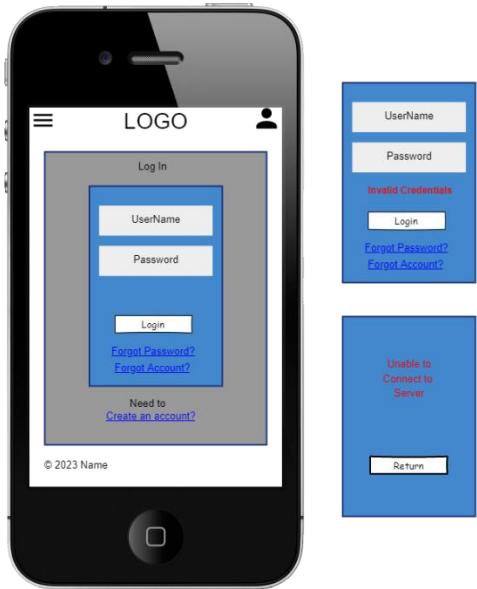
## Deactivate Account Web



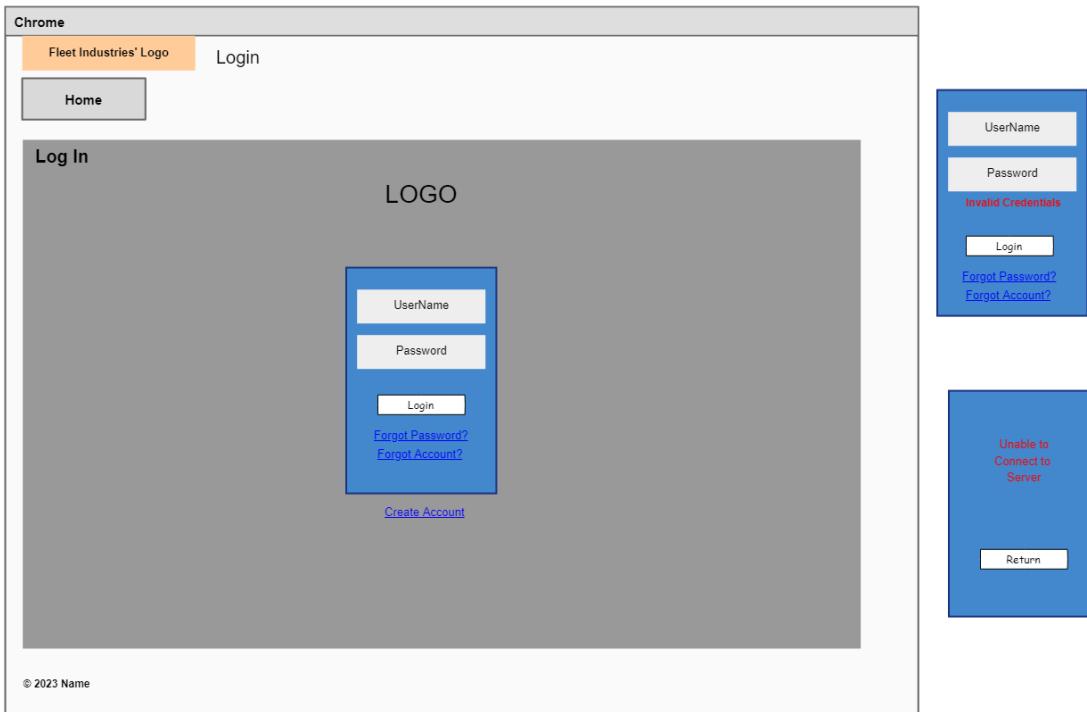
## Log In Desktop



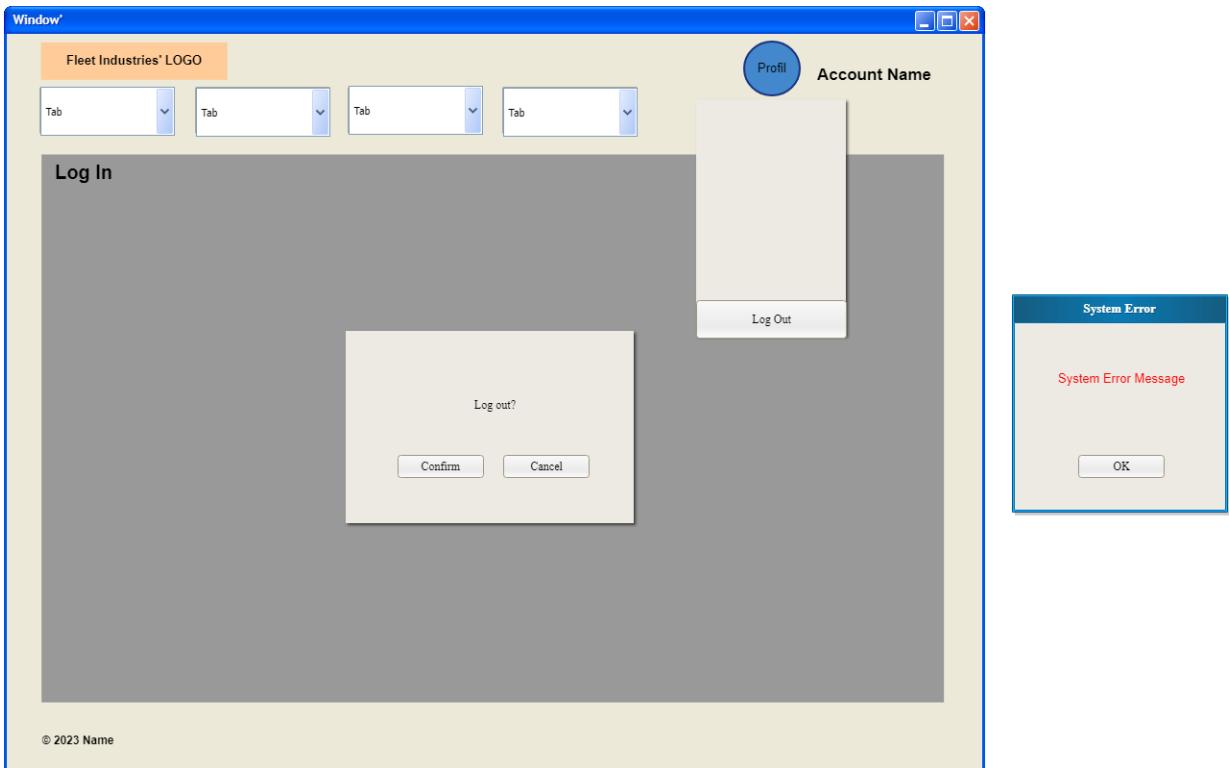
## Log In Mobile



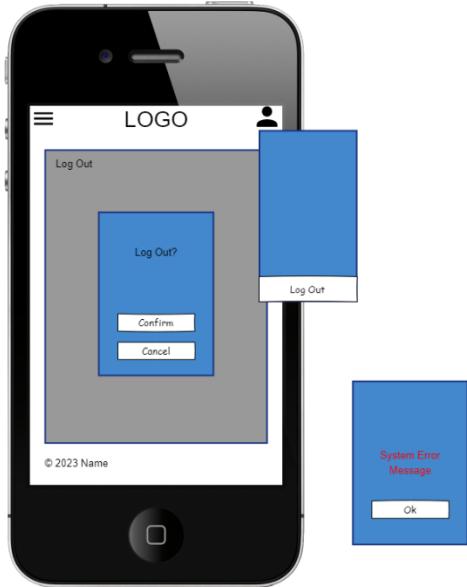
## Log In Web



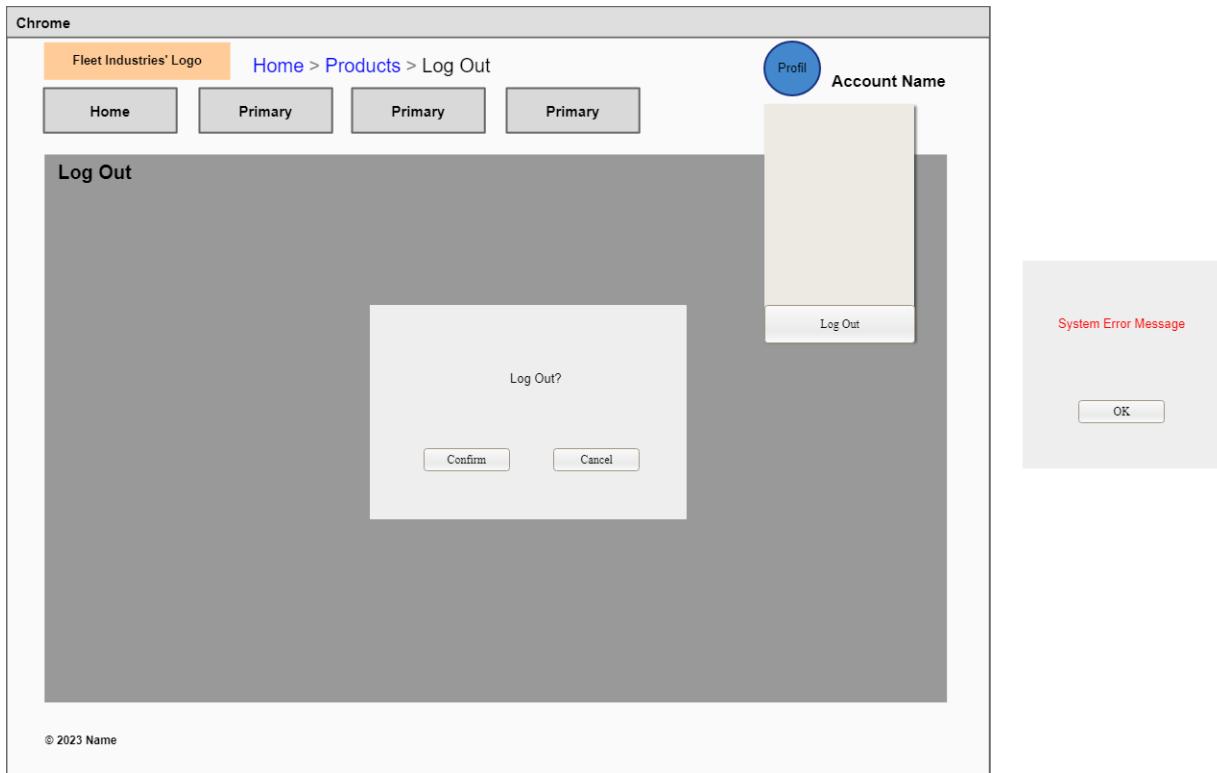
## Log out Desktop



## Log Out Mobile



## Log Out Web



## Modify Payment Info Desktop

Window

Fleet Industries' LOGO

Home Profil Account Name

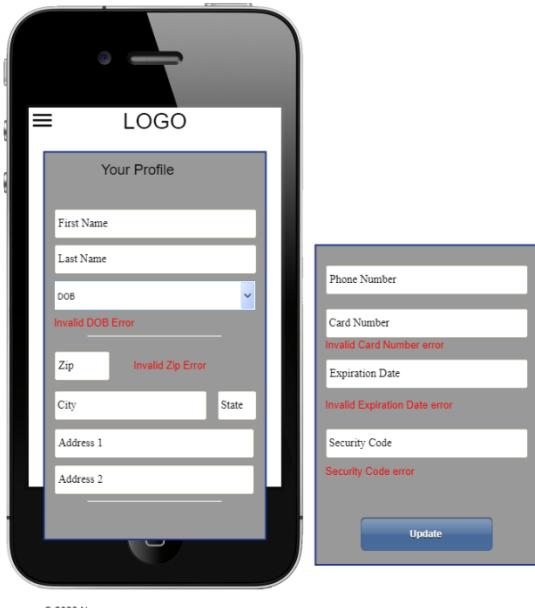
### Payment Method

First Name	Text box	
Last Name	Text box	
DOB	Combo Box <input type="button" value="▼"/>	Invalid DOB Error
Zip Code	Text box	Invalid Zip Error
City	Text box	
State	Text box	
Address 1	Text box	
Address 2	Text box	
Phone Number	Text box	
Card Number	Text box	Invalid Card Number error
Experitaion Date	Text box	Invalid Expiration Date error
Security Code	Text box	Invalid Security Code error

Update Cancel

© 2023 Name

## Modify Payment Info Mobile



© 2023 Name

## Modify Payment Info Web

Window

Fleet Industries' LOGO

Home

Profil Account Name

### Payment Method

First Name	Text box	
Last Name	Text box	
DOB	Combo Box <input type="button" value="▼"/>	Invalid DOB Error
Zip Code	Text box	Invalid Zip Error
City	Text box	
State	Text box	
Address 1	Text box	
Address 2	Text box	
Phone Number	Text box	
Card Number	Text box	Invalid Card Number error
Experitaion Date	Text box	Invalid Expiration Date error
Security Code	Text box	Invalid Security Code error

Update Cancel

© 2023 Name

## Modify Personal Info Desktop

Window

Fleet Industries' LOGO

Home

Profil Account Name

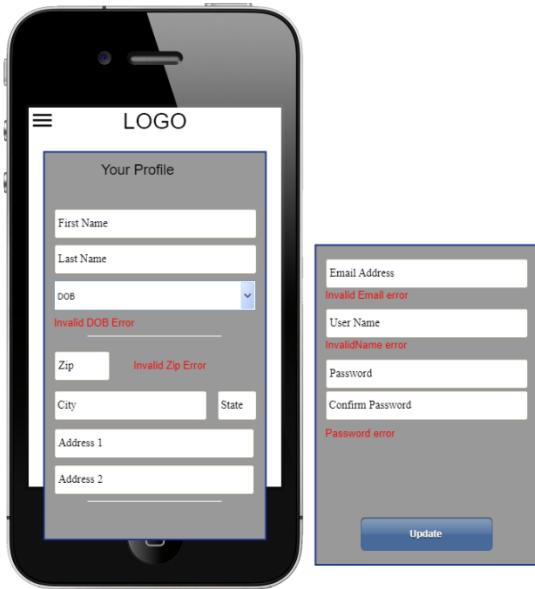
### Your Profile

First Name	Text box	
Last Name	Text box	
DOB	Combo Box <input type="button" value="▼"/>	Invalid DOB Error
Zip Code	Text box	Invalid Zip Error
City	Text box	
State	Text box	
Address 1	Text box	
Address 2	Text box	
Email	Text box	Invalid Email error
User Name	Text box	Invalid Name error
Password	Text box	
Confirm Password	Text box	Password error

Update Cancel

© 2023 Name

## Modify Personal Info Mobile



© 2023 Name

## Modify Personal Info Web

Window

Fleet Industries' LOGO

Home

Profil Account Name

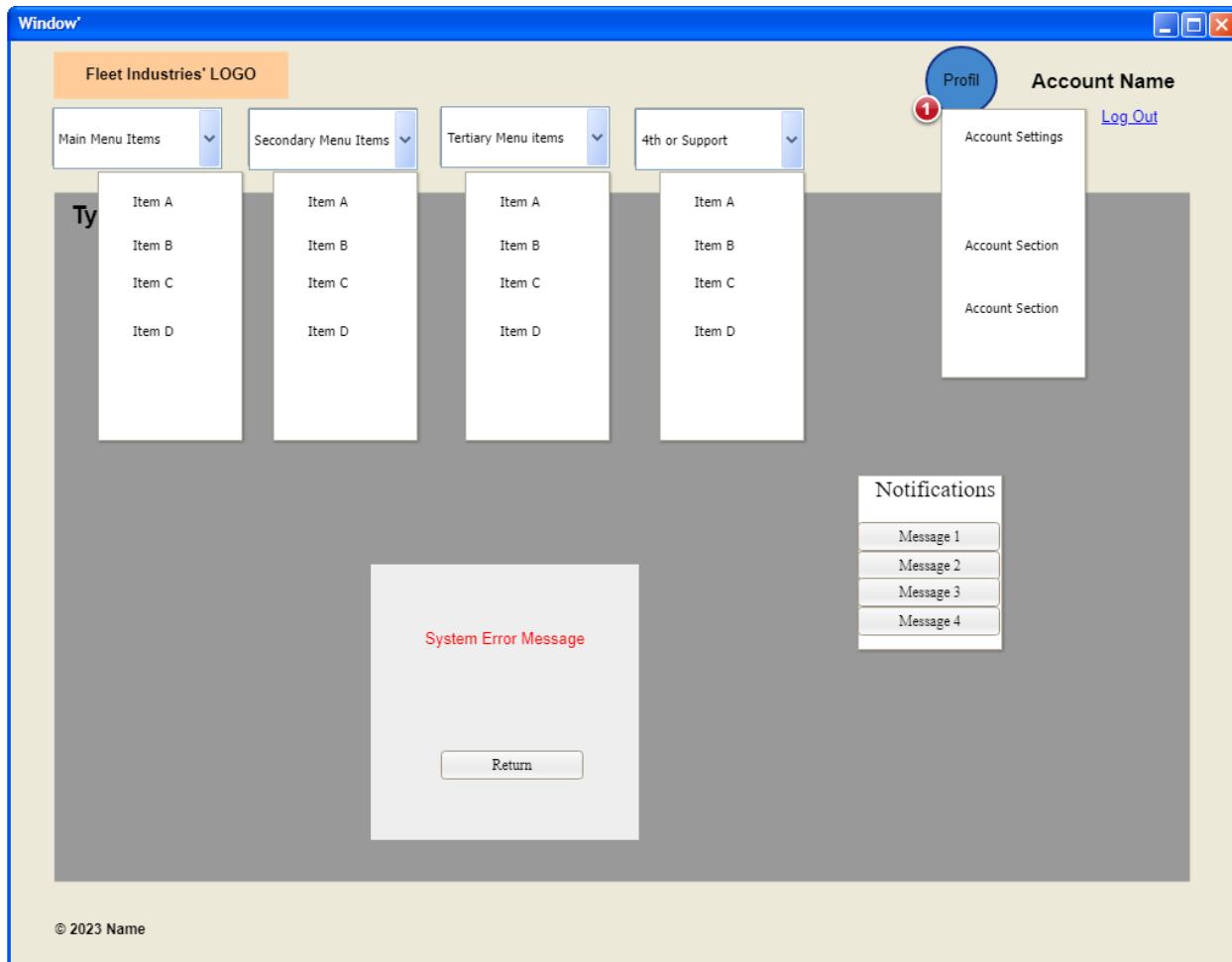
### Your Profile

First Name	Text box	
Last Name	Text box	
DOB	Combo Box <input type="button" value="▼"/>	Invalid DOB Error
Zip Code	Text box	Invalid Zip Error
City	Text box	
State	Text box	
Address 1	Text box	
Address 2	Text box	
Email	Text box	Invalid Email error
User Name	Text box	Invalid Name error
Password	Text box	
Confirm Password	Text box	Password error

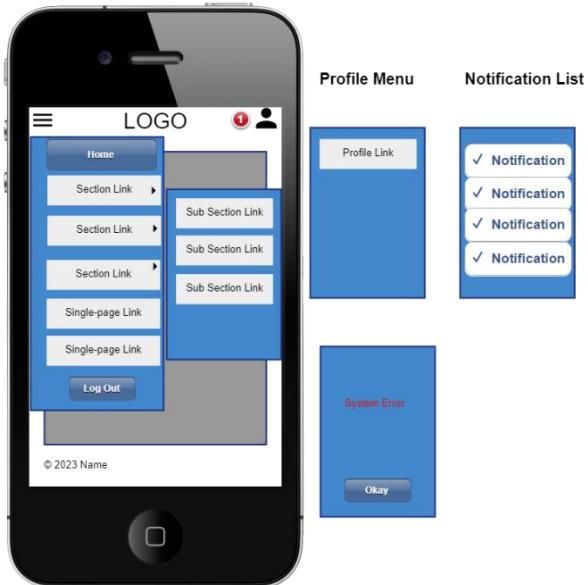
Update Cancel

© 2023 Name

## Navigate Application Desktop



## Navigate Application Mobile



## Navigate Application Web

Chrome

Fleet Industries' LOGO

Home > Section > Page

Main Menu Items ▾ Secondary Menu Items ▾ Tertiary Menu items ▾ 4th or Support ▾

Profil Account Name Log Out

Account Settings

Account Section

Account Section

Item A Item A Item A Item A

Item B Item B Item B Item B

Item C Item C Item C Item C

Item D Item D Item D Item D

System Error Message

Return

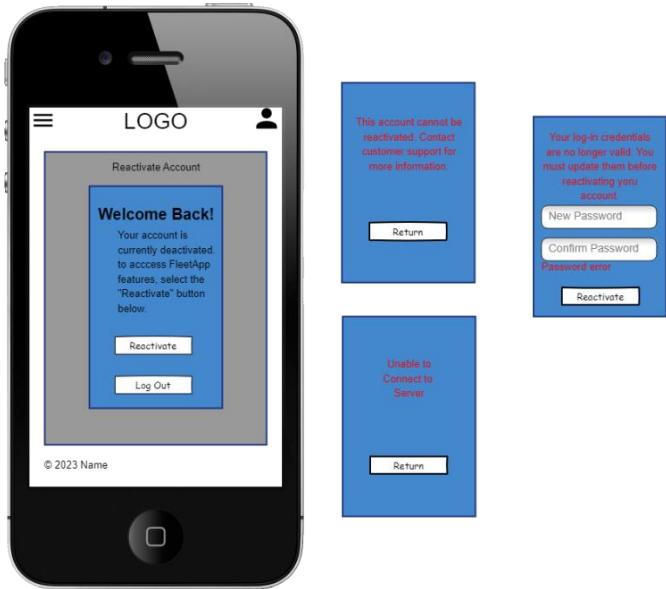
Notifications

Message 1  
Message 2  
Message 3  
Message 4

© 2023 Name

The screenshot displays a web application interface with a header bar containing the Fleet Industries' logo, the current page path (Home > Section > Page), and a user profile section with 'Profil' and 'Account Name' buttons, along with 'Log Out' and 'Account Settings' links. Below the header is a horizontal navigation bar with four dropdown menus: 'Main Menu Items', 'Secondary Menu Items', 'Tertiary Menu items', and '4th or Support'. The main content area is divided into four columns labeled 'Type A', 'Type B', 'Type C', and 'Type D', each containing four items (A, B, C, D). A central modal window titled 'System Error Message' is open, with a 'Return' button at the bottom. To the right of the main content is a 'Notifications' sidebar listing four messages: 'Message 1', 'Message 2', 'Message 3', and 'Message 4'. At the bottom left of the page is a copyright notice: '© 2023 Name'.

## Reactivate Account Mobile



## Reactivate Account Web

The screenshot displays a web application for account reactivation. At the top, there's a header bar with the text "Reactivate Account". Below the header, there are two main sections:

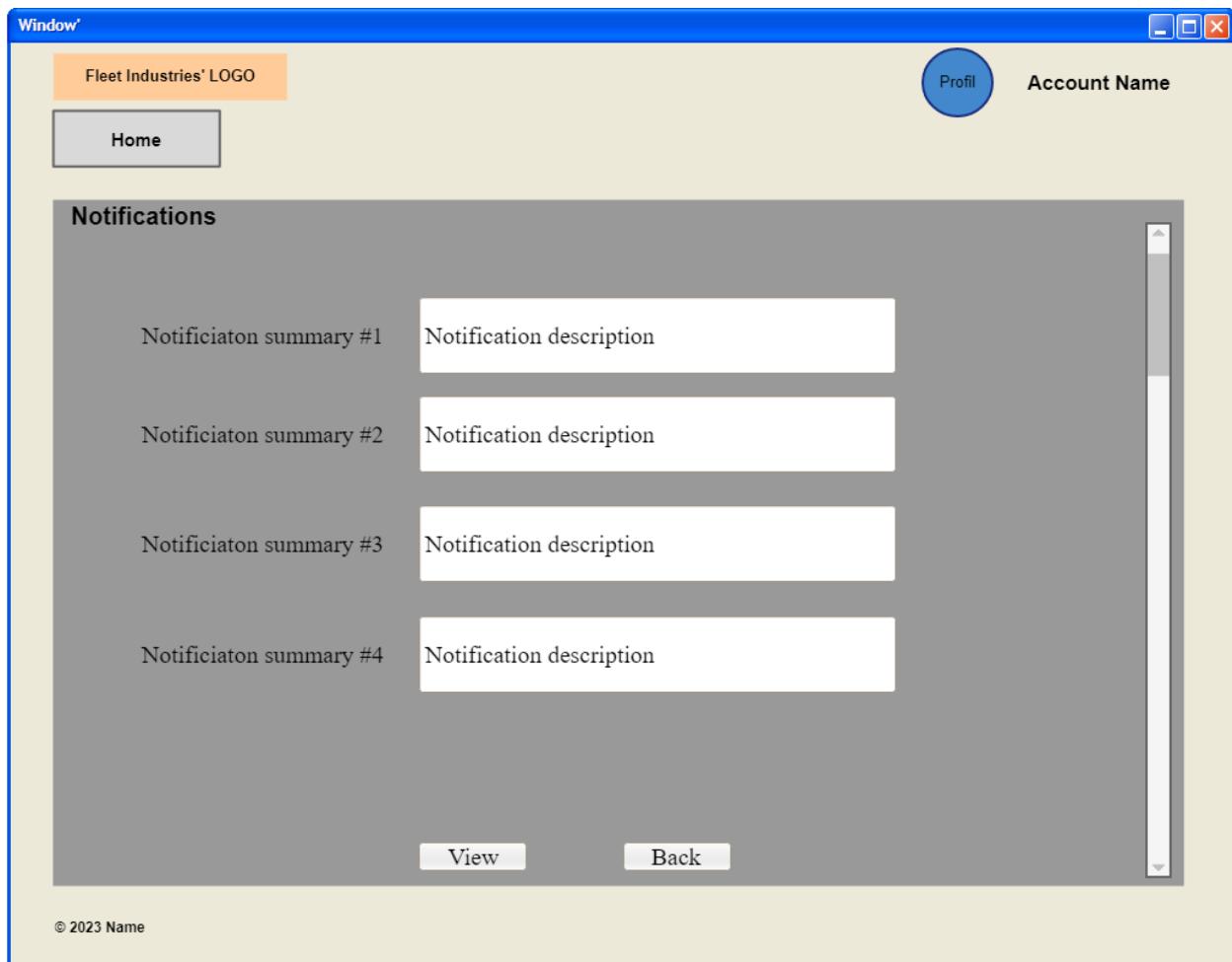
- Left Section (Grey Background):** Contains a "Welcome Back!" message, a "Reactivate" button, and a "Log Out" button.
- Right Section (Light Grey Background):** Contains a "Return" button and a message stating "This account cannot be reactivated. Contact customer support for more information".

At the top right of the main area, there's a red error message: "Unable to Connect to Server". To the right of the main area, there's a sidebar with the following fields and buttons:

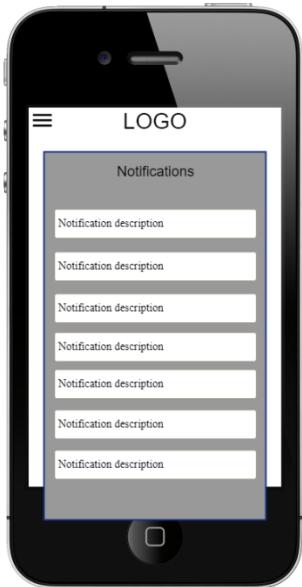
- "New Password" input field
- "Confirm Password" input field
- A red error message: "Password error"
- A "Update and Continue" button

At the bottom left of the main area, there's a copyright notice: "© 2023 Name".

## Receive Notifications Desktop

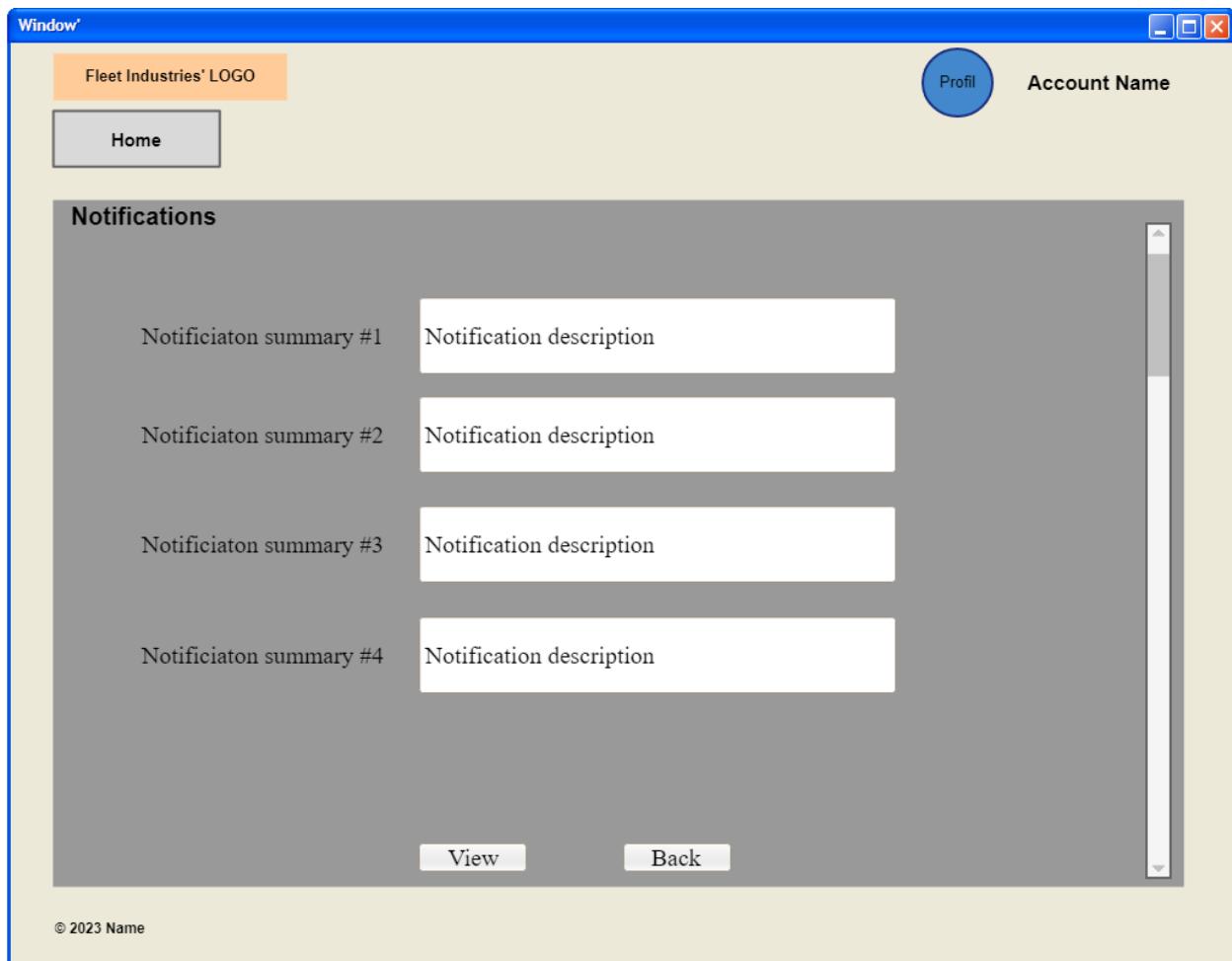


## Receive Notifications Mobile



© 2023 Name

## Receive Notifications Web



## Request Help Desktop

Window

Fleet Industries' LOGO

Home

Account Name

Profil

**Customer Service**

Phone  
123 - 456 - 7890

Email  
fleet@industries.com

Address  
123 Main St

Hours of Operation  
5:00 AM - 12:00 PM

View Back

© 2023 Name

## Request Help Mobile



© 2023 Name

## Request Help Web

Window

Fleet Industries' LOGO

Home

Profil Account Name

**Customer Service**

Phone	123 - 456 - 7890
Email	fleet@industries.com
Address	123 Main St
Hours of Operation	5:00 AM - 12:00 PM

View Back

© 2023 Name

517 | Page

## Reset Password Desktop

Window

Fleet Industries' LOGO

Home Login

Log In

Confirm email to reset password

email Address

invalid email error

Reset via text

Customer Support

Send Reset Link

© 2023 Name

Before email link: After email link:

Confirm phone number to reset password

Phone

invalid phone number error

Reset via text

Customer Support

Send Reset Link

Enter a new password for your account

Password

Confirm Password

Password error

Confirm Reset

You should receive a message at phoneNoEmail.

Please follow the link to reset your password.

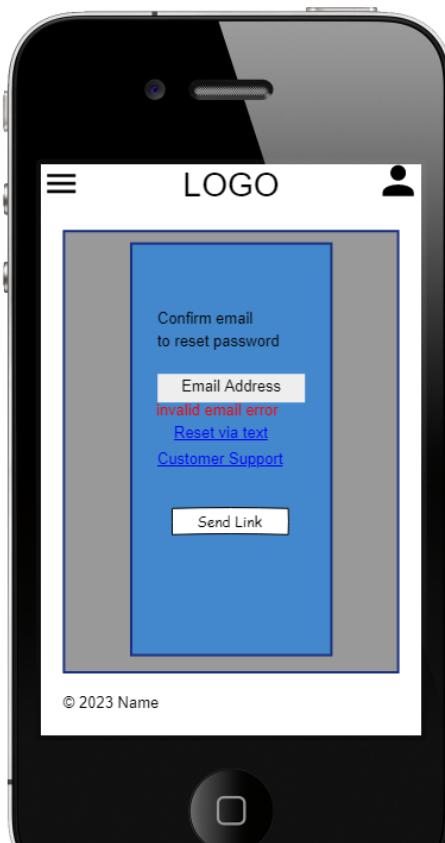
This password reset link has expired.

Go to Account login

To reset your password

The image displays a desktop application window titled 'Window'. At the top left is the 'Fleet Industries' logo. On the right side of the window are 'Login' and 'Home' buttons. Below these buttons is a large grey area labeled 'Log In'. Within this area, there is a smaller white dialog box titled 'Confirm email to reset password'. It contains a text input field for 'email Address' which has an 'invalid email error' message below it. There are three links: 'Reset via text', 'Customer Support', and a 'Send Reset Link' button. At the bottom of the window, there is a copyright notice: '© 2023 Name'. To the right of the window, there are two panels. The first panel, labeled 'Before email link:', shows a form for entering a phone number to reset the password. It includes fields for 'Phone' (with an 'invalid phone number error' message), 'Reset via text', 'Customer Support', and a 'Send Reset Link' button. The second panel, labeled 'After email link:', shows a form for entering a new password. It includes fields for 'Password', 'Confirm Password', and a 'Confirm Reset' button. Below these panels, there is explanatory text: 'You should receive a message at phoneNoEmail.' followed by 'Please follow the link to reset your password.' The final panel on the right states 'This password reset link has expired.', followed by 'Go to Account login' and 'To reset your password'.

## Reset Password Mobile



Before email link:

Confirm phone to reset password

Phone No.  
*invalid email error*

[Reset via email](#)

[Customer Support](#)

[Send Link](#)

After email/text link:

Enter a new password for your account:

Password

Confirm Password

*Password error*

[Confirm Reset](#)

You should receive a message at: \_\_\_\_\_

Please follow the instructions to reset your password.

This password reset link has expired.

Go to [Account login](#)

To reset your password

## Reset Password Web

Chrome

Fleet Industries' Logo Home > Products > Pencil Login Register Now!

Home

**Log In**

Confirm email to reset password

email Address  invalid email error [Reset via text](#) [Customer Support](#) [Send Reset Link](#)

© 2023 Name

**Before email link:**

Confirm phone number to reset password

Phone  invalid phone number error [Reset via email](#) [Customer Support](#) [Send Reset Link](#)

**After email/text link:**

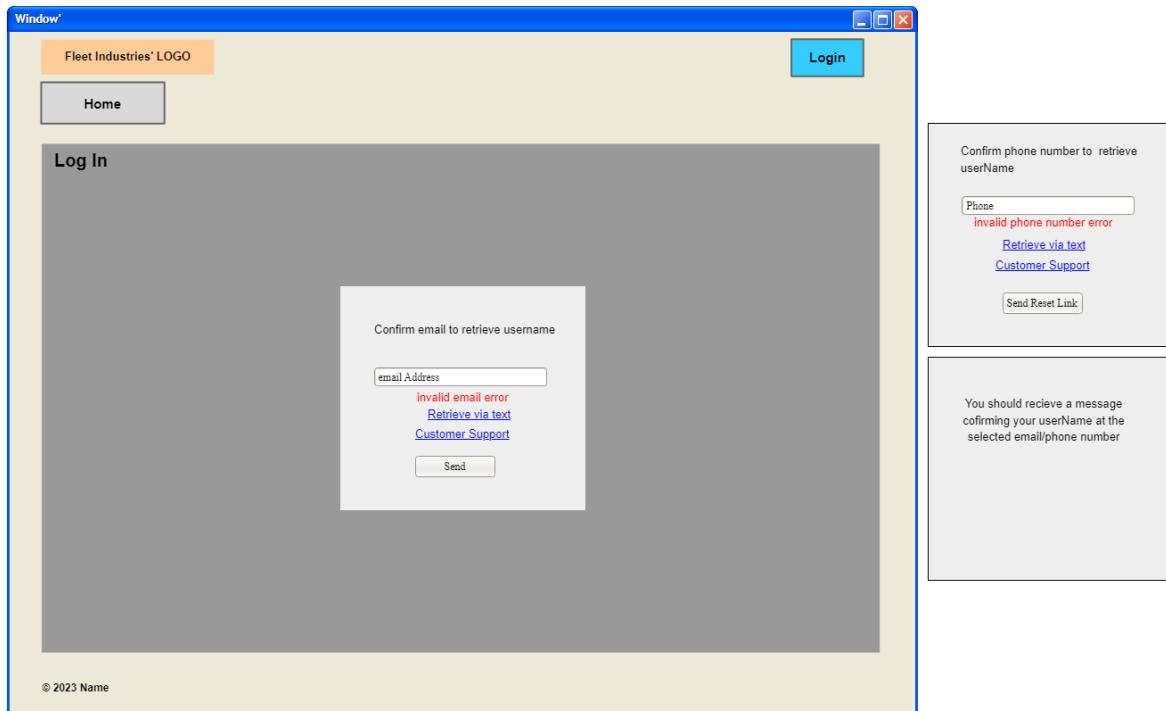
Enter a new password for your account

Password  Confirm Password  Password error [Confirm Reset](#)

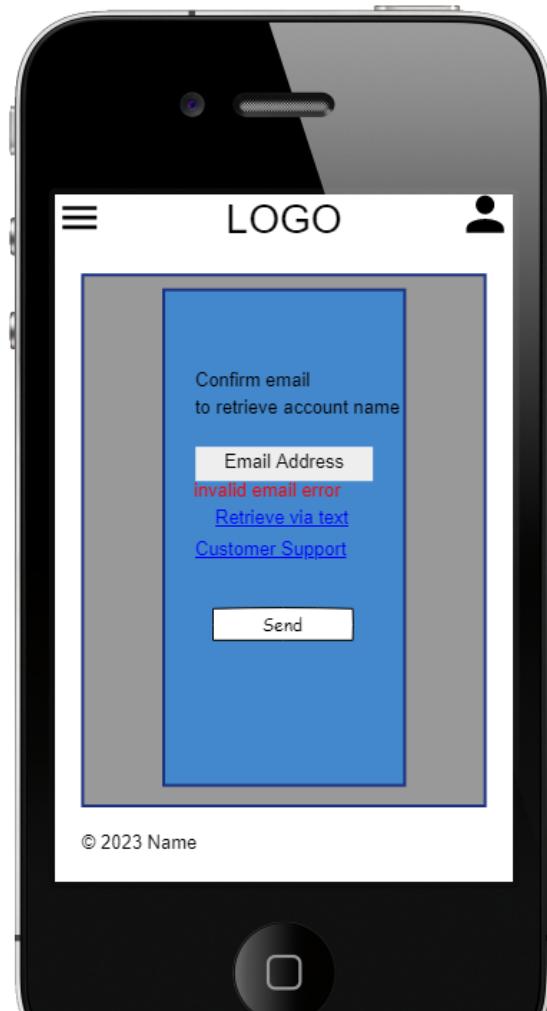
This password reset link has expired.  
Go to [Account Login](#)  
To reset your password.

You should receive a message at:  
phoneNo/Email  
Please follow the link to reset your password.

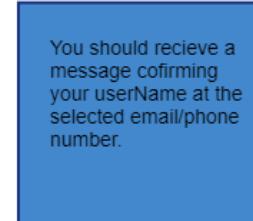
## Retrieve Username Desktop



## Retrieve Username Mobile



### Before email link:



## Retrieve Username Web

Chrome

Fleet Industries' Logo Home > Products > Pencil Login Register Now!

Home

Log In

Confirm email to retrieve username

invalid email error

[Retrieve via email](#) [Retrieve via text](#) [Customer Support](#)

© 2023 Name

Before email link:

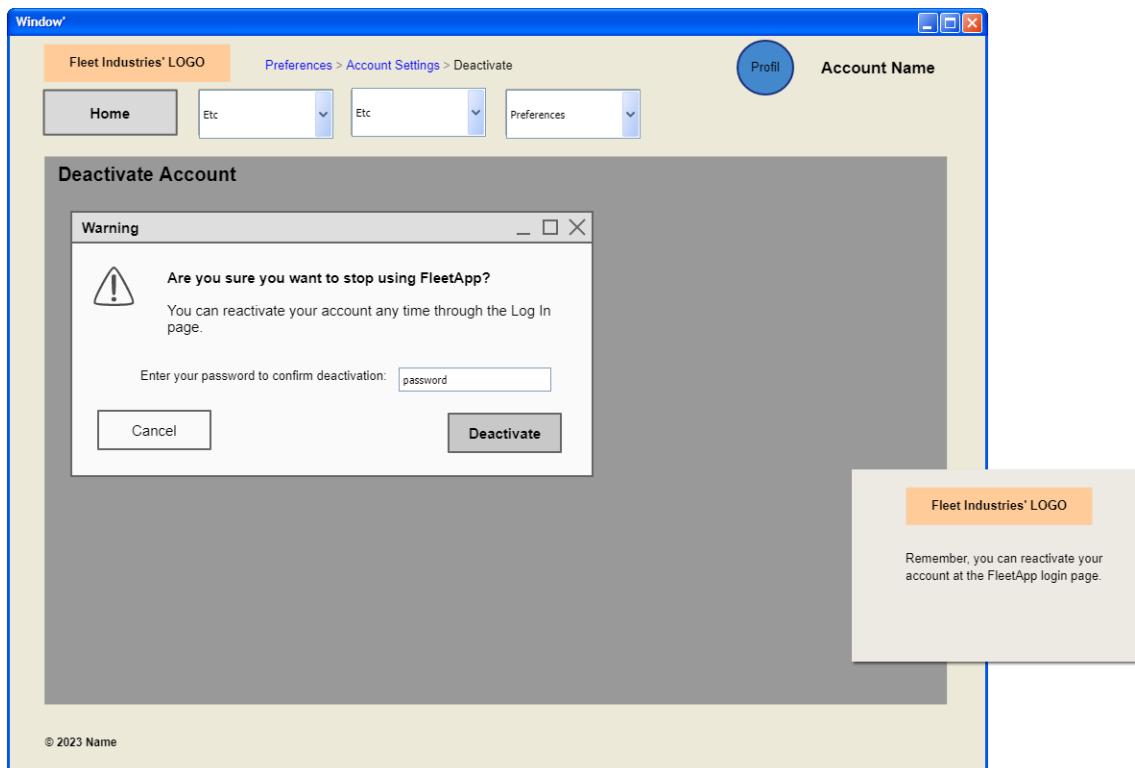
Confirm phone number to retrieve username

invalid phone number error

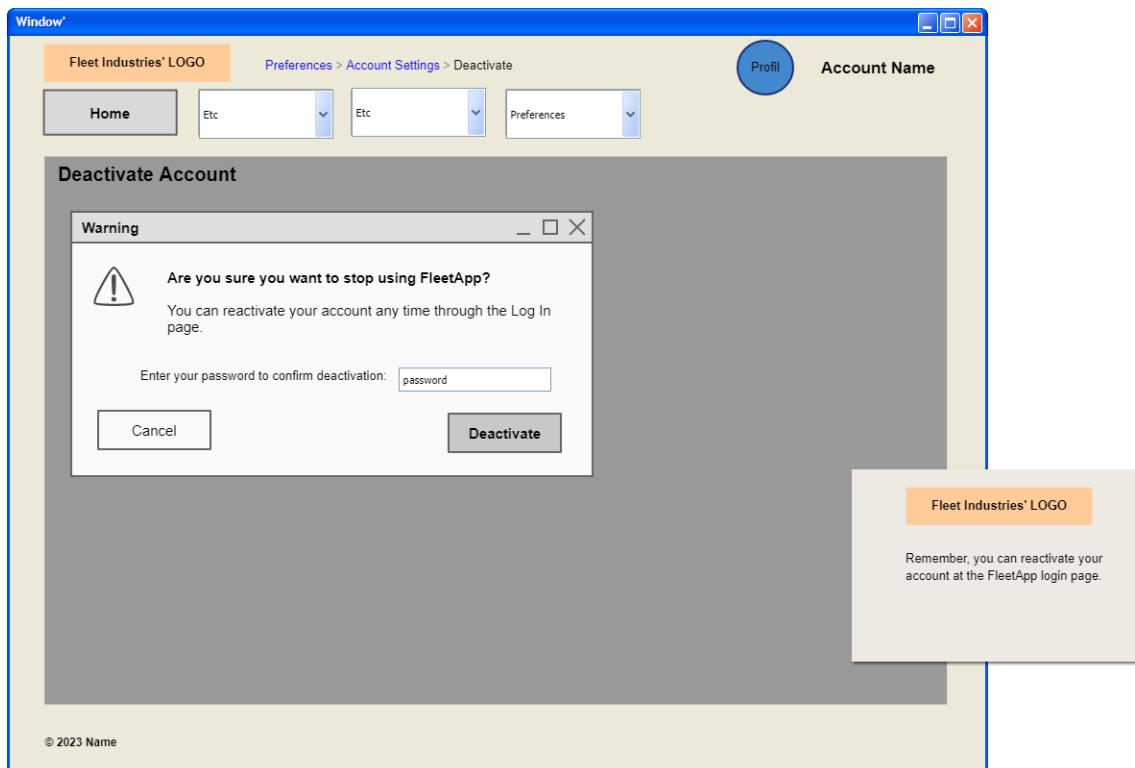
[Retrieve via email](#) [Customer Support](#)

You should receive a message confirming your userName at the selected email/phone number

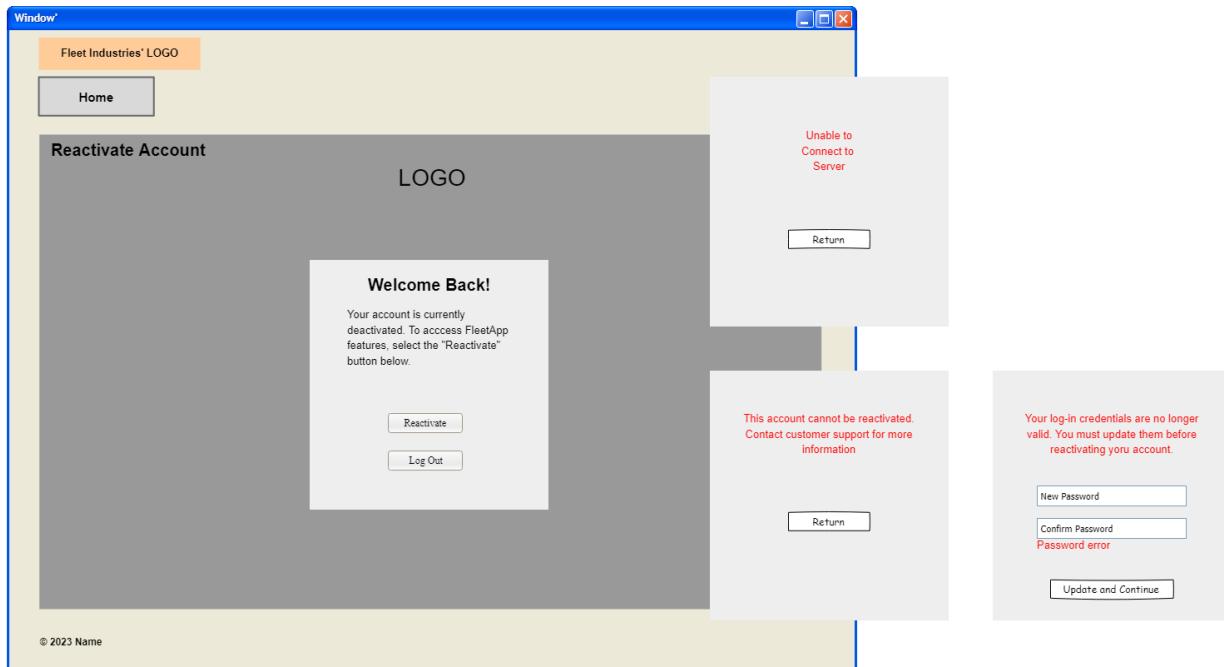
## UI-User.01.05.DESK.NOTUSED



## UI-User.01.06.DESK.NOTUSED



## UI-User.01.07.DESK.NOTUSED



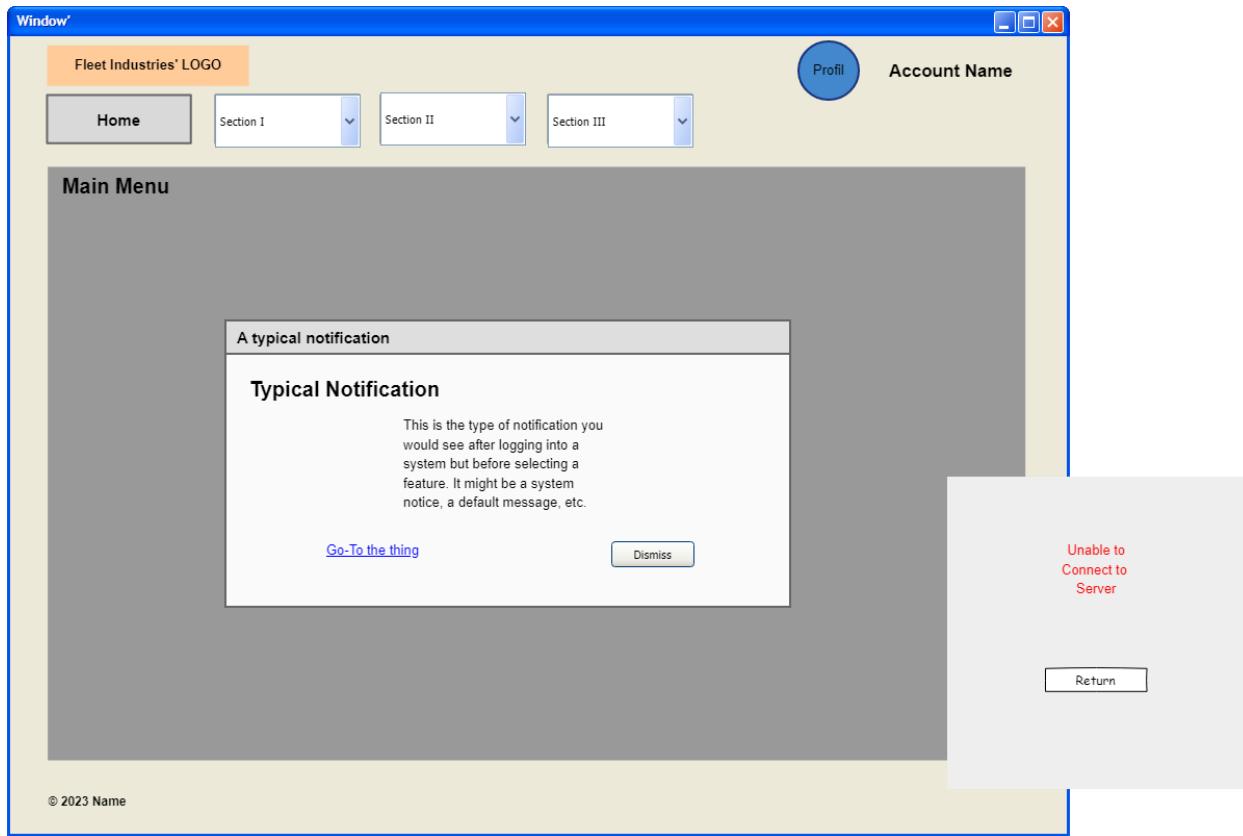
## UI-User.22.01.MOB



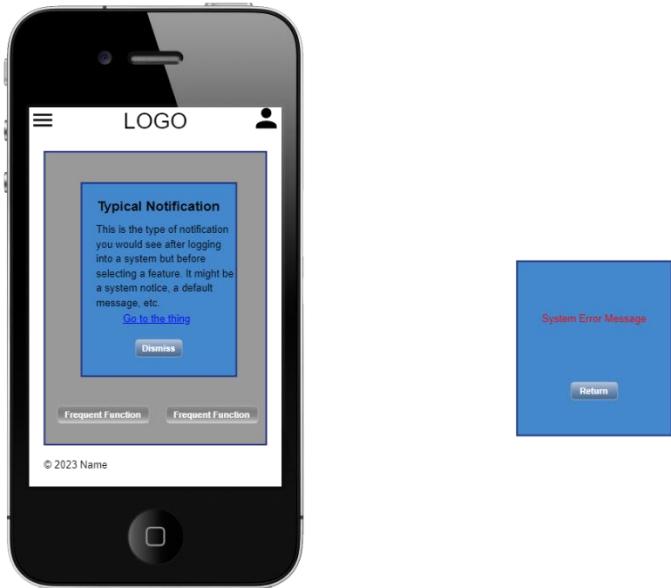
## UI-User.22.02.MOB



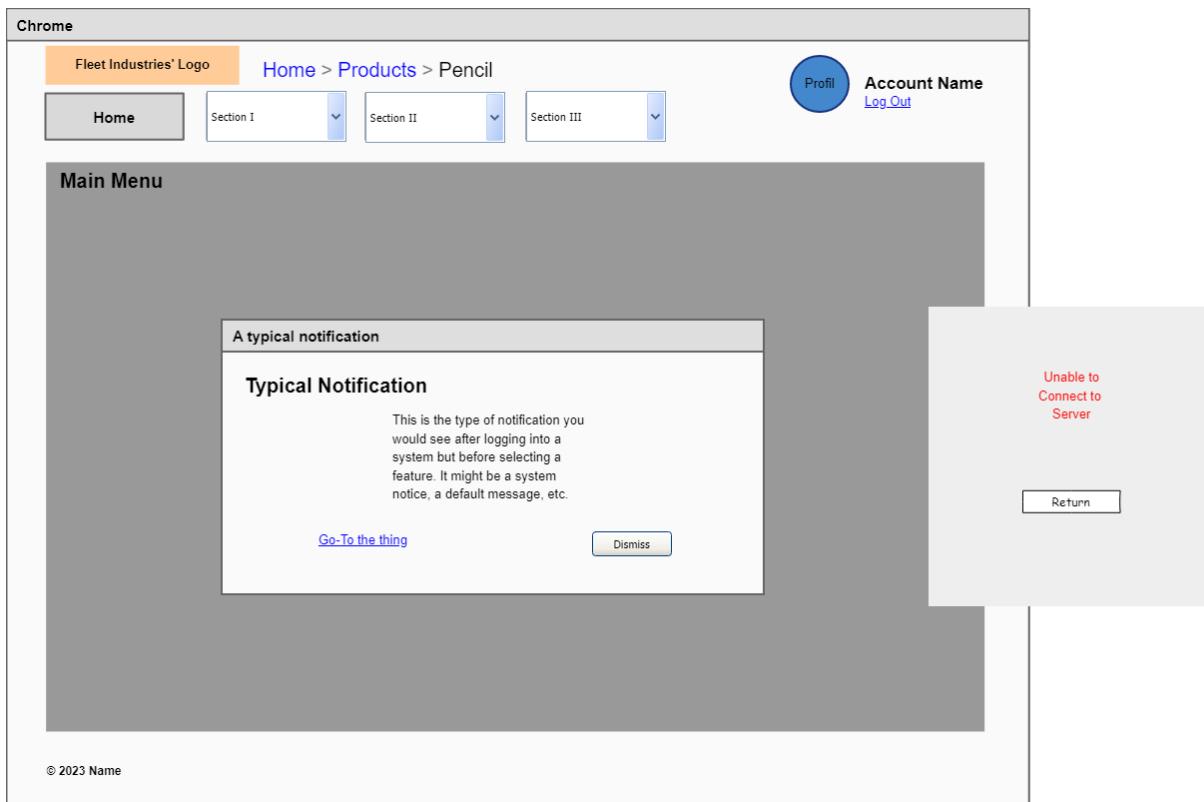
## User Main Menu Desktop



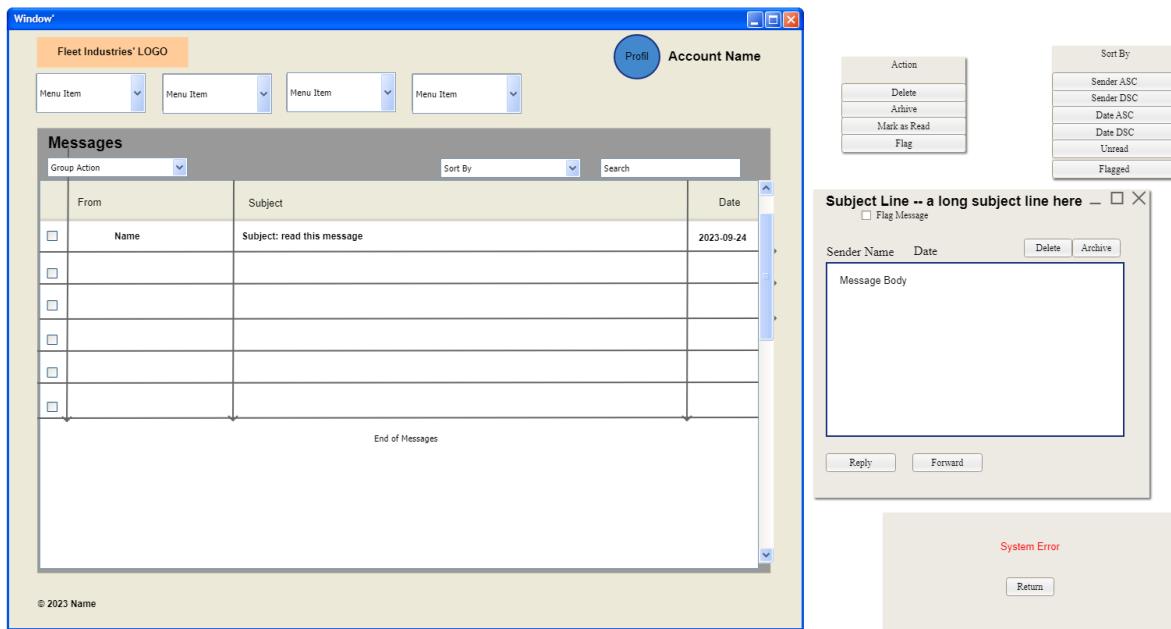
## User Main Menu Mobile



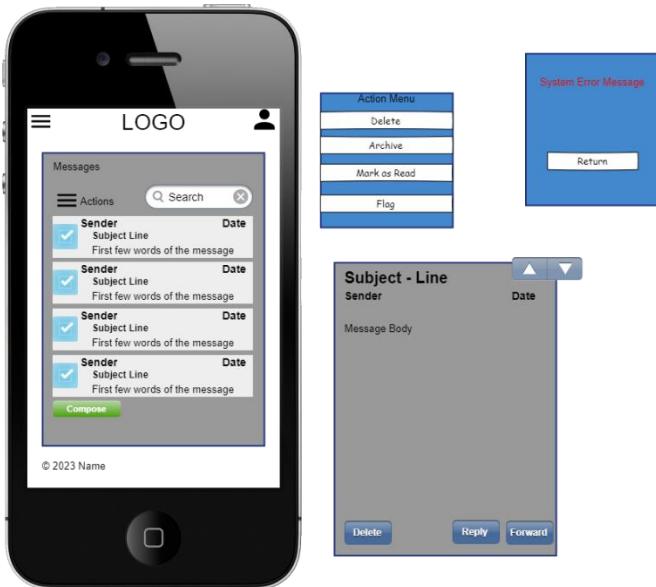
## User Main Menu Web



## View Messages Desktop



## View Messages Mobile



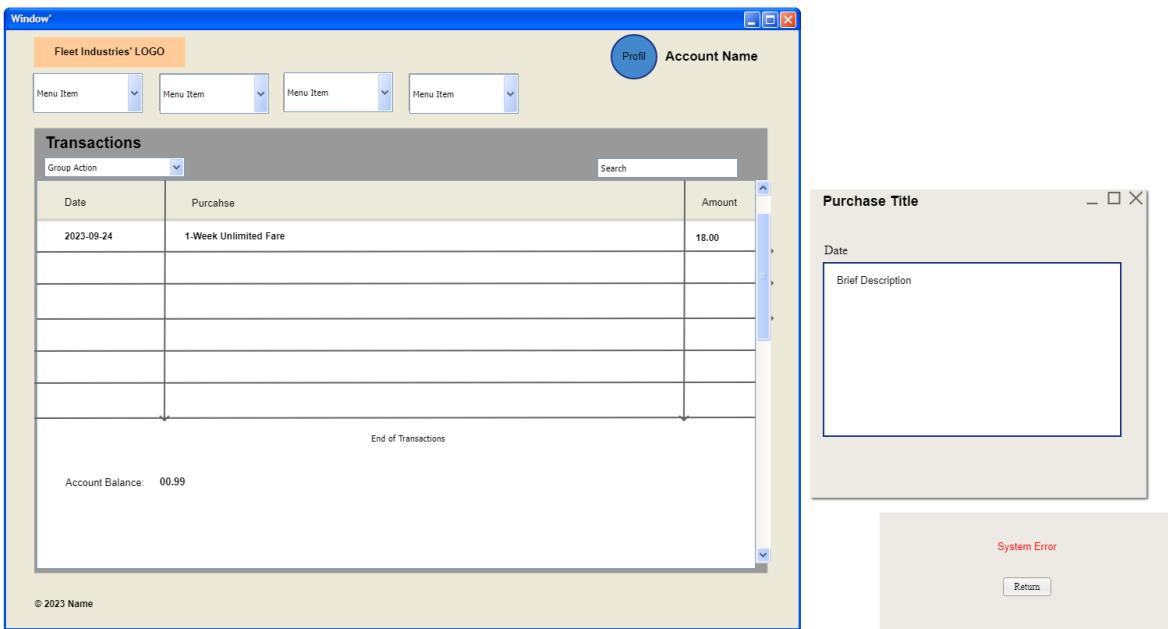
## View Messages Web

The screenshot displays a web-based application for viewing messages. At the top, there is a header bar with the text "Fleet Industries' Logo", "Home > Products > Pencil", "Profile", and "Account Name". Below the header, there is a navigation menu with four items: "Menu Item", "Menu Item", "Menu Item", and "Menu Item". To the right of the menu is a vertical toolbar with the following options: "Action" (Delete, Archive, Mark as Read, Flag), "Sort By" (Sender ASC, Sender DSC, Date ASC, Date DSC, Unread, Flagged), and a search bar labeled "Search" and a "Sort By" dropdown.

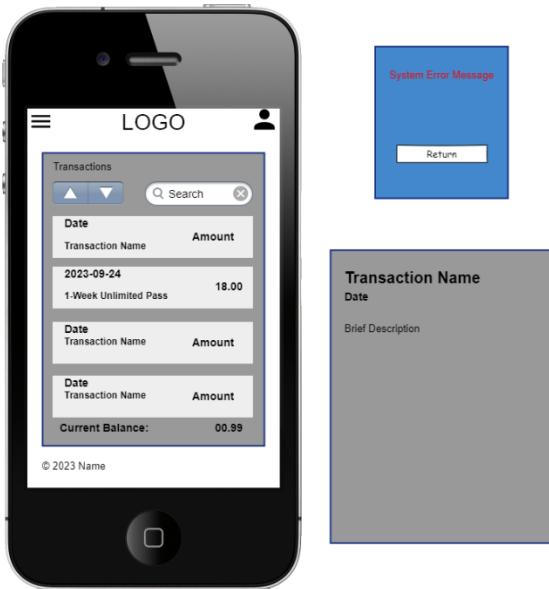
The main area is titled "Messages" and contains a list of messages. The first message in the list has the subject "Subject: read this message" and a date of "2023-09-24". Below the message list are "Reply" and "Forward" buttons. To the right of the message list is a detailed view of the selected message, showing the "Sender Name" and "Date" (2023-09-24), the "Message Body" (which is currently empty), and "Delete" and "Archive" buttons. A red error message "System Error" is displayed at the bottom of this panel, with a "Return" button below it.

At the bottom left of the main message list area, there is a copyright notice: "© 2023 Name".

## View Purchases Desktop



## View Purchases Mobile



## View Purchases Web

The screenshot shows a web application interface for viewing purchases. At the top, there is a header bar with the text "Fleet Industries' Logo", "Home > Products > Pencil", and a "Profile" button. Below the header, there are four dropdown menus labeled "Menu Item".

The main area is titled "Transactions" and displays a table of purchase history. The columns are "Date", "Purchase", and "Amount". A single row is visible, showing "2023-09-24", "Weekly Unlimited Fare", and "\$15.00". There are also "Sort" buttons and a "Search" input field.

Below the table, there is a section for "Current Balance" which is currently empty. A link "Previous period" is located at the bottom right of this section.

At the bottom left, there is a copyright notice: "© 2023 Name".

To the right of the main window, there is a modal dialog titled "Transaction Name". It contains fields for "Date" and "Brief Description".

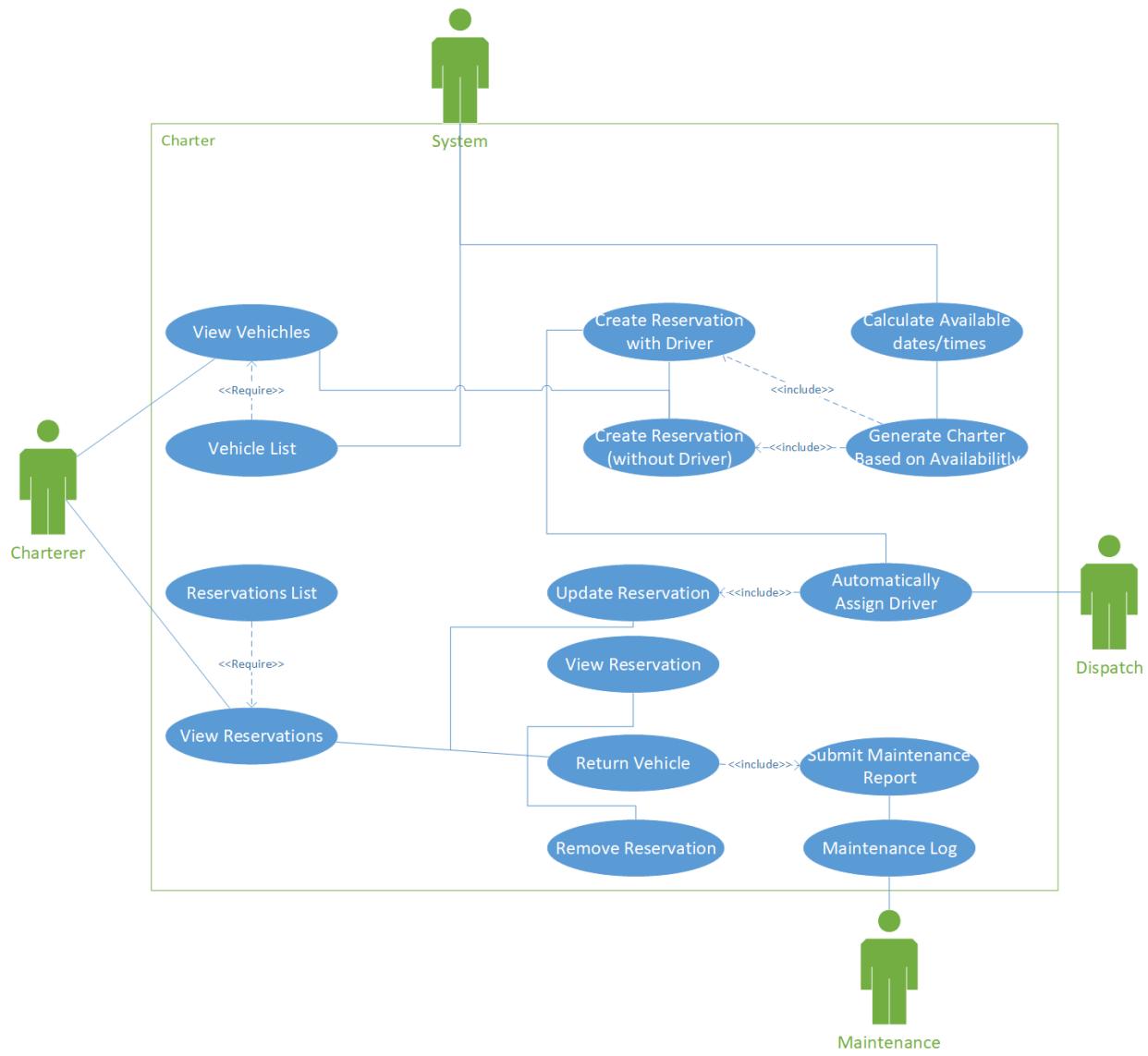
Below the modal, there is a separate message box with the title "System Error" and a "Return" button.



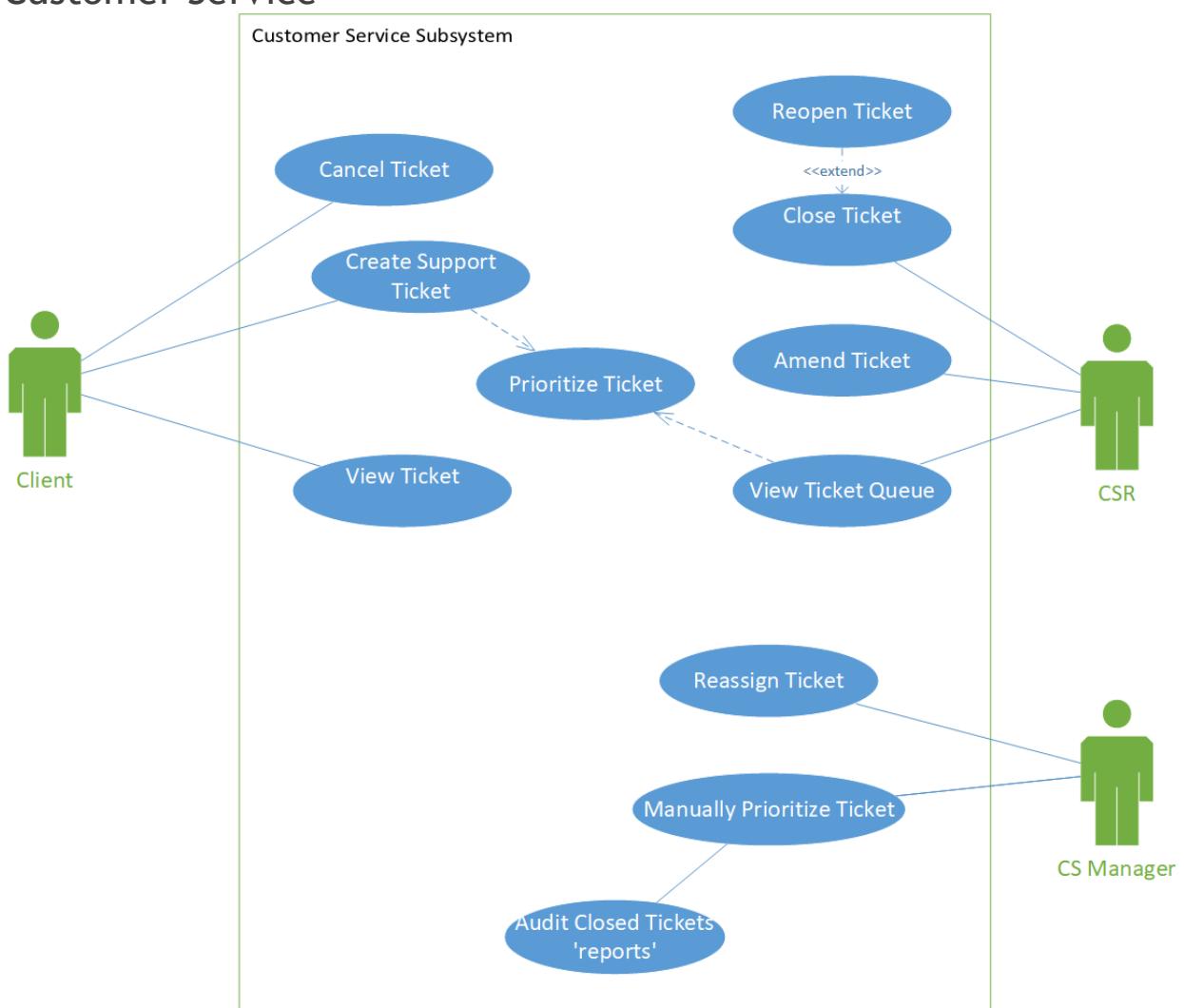
# Use Case Context and Subsystem Diagrams

The Use Cases can be organized Subsystems which describe groups of features which can be satisfied by a cohesive domain of system functionality. Subsystem diagrams describe the flow of Use Cases which connect User Stories to underlying system functionality. Subsystem diagrams will be used to identify functionality which can satisfy several User Stories, the interactions between areas of the system, and potential automations.

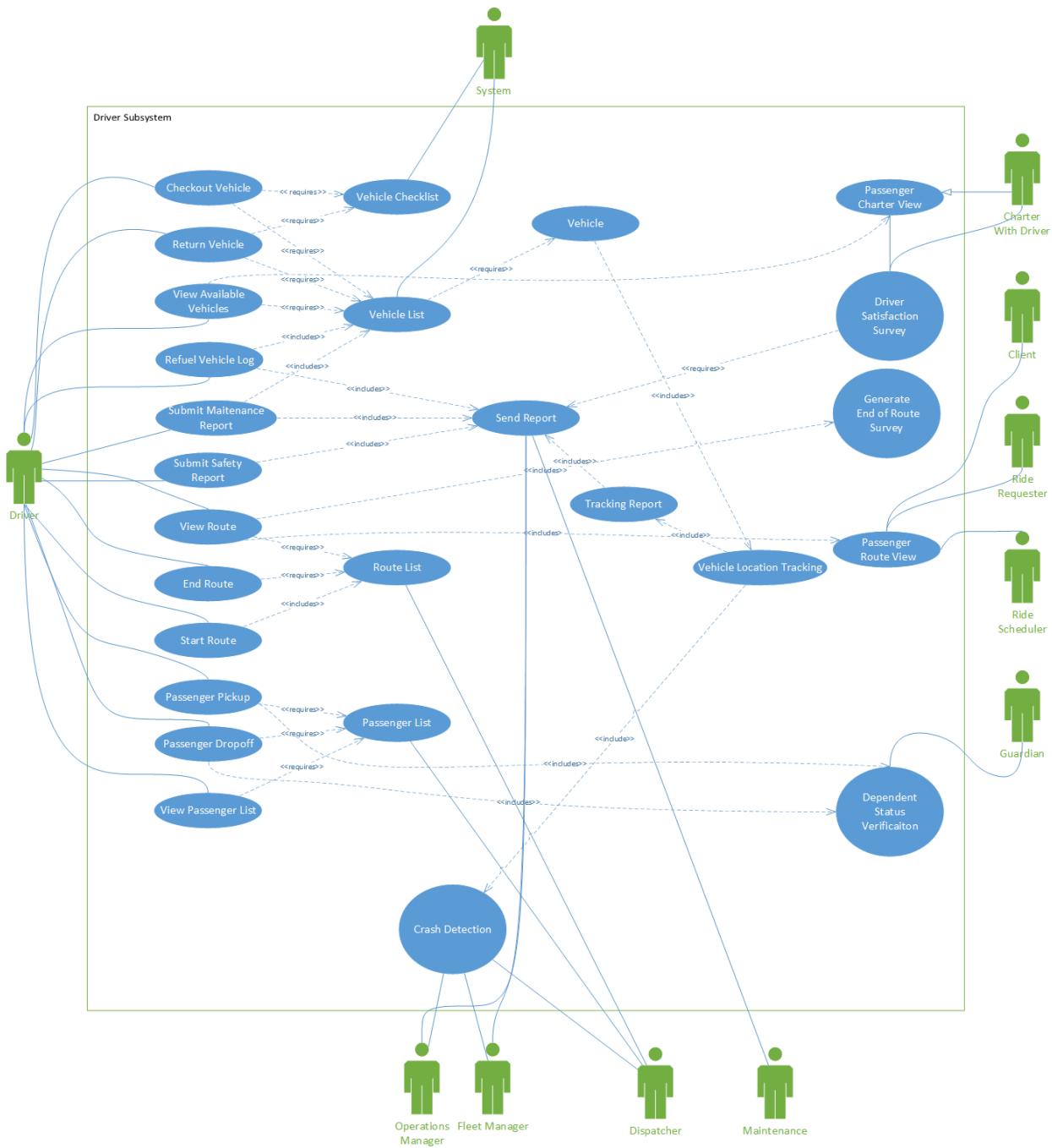
## Charter



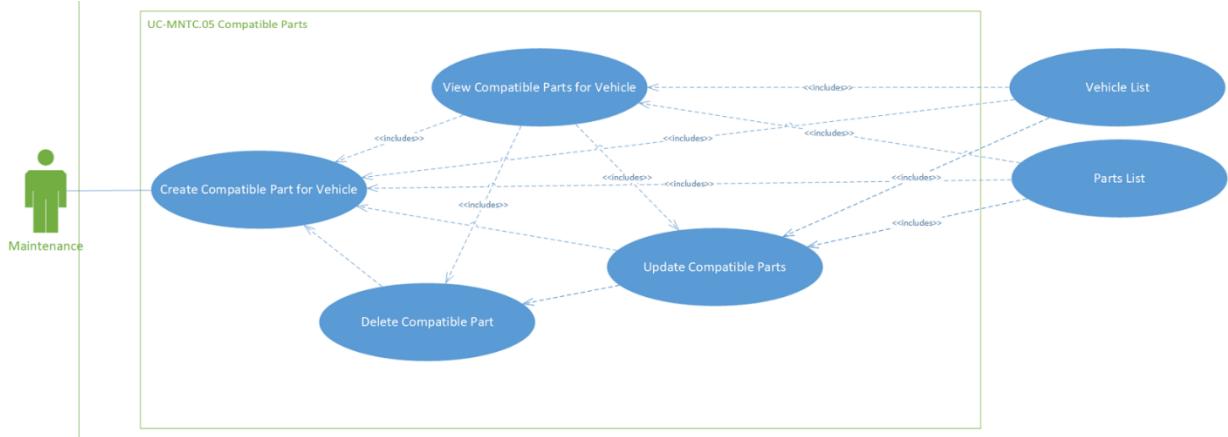
## Customer Service



# Driver



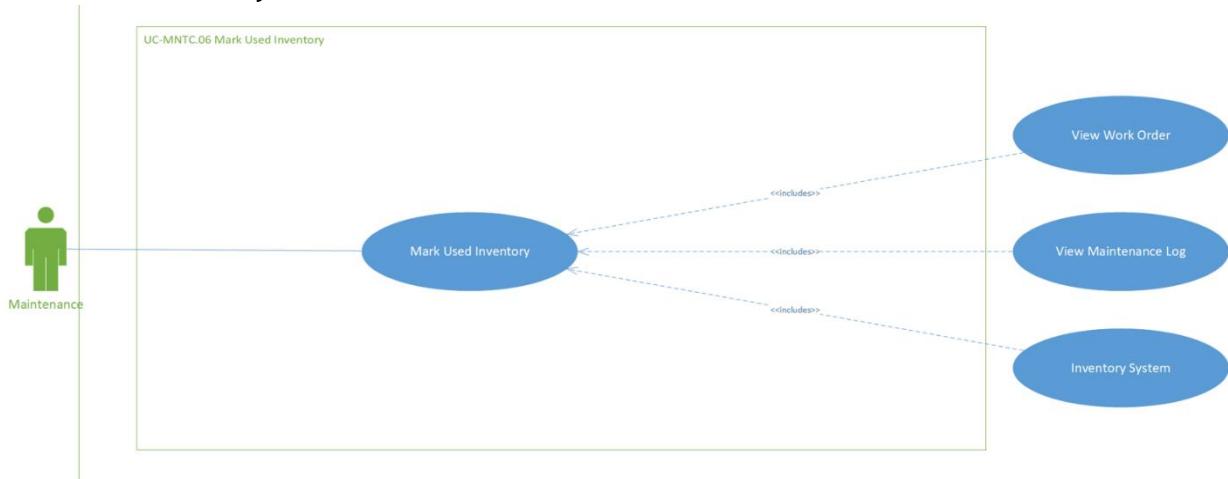
## Maintenance Compatible Parts



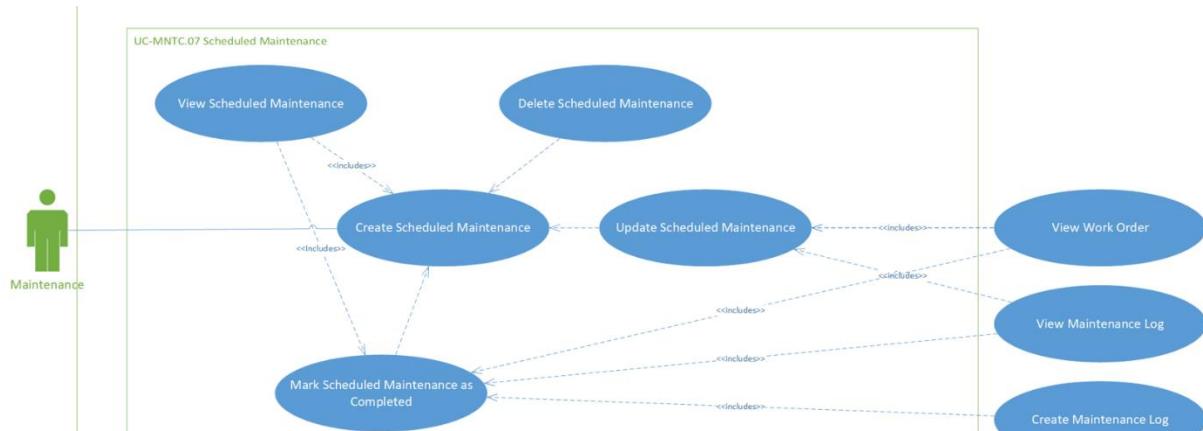
## Maintenance Log



## Mark Used Inventory



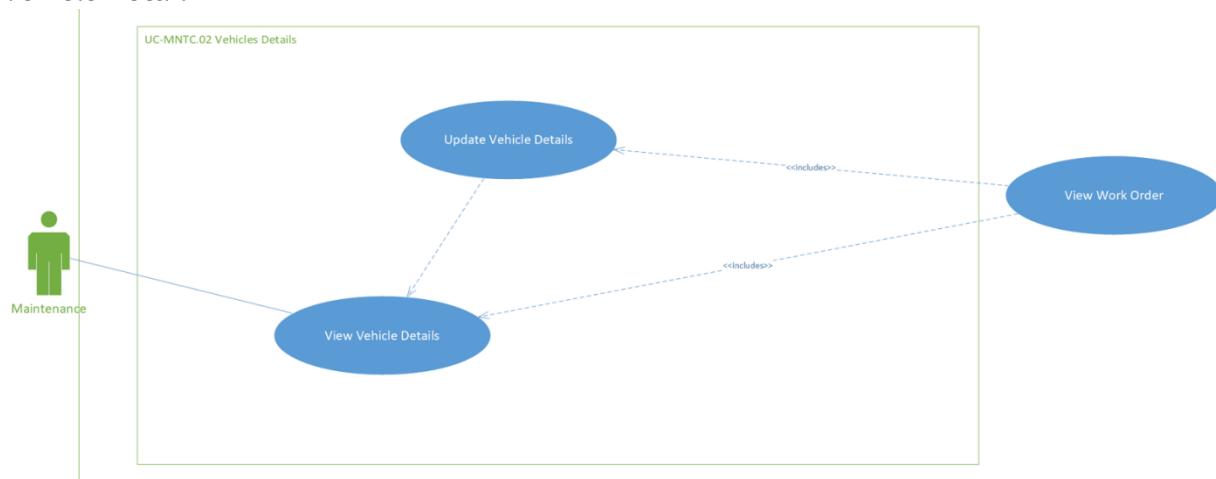
## Scheduled Maintenance



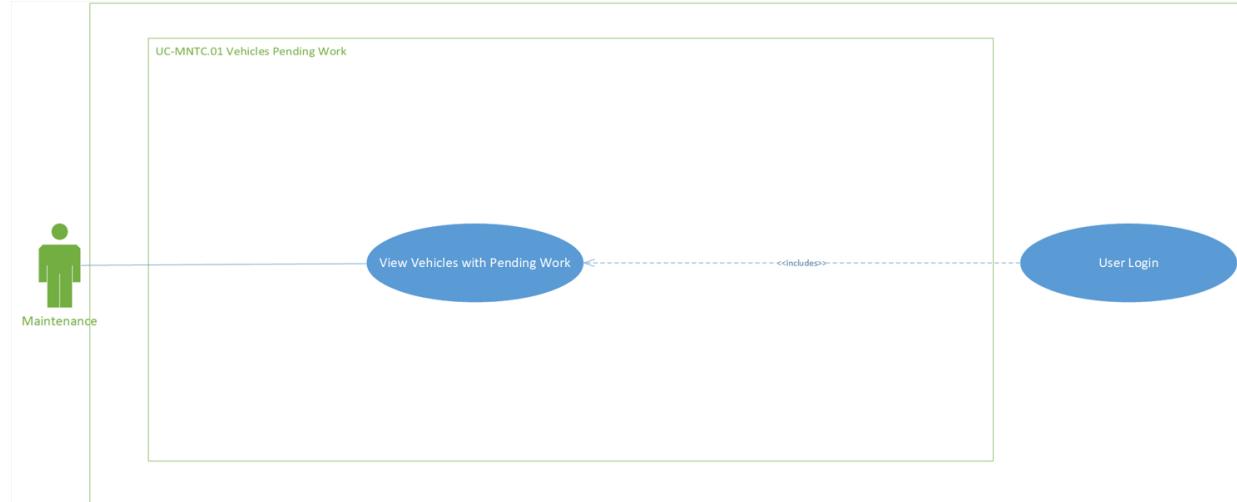
## Special Service Order



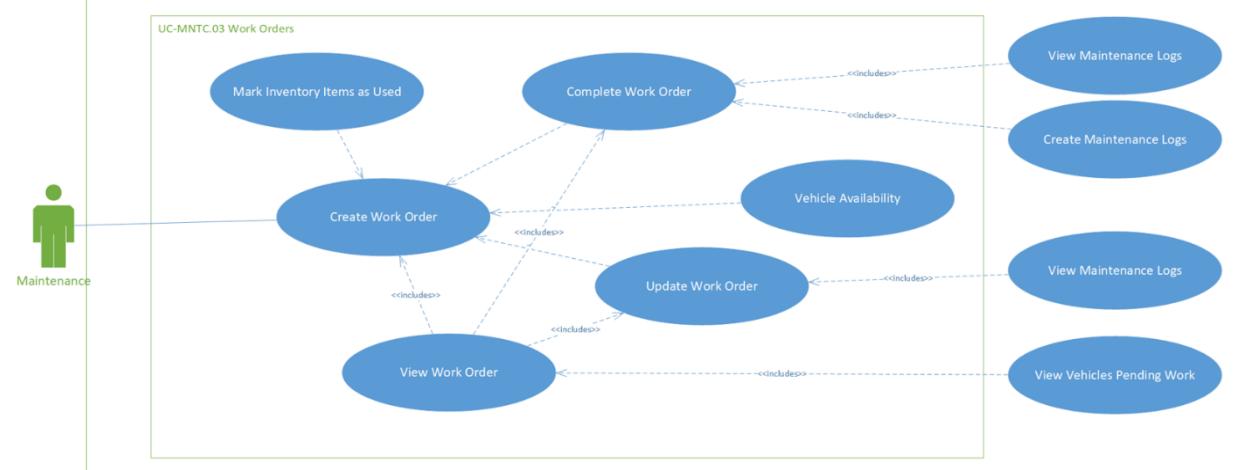
## Vehicle Detail



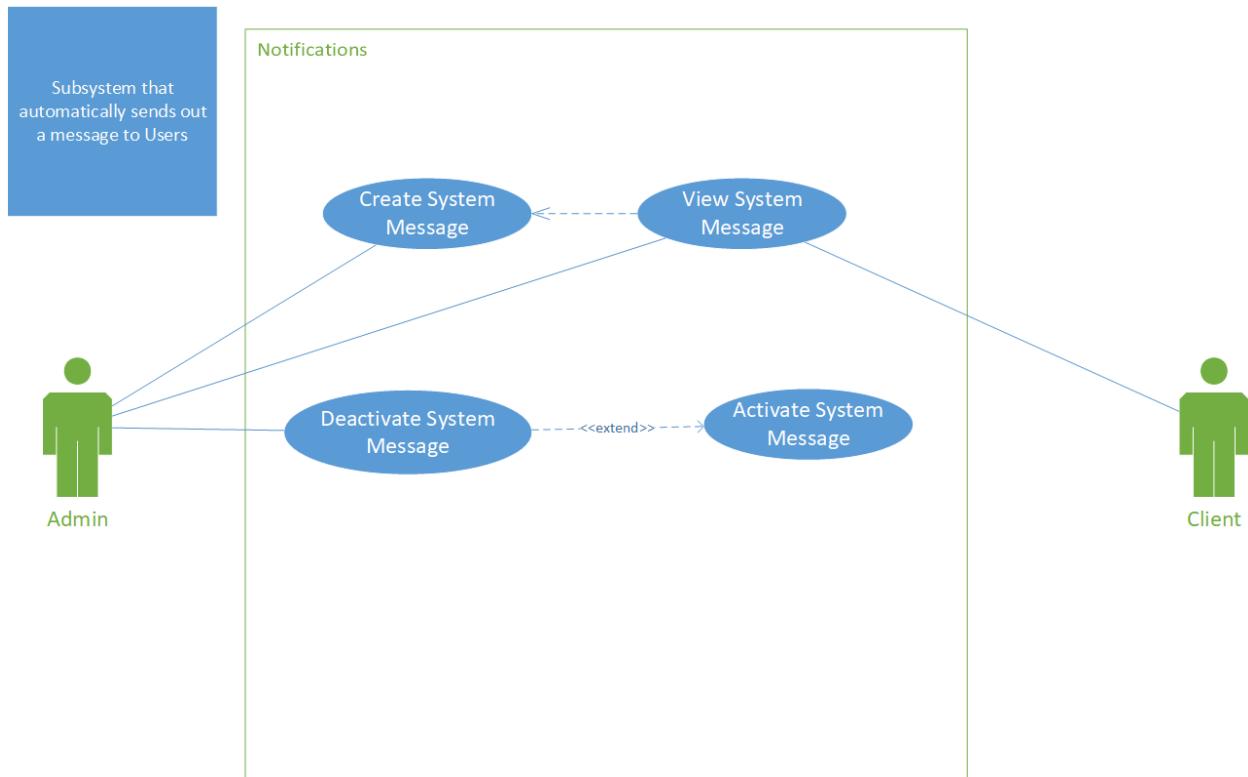
## Vehicle Pending Work



## Work Order

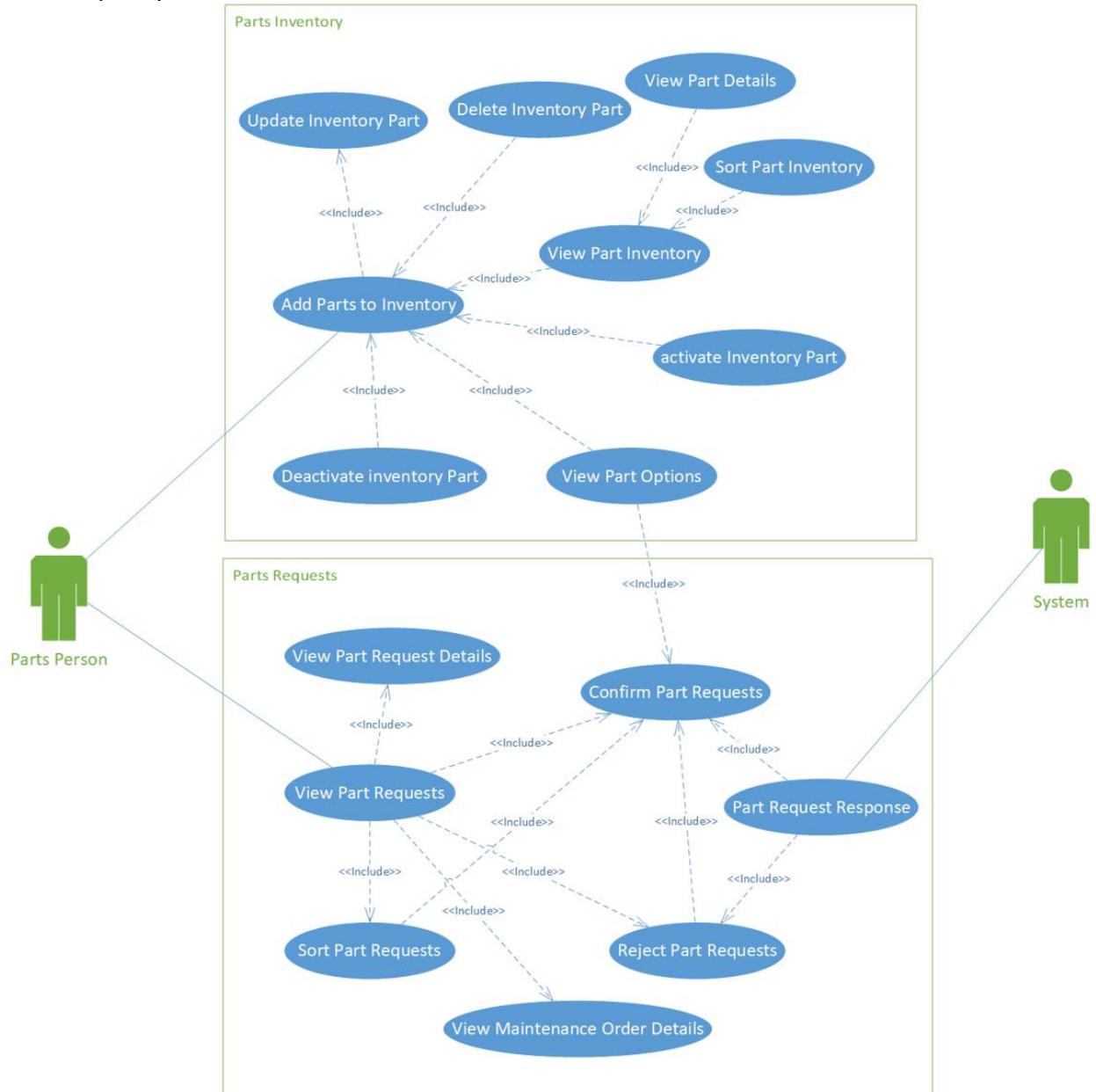


## Notification

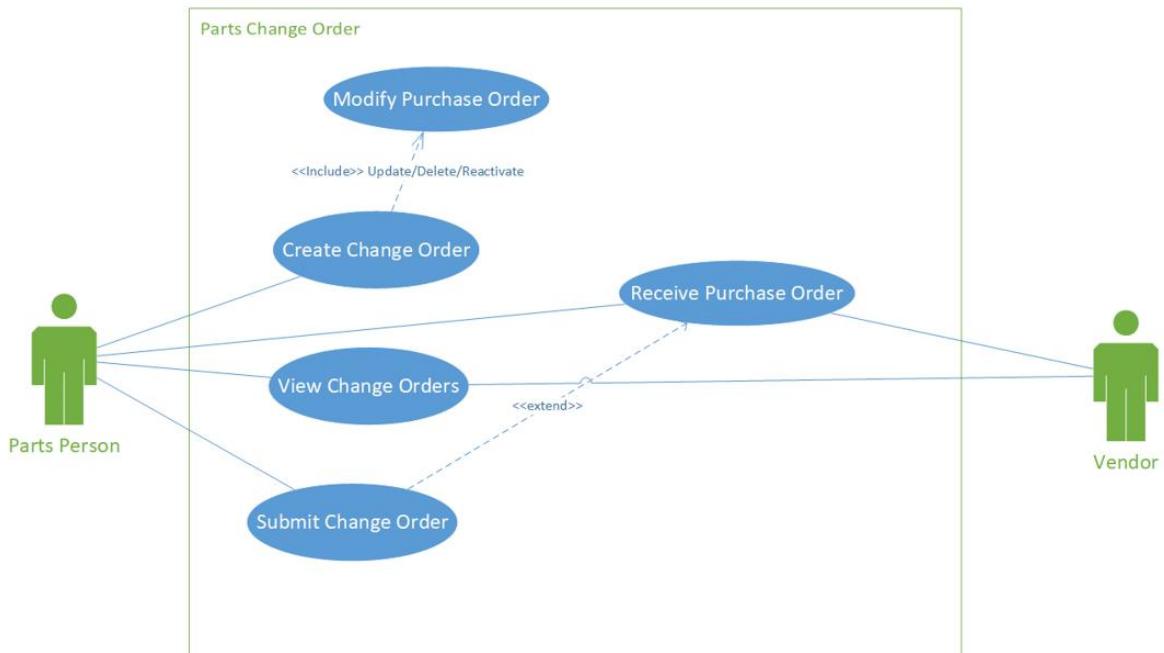


# Parts Inventory

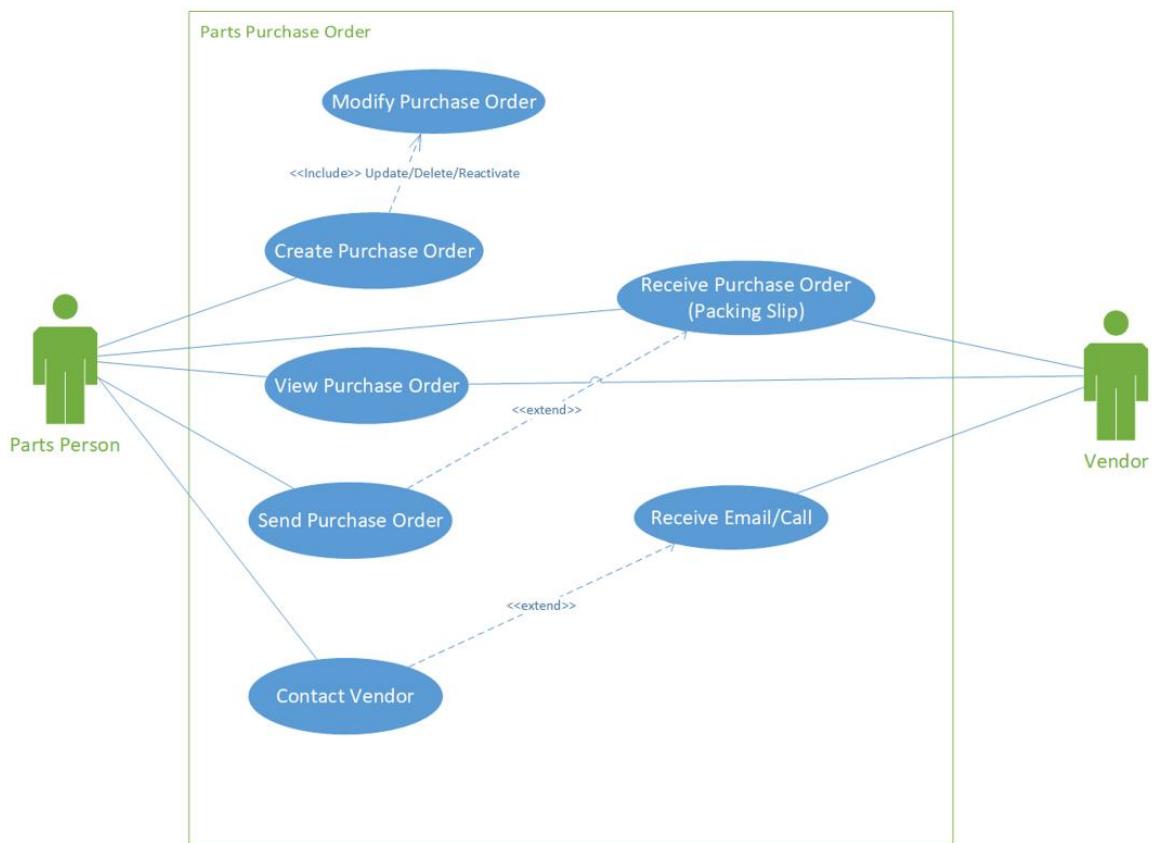
## Inventory Request



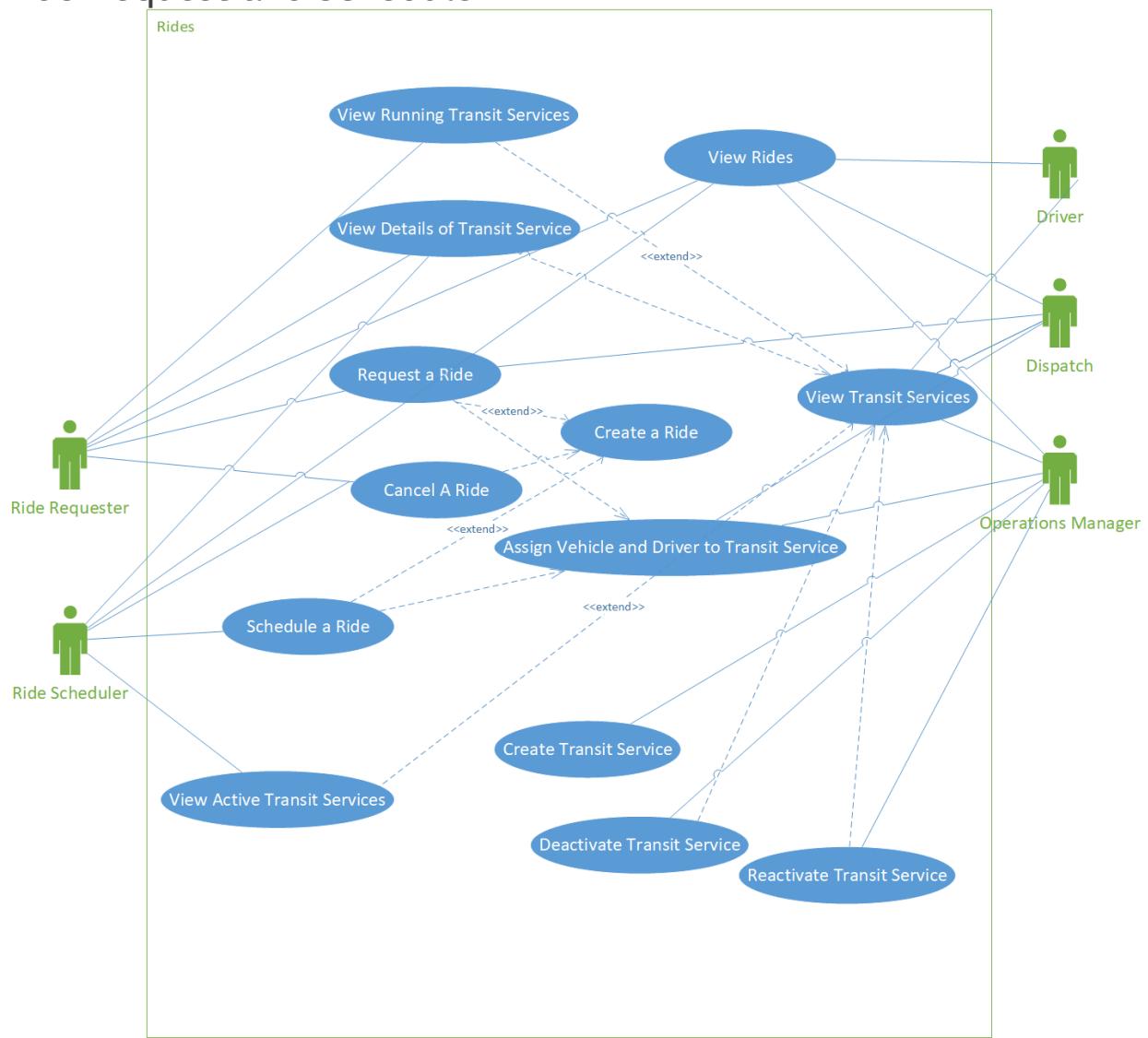
## Part Change Order



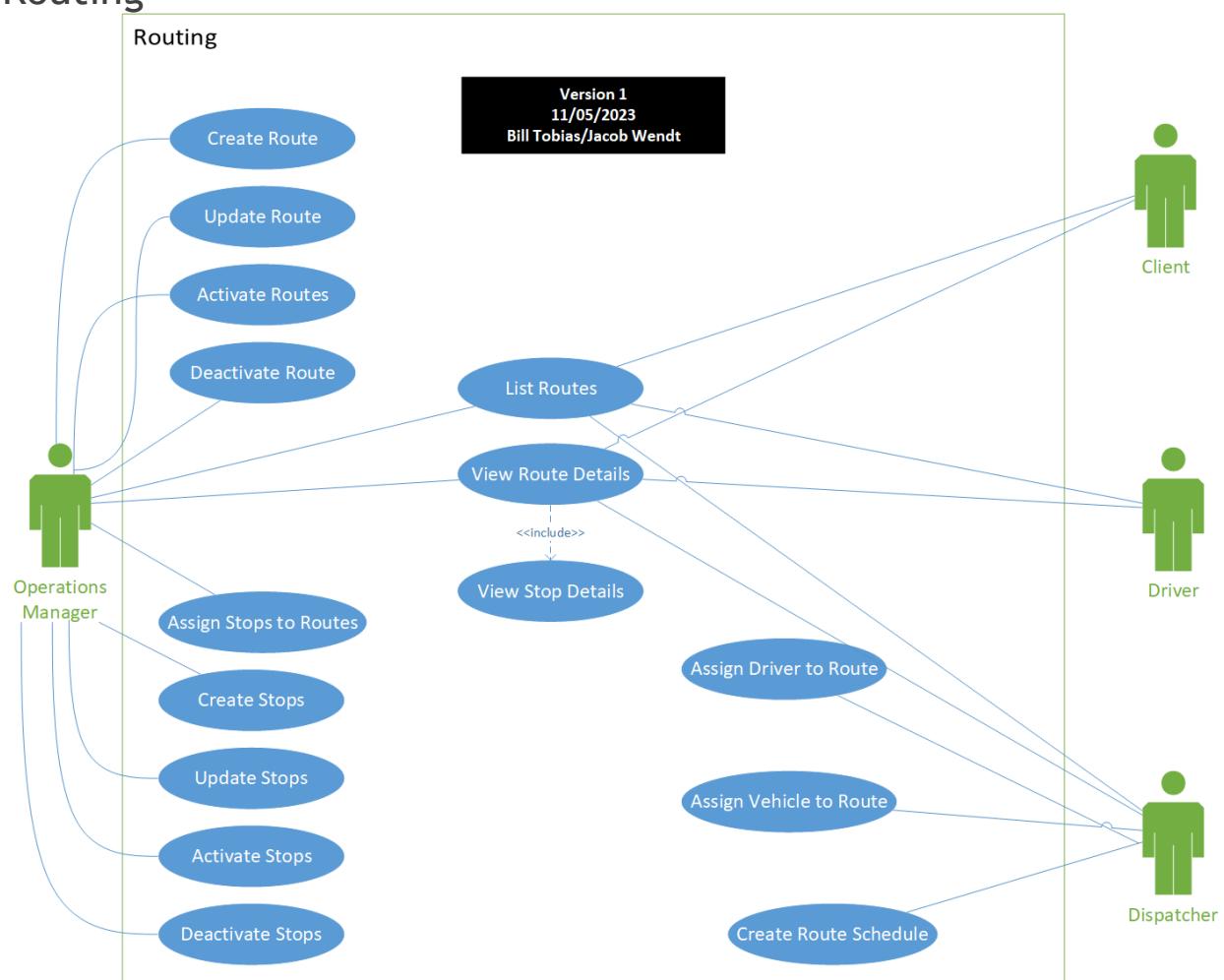
## Part Purchase Order



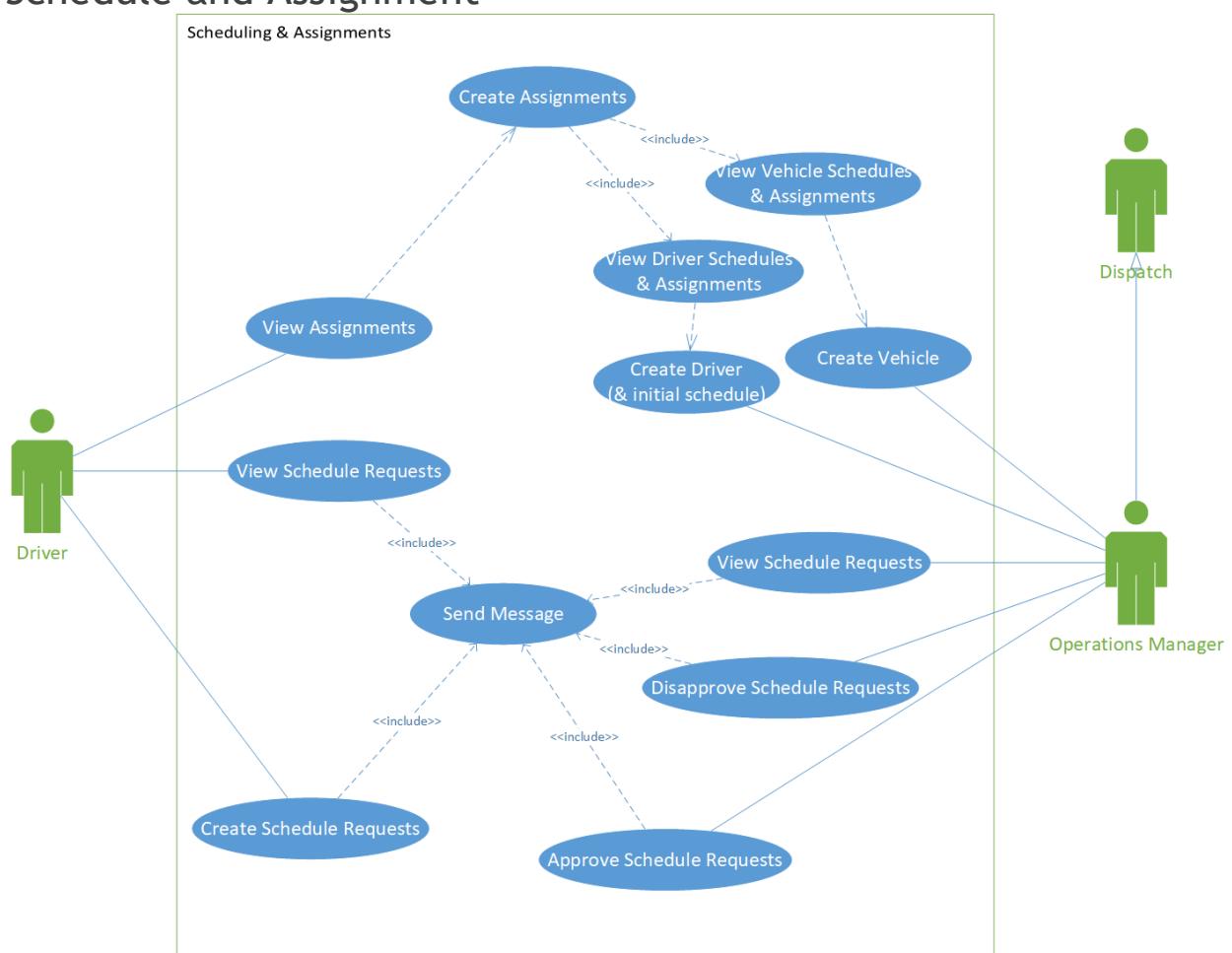
## Ride Request and Schedule



# Routing



## Schedule and Assignment



# Database Design

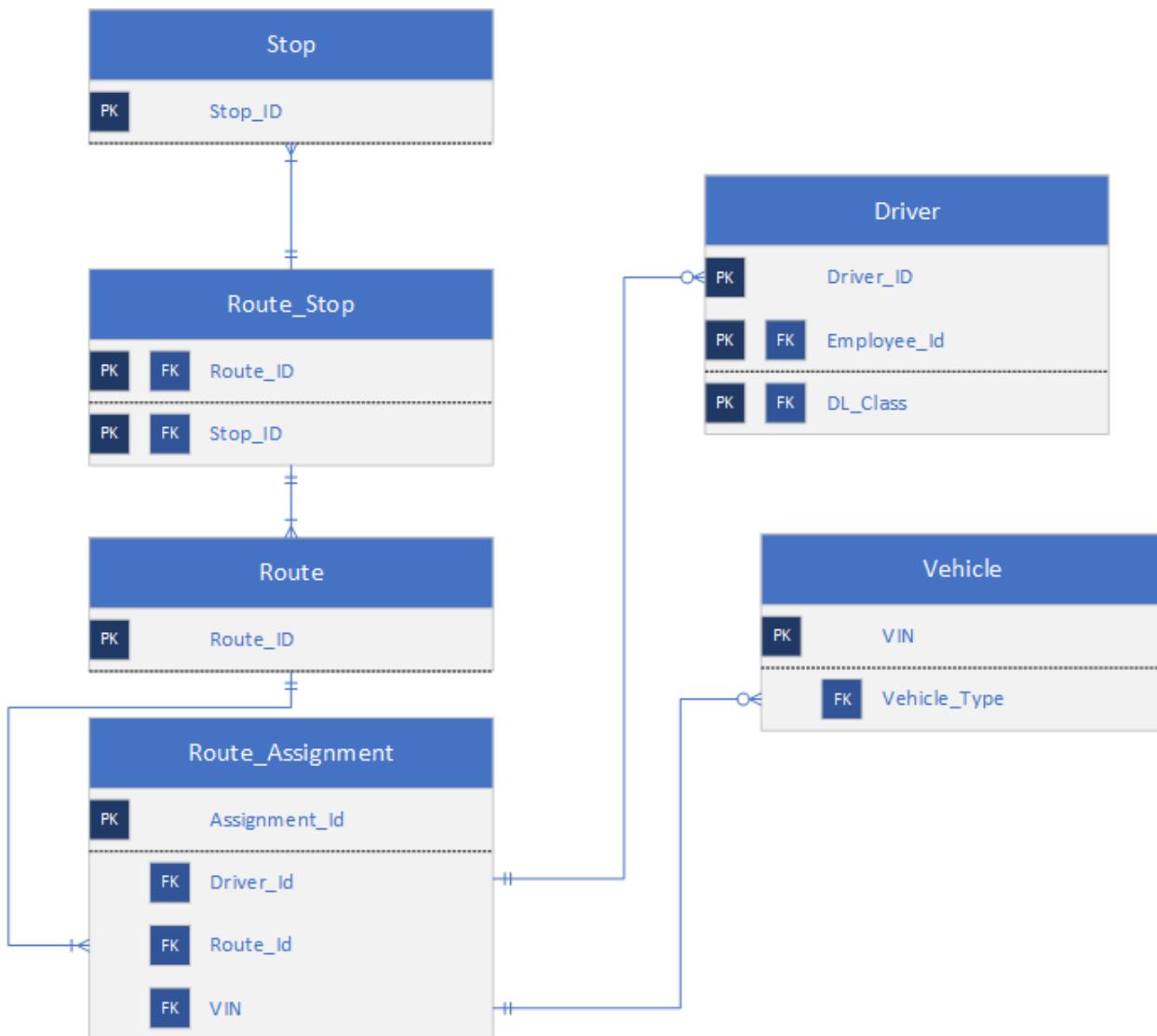
## Entity Relationship Diagrams

The Entity Relationship Diagram describes the relationships between the data objects which will comprise the system data. The objects in the Entity Relationship Diagram are based on the permanence of data which is required to satisfy Use Case functionality across instances of the application. The relationships between the objects are based on the data flows inferred from the Use Cases.

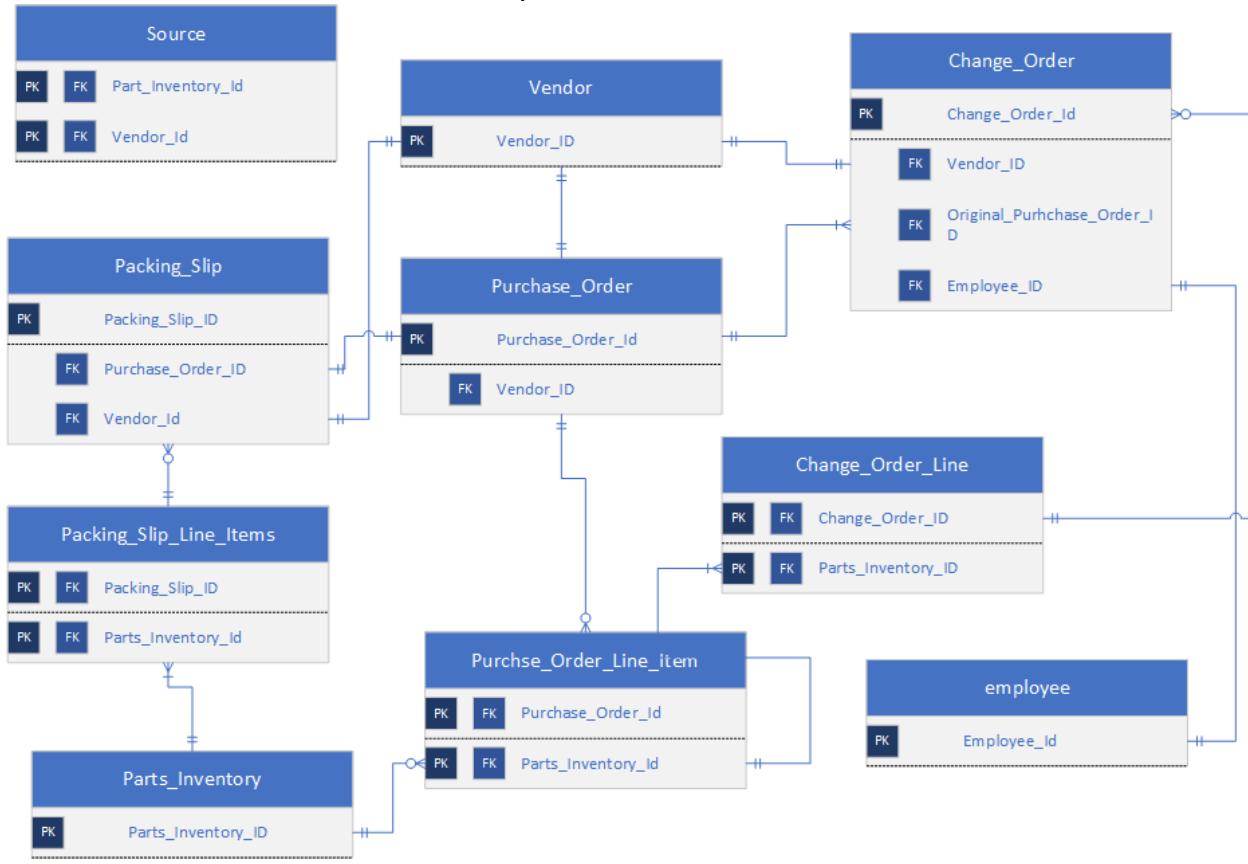
## Employee and Role Relationship



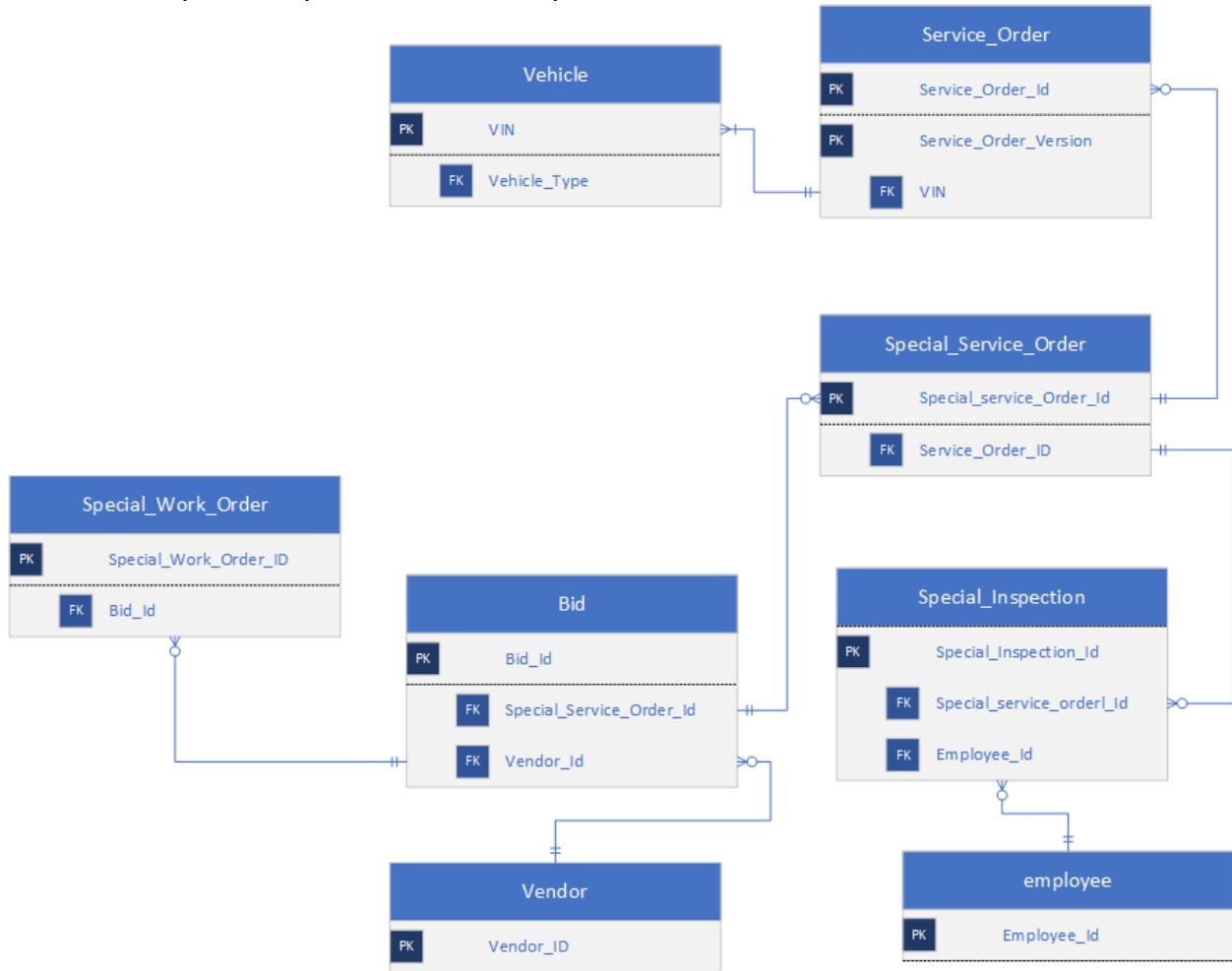
## Route and Stop Relationship



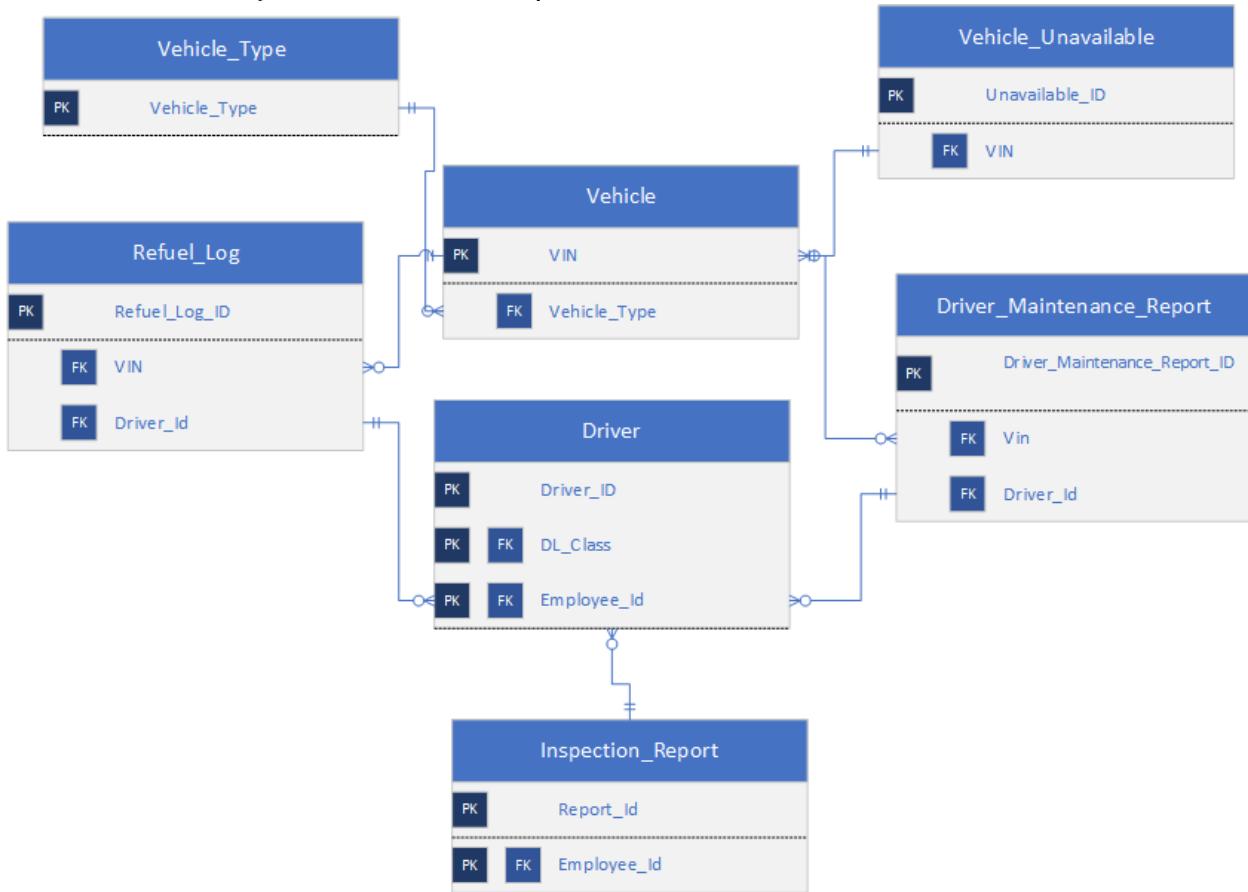
## Vendor and Purchase Order Relationships



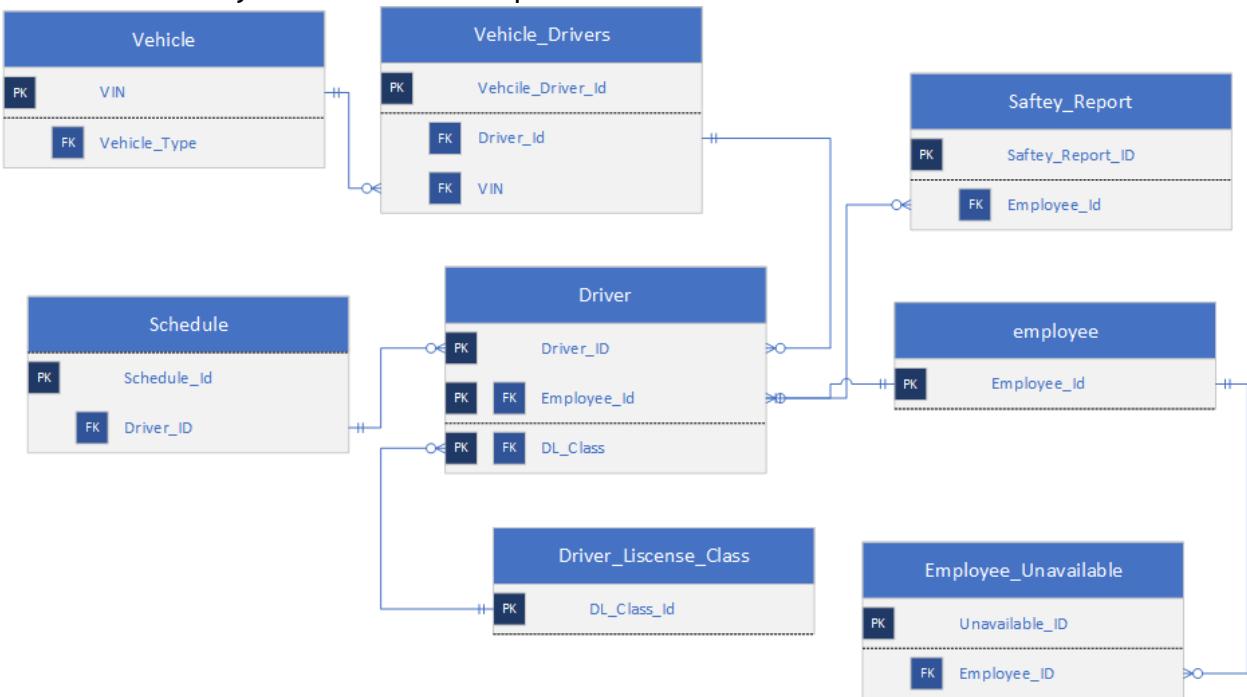
## Vehicle and Special Inspection Relationships



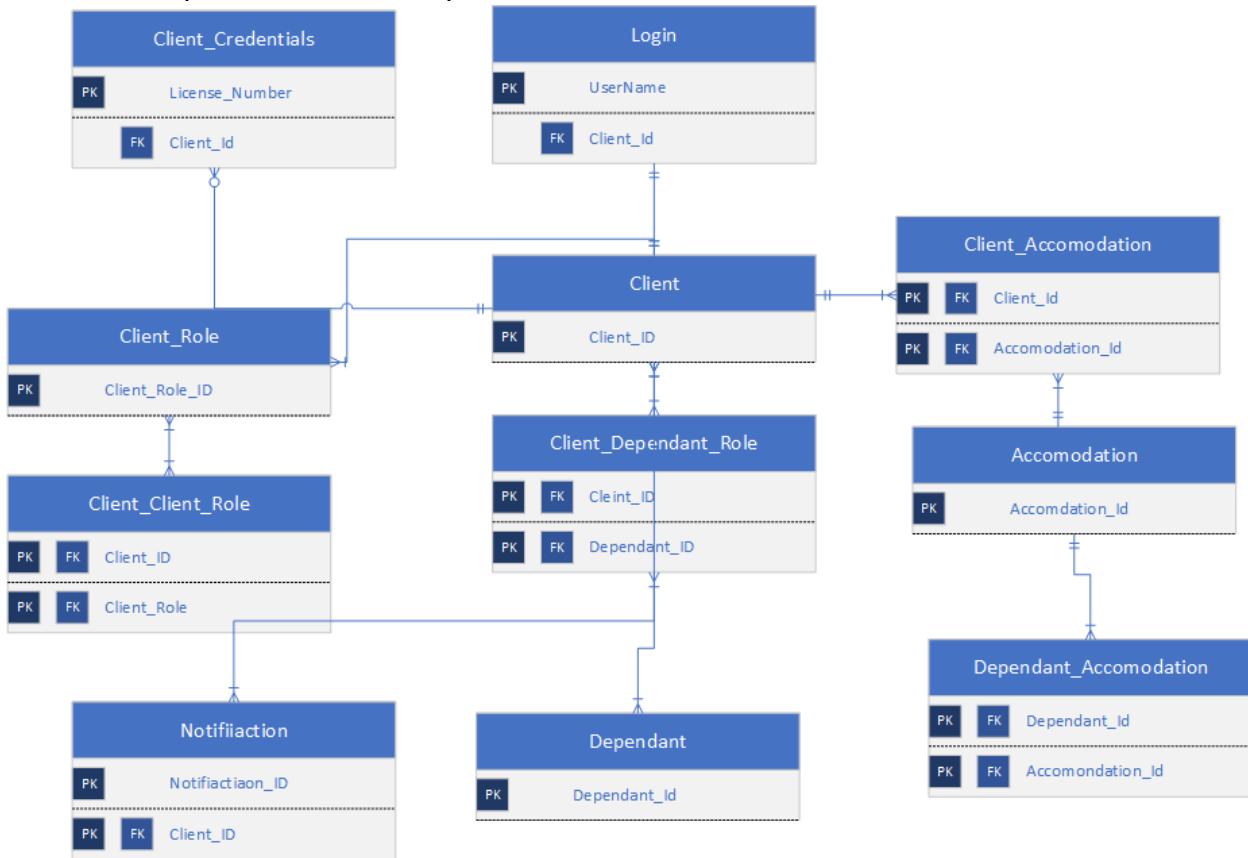
## Vehicle and Directly Related Relationships



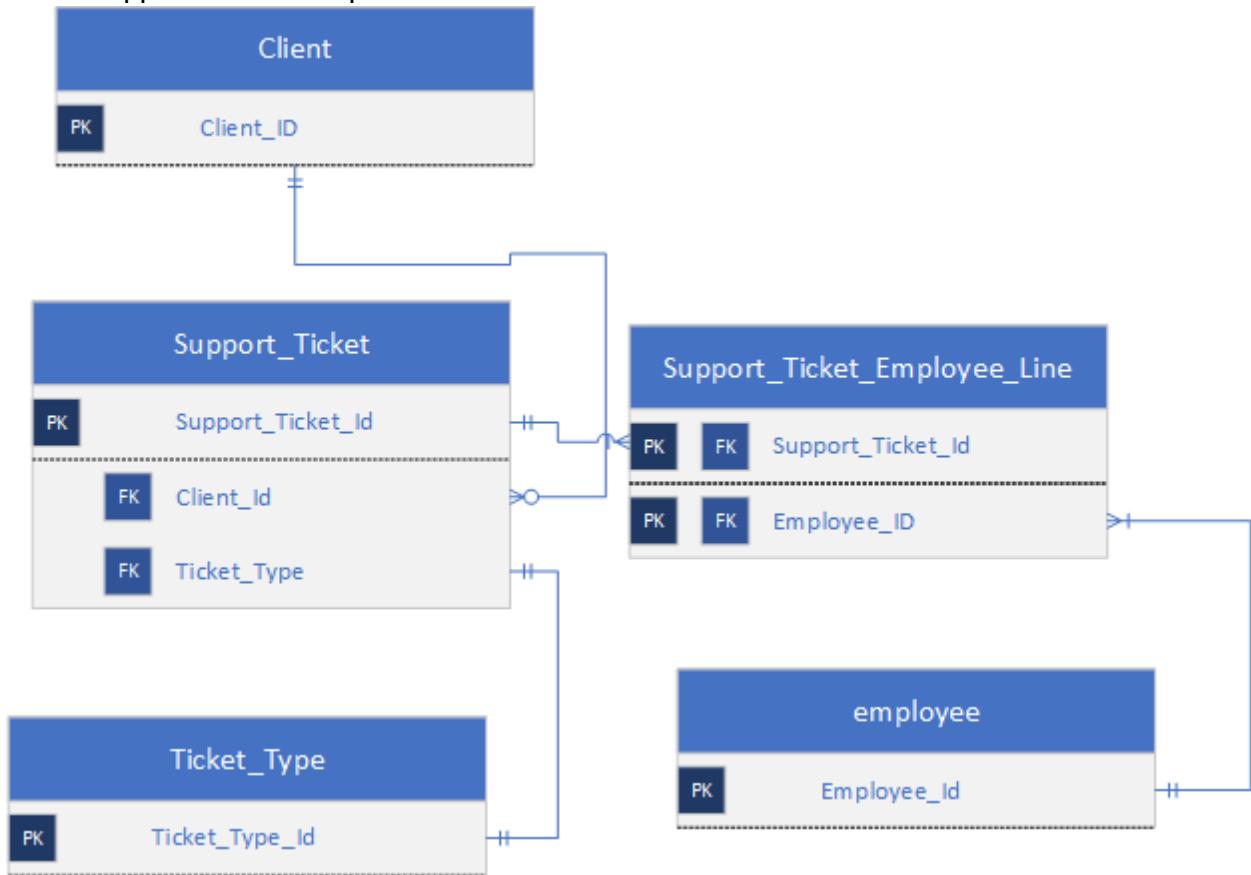
## Driver and Directly Related Relationships



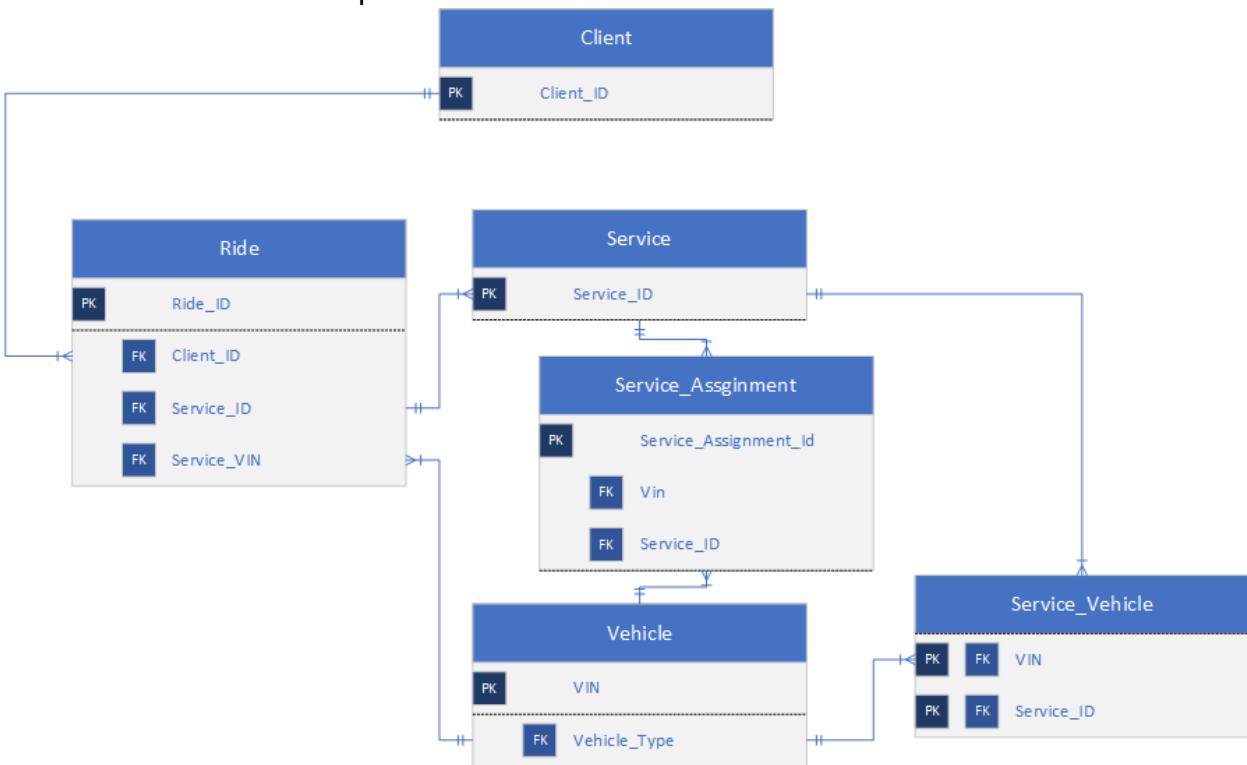
## Client and Dependent Relationships



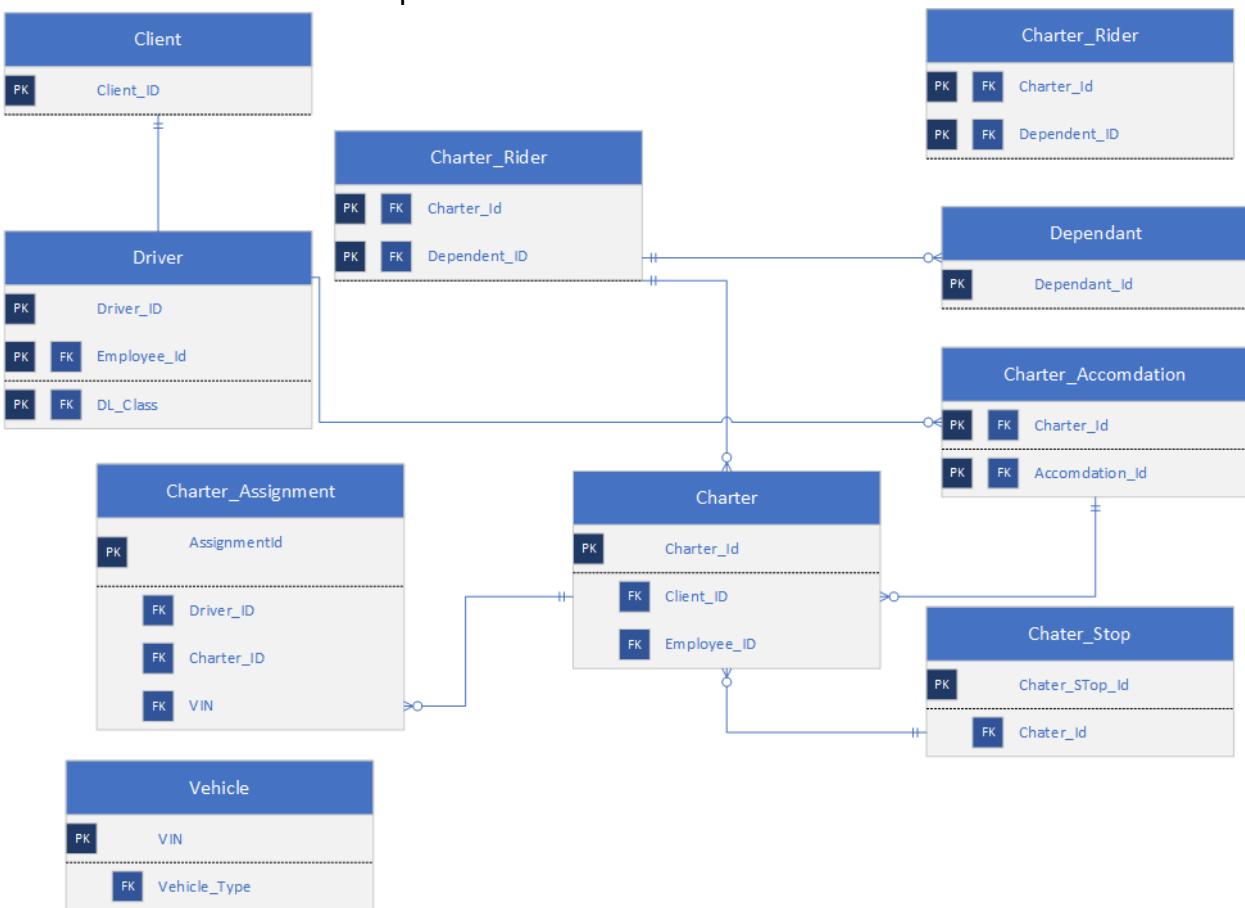
## Client Support Relationships



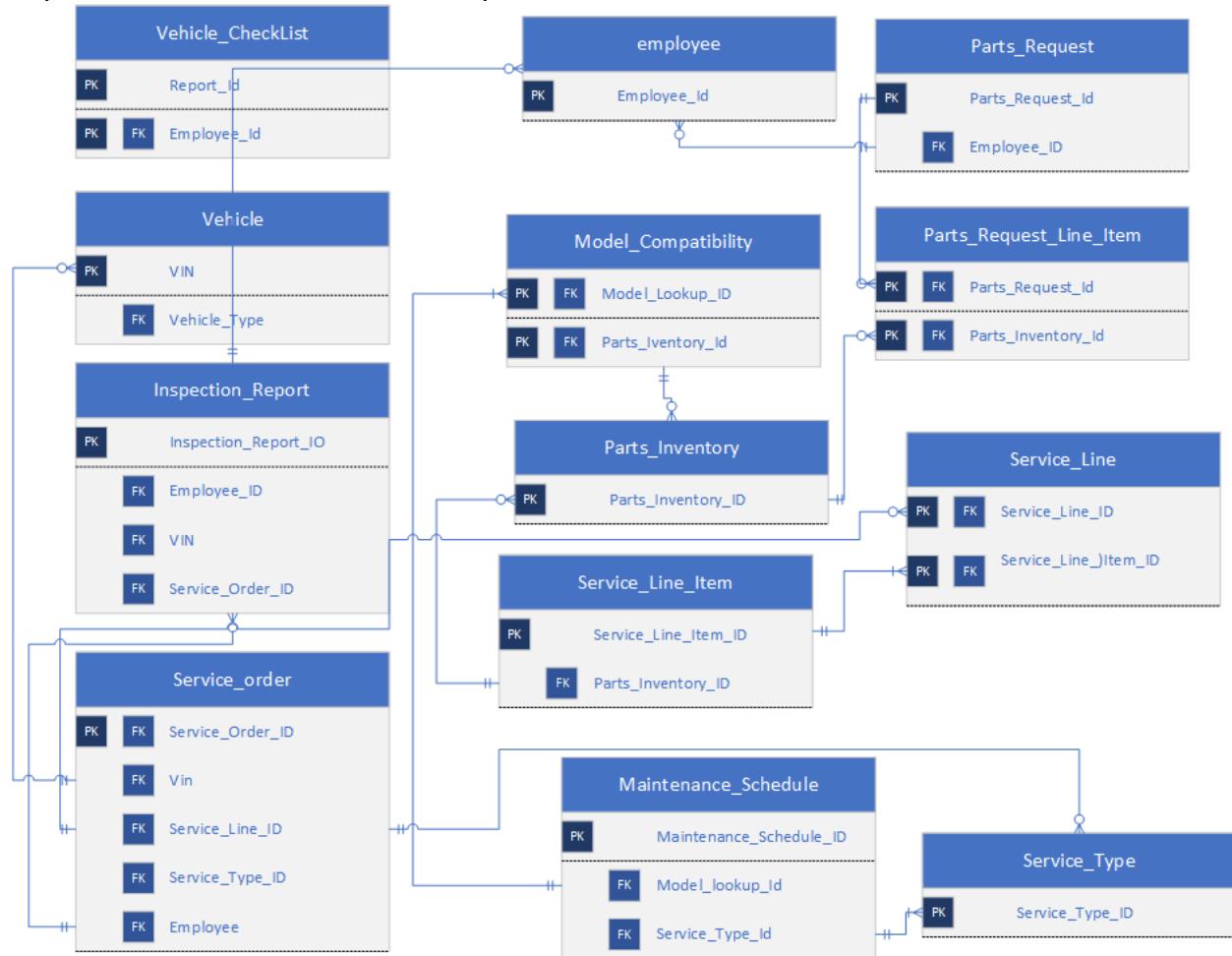
## Client and Ride Relationships



## Client and Charter Relationships



## Repair and Maintenance Relationships



## Data Dictionary

The Data Dictionary fully describes all the fields inside of the data objects. The tables and reference fields in the objects of the Data Dictionary will be used to build the database. The fields in the Data Dictionary will be based on the data necessary to satisfy the Use Case functionality.

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	Inc	NULLS	INDEX	UNIQUE	PK	FK	INTEGRITY	REFERENCES	COMMENTS
<b>Vendor</b>														
Vendor_ID	int			IDENTITY	100000	1	not null		yes					Vendor name
Vendor_Name	nvarchar	100					not null							The name of the vendors company
Vendor_Contact_Given_Name	nvarchar	50					not null							vendor contact person given name
Vendor_Contact_Family_Name	nvarchar	50					not null							vendor contact person family name
Vendor_Contact_Phone_Number	nvarchar	11					not null							vendor contact phone number
Vendor_Contact_Email	nvarchar	255					not null							vendor contact email
Vendor_Phone_Number	nvarchar	11					not null							vendor main phone number
Vendor_Address	nvarchar	50					not null							vendor address
Vendor_Address2	nvarchar	50					null							vendor address detail
Vendor_City	nvarchar	20					not null							vendor city
Vendor_State	nvarchar	2					not null							vendor state
Vendor_Country	nvarchar	3					not null							vendor country
Vendor_Zip	nvarchar	9					not null							vendor zip
Preferred	bit		0				not null							is the vendor preferred for ordering from
Is_Active	bit		1				not null							is the vendor currently in use
<b>Purchase_Order</b>														
Purchase_Order_ID	int			IDENTITY	100000	1	not null		yes					purchase order id number
Vendor_ID	int						not null		yes		vendor.vendor_id			vendor number for the purchase order
Purchase_Order_Date	date		now				not null							date of the purchase order was submitted
Delivery_Address	nvarchar	50					null							address to send products
Delivery_Address2	nvarchar	50					null							address detail to send products
Delivery_City	nvarchar	20					null							city to send products
Delivery_State	nvarchar	2					null							state to send products
Delivery_Country	nvarchar	3					null							country to send products
Delivery_Zip	nvarchar	9					null							zip code to send products
Is_Active	bit		1				not null							not all items have been received

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	Inc	NULLS	INDEX	UNIQUE	PK	FK	INTEGRITY	REFERENCES	COMMENTS
<b>Packing_Slip</b>														
Packing_Slip_ID	int			IDENTITY	100000	1	not null		yes	yes				packing slip id
Purchase_Order_ID	int						not null				yes	purchase_order.purchase_order_ID		composite pk
Receiving_Notes	nvarchar	256					null				yes	vendor.vendor_id		relevant notes
Vendor_ID	int						not null					vendor.vendor_id		composite pk
Creation_Date	date			now			not null							receiving date
Is_Active	bit			1			not null							packing slip active field
<b>Purchase_Order_Line_Item</b>														
Purchase_Order_ID	int						not null			yes	purchase_order.purchase_order_id			composite key, purchase order id
Parts_Inventory_ID	int						not null			yes	parts_inventory.parts_Inventor_y_ID			composite key, part id from parts table
Line_Number	int				1	1	not null							line number for the item
Line_Item_Name	nvarchar	30					null							line item name
Line_Item_Qty	int						not null							Parts_Inventory_ID
Line_Item_Description	nvarchar	100					null							line item description
Is_Active	bit			1			not null							order has been received
<b>Parts_Inventory</b>														
Parts_Inventory_ID	int			IDENTITY	100000	1	not null			yes				part id number
Part_Name	nvarchar	30					not null							part name
Part_Quantity	int		0				not null							part quantity
Item_Description	nvarchar	100					not null							detailed description for the item
Item_Specifications	nvarchar	max					null							detailed specifications for the item
Part_Photo_URL	nvarchar	255					null							photo of part if available
Ordered_Qty	int		0				null							How many have been ordered but not received
Stock_Level	int		0				not null							How many of an item we would like to have in stock at any one time
Is_Active	bit		1				not null							part is currently available

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	Inc	NULLS	INDEX	UNIQUE	PK	FK	INTEGRITY	REFERENCES	COMMENTS
Route_ID	int			yes	1	1	not null		yes					Auto-incrementing ID to define this record
Route_Name	nvarchar	255					not null							Name of the route
Route_Start_Time	time						not null							For example, route cycles start at 6:00 AM.
Route_Cycle	int						not null							Number of minutes one cycle of the route is allotted (route starts every 90 minutes).
Route_End_Time	time						not null							For example, route cycles continue until 8:00 PM.
Days_Of_Service	char	7	0000000				not null							Bit pattern for the days of week that the bus route will run -- each character in the character string corresponds to the day of the week [Sunday][Monday][Tuesday][Wednesday][Thursday][Friday][Saturday] -- a '1' value is true for that day, '0' is false. For instance, M/W/F would be a bit pattern of 0101010.
Is_Active	bit			1			not null							Active status of this route

Stop														
Stop_ID	int			yes	1	1	not null		yes					An auto-incrementing ID of this record
Street_Address	nvarchar	255					not null							Address of the stop
Zip_Code	varchar	5					not null							Zipcode of this stop
Latitude	decimal	(8,6)					not null							allows quick mapping of location without real-time geocoding
Longitude	decimal	(9,6)					not null							allows quick mapping of location without real-time geocoding
Is_Active	bit			1			not null							Active status of this stop

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI QUE	P K	F K	INTEGRI TY	REFERENC E S	COMMENTS
<b>Route_Stop</b>														
Route_ID	int						not null			ye s	ye s		route.route_id	The associated Route_ID of this stop from the Route table
Stop_ID	int						not null			ye s	ye s		stop.stop_id	The associated Stop_ID from Stop table
Route_Stop_Number	int						not null			ye s				1st stop, 2nd stop, 3rd stop, etc
Start_Offset	int						not null							arrival time defined as the number of minutes after the start of the route cycle -- for instance, the first route cycle starts at 6:00AM and this field has a value of 20, then the stop will occur at 6:20 AM for the first route cycle.
Is_Active	bit			1			not null							Active status of this route stop
<b>Vehicle_Type</b>														
Vehicle_Type	nvarchar	50					not null			ye s	ye s			a type of vehicle, like bus or van
Is_Active	bit			1			not null		no	no	no	no		
<b>Vehicle</b>														
VIN	nvarchar	17					not null			ye s	ye s			Vehicle Identification Number
Vehicle_Number	nvarchar	10					not null			ye s				vehicle number, DO NOT REMOVE vehicle number that would be printed on the vehicle and is easy to reference.
Vehicle_Mileage	int						not null							last reported mileage of the vehicle
Vehicle_License_Plate	nvarchar	10					not null							vehicles license plate
Vehicle_Type	nvarchar	50					not null		yes	no	no	ye s	Vehicle_Type. Vehicle_type	The type of vehicle: van, bus, ect..
Date_Entered	date						not null							date vehicle was entered into the system

FIELD	DATATYPE	LEN GTH	DEFAULT	IDENTITY	START	In c	NULLS	INDE X	UNI Q UUE	P K	F K	INTEGRITY	REFERENCE S	COMMENTS
Rental	bit		0				not null							flag for if a vehicle is a rental or owned
Is_Active	bit		1				not null							
Vehicle_Driver	<b>This is being replaced</b>													
Vehicle_Driver_ID	int						not null		ye s	ye s				
Driver_ID	int						null				ye s		Driver.Employee_ID	
VIN	nvarchar	17					not null	yes	ye s	ye s			Vehicle.VIN	
Date_assignment_started	date						not null							
Date_assignment_ended	date						null							
Notes	nvarchar						null						notes about why the vehicle driver assignment ended	
Is_Active	bit		1				not null							
Schedule														
Schedule_ID	nvarchar	50					not null		ye s				schedule name	
Driver_ID	int						not null			ye s		Driver.Employee_ID	Driver's employeeID number	
Week_Days	char	7	0000000				not null						days of the week the schedule is active	
Start_Time	time						not null						when the shift starts (time)	
End_Time	time						not null						when the shift ends (time)	
Start_Date	date						not null						when the schedule starts (date)	
End_Date	date						null						when the schedule ends (date)	
Notes	nvarchar	255					null						relevant notes	
Is_Active	bit		1				not null						is this schedule still active	
Employee														
Employee_ID	int			IDENTIT Y	100000	1	not null		ye s				Employee ID number	
Given_Name	nvarchar	50					not null						employees given name	

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI QUE	P K	F K	INTEGRI TY	REFERENC E S	COMMENTS
Family_Name	nvarchar	50					not null							employees family name
Address	nvarchar	50					not null							employees address
Address2	nvarchar	50					null							employees address detail
City	nvarchar	20					not null							employess city
State	nvarchar	2					not null							employees state
Country	nvarchar	3					not null							employess country
Zip	nvarchar	9					not null							employees zip
Phone_Number	nvarchar	20					not null							employees phone number
Email	nvarchar	50					not null							employees email
Position	nvarchar	20					not null							employees title
Is_Active	bit		1				not null							currently active employee

Roles														
Role_ID	nvarchar	25					not null	Ye s	ye s					pk, an employee role
Is_Active	bit		1				not null							Activity status of this role

Employee_Role												
Employee_ID	int		IDENTIT Y	100000	1	not null	ye s	ye s	Employee.Em ployee_ID	fk Employee_ID from the Employee table		
Role_ID	nvarchar	25				not null	ye s	ye s	Roles.Role_ID	The associated Role_ID of the employee from the Roles table		
Is_Active	bit		1			not null				Activity status of this employee role		

Driver_License_Class													
Driver_License_Class_ID	nvarchar	6	"c"				null	ye s					driver license class
Max_Passenger_Count	int		0				null				max number of passengers that can be transported with the driver license class		
Is_Active	bit		1				not null				activity status of the driver license class		

Driver													
Employee_ID	int					no	ye s	ye s	ye s	Employee.Em ployee_ID	fk, the associated employee_ID from the Employee table		

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	In c	NULLS	INDE X	UNI Q UUE	P K	F K	INTEGRIT Y	REFERENCE S	COMMENTS
Driver_License_Class_ID	nvarchar	6	"C"				no		no	no	yes		Driver_License_Class	The highest driver license class held by the driver
Is_Active	bit		1				not null							Active status of the driver

Route_Assignment														
Assignment_ID	int		yes	100000	1	no		ye s	ye s					Id for assigned route
Driver_ID	int									ye s		Driver.Employee_ID		id for driver assigned to route
Route_ID	int									ye s		Route.Route_ID		id of the route assigned
VIN	nvarchar	17				no		yes	ye s	ye s		Vehicle.VIN		id of the vehicle assigned
Date_Assignment_Started	datetime		now			no		yes	no	no	no			date the assignment began on
Date_Assignment_Ended	datetime					no		yes	no	no	no			date the assignment ended on
Is_Active	bit		1			not null								is the assigned route currently active?

Safety_Report														
Safety_Report_ID	int		IDENTIT Y	100000	1	not null		ye s	ye s	no				report identifier
Employee_ID	int					not null		yes	no	no	ye s	Employee.Employee_ID		id of employee filing report
Date	datetime					not null			no	no	no			date when report is filed
Time_Of_Event	datetime					not null			no	no	no			time of the event
Affected_Party	nvarchar	100				not null		yes	no	no	no			yourself, coworker, passenger, civilian
Description	nvarchar	1000				null		no	no	no	no			description of the event
Result_In_Injury	bit		0			not null		yes	no	no	no			did this event result in an injury?
Is_Active	bit		1			not null		yes	no	no	no			is the report still active?

Refuel_Log														
Refuel_Log_ID	int		IDENTIT Y	100000	1	not null		ye s						generated id for refuel
Driver_ID	int					null				ye s		Driver.Employee_ID		driver that purchased the fuel

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	In c	NULLS	INDEX	UNIQUE	P K	F K	INTEGRITY	REFERENCES	COMMENTS
VIN	nvarchar	17					not null			yes			Vehicle.VIN	vehicle the fuel was put in
Date_Time	datetime		now				not null						auto entry as datetime.now	date fuel was purchased
Mileage	int						not null							mileage of the vehicle when fueled
Fuel_Quantity	int						not null							amount of fuel purchased
Fuel_Price_Per_Gal	smallmoney						not null							price of the fuel
Total_Sale	smallmoney						not null							sale amount for the fuel
Notes	nvarchar	250					null	no	no	no	no			notes from the driver
Is_Active	bit		1				not null							Active status of this refuel log
<b>Service_Type</b>														
Service_Type_ID	nvarchar	256							yes	yes				Used to link a reoccurring schedule to a Service_Order
Service_Description	nvarchar	256												
Is_Active	bit		1				not null							
<b>Model_Lookup</b>														
Model_Lookup_ID	int						not null			yes				A Lookup ID for finding a particular vehicle.
VIN	nvarchar	17					not null			yes			Vehicle.VIN	The Vehicle Identification Number from the Vehicle table.
Max_Passengers	int						not null							The max passengers the vehicle can hold at once.
Vehicle_Make	nvarchar	255					not null							The Vehicle's make.
Vehicle_Model	nvarchar	255					not null							The Vehicle's model.
Vehicle_Year	nvarchar	255					not null							The Vehicle's year.
Is_Active	bit		1				not null							A Vehicle's active/inactive definition.
<b>Maintenance_Schedule</b>														
Maintenance_Schedule_ID	int						not null			yes	yes			

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI QUE	P K	F K	INTEGRI TY	REFERENC E S	COMMENTS
Model_Lookup_ID	int						not null					yes		Model_Lookup.Model_Lookup_ID
Service_Type_ID	nvarchar	256					not null					yes		Service_Type.Service_Type_ID
Frequency_In_Months	int						not null							
Frequency_In_Miles	int													
Is_Completed	bit						not null							
Is_Active	bit						not null							
<b>Service_Order</b>														List of all services to be performed
Service_Order_ID	int						not null					yes		
Service_Order_Version	int						not null					yes		
VIN	nvarchar	17					not null					yes	Vehicle.VIN	vehicle number
Service_Line_Item_ID	int											yes		Service_Line.Service_Line_ID
Service_Type_ID	nvarchar	256										yes		Service_type.Service_Type_Id
Created_By_Employee_ID	int						not null					yes		Employee.Employee_id
Serviced_By_Employee_ID	int											yes		Employee.Employee_id
Date_Begun	DateTime						not null							
Date_Finished	DateTime						not null							
Is_Active	bit													
Critical_Issue	bit						not null							A flag that's set when the service order describes a problem that prevents the vehicle from working or being driven with reasonable safety.
<b>Service_Line</b>														Details of a specific Service_Order
Service_Line_ID	int						not null					yes		Service_Order.Service_Line_ID

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	In c	NULLS	INDEX	UNIQUE	P K	F K	INTEGRITY	REFERENCES	COMMENTS
Service_Line_Item_ID	int						not null		yes	yes			Service_Line_Item.Service_Line_Item_ID	
<b>Service_Line_Item</b>													Details of a specific Service_Order	
Service_Line_Item_ID	int						not null		yes	yes				
Parts_Inventory_ID	int						not null				yes		Parts_Inventory.Parts_Inventory_ID	
Quantity	int						not null							
<b>Special_Service_Order</b>														
Special_Service_Order_ID	int			IDENTITY	100000	1			yes				The ID for this special service	
Service_Order_ID	int									yes		Service_order.	Service_order_id	The original Service Order ID
Event_Description	nvarchar	500												A description of why this got sent to the speical service section
Priority	nvarchar	500												A radio button for low, medium, high
Is_Active	bit		1											
<b>Special_Inspection</b>														
Special_Inspection_ID	int			IDENTITY	100000	1	not null		yes				The ID for this inspecton	
Special_Service_Id	int						not null			yes		Special_Service_order.Speci	al_Service_Id	The speicla service order this is tied to
Inspection_Description	nvarchar	500					not null							Notes from the person who did the inspection
Date	datetime						not null							Date of inspection
Employee_ID	int						not null			yes		Employee.Employee_id		Employee who did this inspection
Is_Active	bit		1				not null							Is This special inspection still active
<b>Bid</b>														
Bid_ID	int			IDENTITY	100000	1	not null		yes				The id number of the bid we received	

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	In c	NULLS	INDEX	UNIQUE	P K	F K	INTEGRITY	REFERENCES	COMMENTS
Special_service_Id	int						not null				yes		Special_Service_Order.Special_Service_Order_ID	The speicla service order this is tied to
vendor_id	int						not null				yes		vendor.vendor_id	The vendor who supplied this bid
Bid_Description	nvarchar	500					not null							A description of the work to be done int eh bid, given to us by the vendor.
Date	datetime						not null							Date bid was received
Amount	float						not null							The quoted amount from the vendor
is_Approved	bit	0					not null							Did we approve this bid
Is_Active	bit	1					not null							Is this bid still active/relevent

Special_Work_Order															
special_Work_Order_ID	int		IDENTITY	100000	1	not null			yes						The ID of this work order
Bid_ID	int						not null				yes		Bid.Bid_ID	The ID of this special approval	
Work_Order_Description	nvarchar	500					not null							A overview of what tasks to do	
Drop_Off_Date	datetime						not null							the date that this work order will occur	
Pick_Up_DAte	datetime						not null							The date we will pick up this order (estiamted)	
Is_Active	bit	1					not null							Is this work order still active/relevent	

Change_Order															
Change_Order_ID	int					not null			yes						change order number
Vendor_Id	int						not null				yes		Vendor.Vendor_ID	vendor being ordered from	
Change_Order_Date	datetime						not null							date change order created	
Original_PO_Number	int						not null				yes		purchase_order.purchase_order_id	original purchase order being changed	
Employee_ID	int						not null				yes		Employee.Employee_ID	the employee that made the change order	

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI Q UE	P K	F K	INTEGRITY	REFERENCE S	COMMENTS
Is_Active	bit		1				not null							reference original purchase order is_active
<b>Change_Order_Line</b>														
Change_Order_ID	int						not null		ye s	ye s		Change_order .Change_orde r_id		change order number
Parts_Inventory_ID	int						not null			ye s		Parts_Inventor y.Parts_invent ory_id		inventory part number
Original_Qty	int						not null							original qty
Updated_Qty	int						null							updated qty
Is_Active	bit		1				not null							references Change_Order.Is_Active
<b>Packing_Slip_Line_Items</b>														
Packing_Slip_Id	int						not null		ye s	ye s		Packing_Slip.Packing_slip_id		
Qty_Recieved	int		0				not null							
Parts_Inventory_Id	int						not null		ye s	ye s		Parts_inventory.Parts_inventory_id		
Is_Active	bit		1				not null							
<b>Model_Compatibility</b>														
Model_Lookup_ID	int						not null		ye s	ye s		Model_Lookup.Model_Lookup_ID		
Parts_Inventory_ID	int						not null			ye s	ye s		Parts_inventory.Parts_inventory_id	
Is_Active	bit		1				not null							
<b>Parts_Request</b>														
Parts_Request_ID	int						not null		ye s					identifying marker for parts requests
Employee_ID	int						not null			ye s		Employee.Employee_id		the employee that created the request
Service_Detail_ID	int						not null			ye s		Service_Detail .Service_Detai l_ID		the service that this request would fulfill
Date_requested	date						not null							the date that the request was made

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	In c	NULLS	INDEX	UNIQUE	P K	F K	INTEGRITY	REFERENCES	COMMENTS
Is_Active	bit		1				not null							has the request been resolved
<b>Parts_Request_Line_Items</b>														
Parts_Request_ID	int						not null			ye s	ye s		Parts_request. Parts_Request _ID	marks which parts request the line is linked to
Parts_Inventory_ID	int						not null			ye s	ye s		Parts_inventory.Parts_invent ory_id	which part is being requested
Qty_requested	int						not null							how many of the part are being requested
Is_Active	bit		1				not null							references parts_request.is_active
<b>Inspection_Report</b>														
Inspection_Report_ID	int						not null			ye s				Auto-incrementing ID to refer to this inspection report record
Employee_ID	int						not null			ye s		Driver.Employ ee_ID	fk, The employee who entered the record. From Employee.Employee_ID table	
VIN	nvarchar	17					not null			ye s		Vehicle.VIN	Vehicle Identification Number of the vehicle that is being inspected	
Service_Order_ID	int						not null			ye s		Service_order. Service_order _id	The service order ID	
Oil_Level	nvarchar	4					null							Used to record if the checked oil level was low, good, or high, prior to trip.
Tire_Pressure	int						null							Used to record the tire pressure prior to trip
Front_Left_Turn_Signal	bit						null							Used to record if the turn signal was working or not.
Front_Right_Turn_Signal	bit						null							Used to record if the turn signal was working or not.
Rear_Left_Turn_Signal	bit						null							Used to record if the turn signal was working or not.

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI QUE	P K	F K	INTEGRIT Y	REFERENC E S	COMMENTS
Rear_Right_Turn_Signal	bit						null							Used to record if the turn signal was working or not.
Left_Brake_Light	bit						null							Used to record if the brake light was working or not
Right_Brake_Light	bit						null							Used to record if the brake light was working or not
Windshield_Washer_Fluid	nvarchar	4					null							Used to record if the checked windshield washer fluid was low, good, or high, prior to the trip.
Problem_Description	nvarchar	256					null							A field used to record if there is any additional vehicle problems prior to the trip.
Is_Active	bit		1				not null							The activity status of the report

Client												
Client_ID	int			IDENTIT Y	100000	1	not null	yes	ye s	ye s		auto number
Given_Name	nvarchar	50					not null					
Family_Name	nvarchar	50					not null	yes				
Middle_Name	nvarchar	100					null					
DOB	date						not null					
Email	nvarchar	255					not null		ye s			
Postal_Code	nvarchar	9					null					
City	nvarchar	50					null					
Region	nvarchar	50					null					
Address	nvarchar	100					null					
Text_Number	nvarchar	11					null		ye s			used for text notification
Voice_Number	nvarchar	11					null		ye s			used for voice communication
Is_Active	bit		1				not null					deactivate rather than delete

Client_Credential												
License_Number	nvarchar	12					not null	yes	ye s	ye s		Name of credential

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	In c	NULLS	INDE X	UNI Q UE	P K	F K	INTEGRI TY	REFERENCE S	COMMENTS
Driver_License_Class_ID	nvarchar	6					not null				yes		Driver_Licence_Class.Driver_License_Class_ID	
License_Expiration	date						not null							
Client_ID	int						not null				yes		Client.Client_ID	
Permission	nvarchar	50					null						level of driving permission	
Certified	bit						null						in-organization certification	
Certification_Description	nvarchar	250					null						name or description of in-organization certification	
Certification_Date	date						null						date of certification	
Is_Active	bit		1				not null							
<b>Login</b>														
Username	nvarchar	50		yes			not null		yes	yes	yes		used for login	
Password_Hash	nvarchar	100	9c9064c59f1ffa2e174ee754d2979be80dd30db552ec03e7e327e9b1a4bd594e'				not null						sha256 = default is hash value for newuser	
Client_ID	int								yes	yes	yes		Client.Client_ID	
Security_Question_1	nvarchar	100												
Security_Question_2	nvarchar	100												
Security_Question_3	nvarchar	100												
Security_Response_1	nvarchar	100												
Security_Response_2	nvarchar	100												
Security_Response_3	nvarchar	100												
Is_Active	bit						not null							
<b>Client_Role</b>														
Client_Role_ID	nvarchar	100	yes				not null		yes	yes	yes		name of role	
Role_Description	nvarchar	500					null						optional description	

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI Q UE	P K	F K	INTEGRITY	REFERENCE S	COMMENTS
Is_Active	bit		1				not null							deactivate rather than delete
<b>Client_Client_Role</b>														
Client_ID	int						not null	yes	ye s	ye s	CASCAD E UPDATE	Client.Client_I D	compound primary key	
Client_Role_ID	nvarchar	100					not null	yes	ye s	ye s	CASCAD E UPDATE	Client_Role.Cli ent_Role_ID	compound primary key	
Is_Active	bit		1				not null							deactivate rather than delete
<b>Accommodation</b>														
Accommodation_ID	nvarchar	100		yes			not null	yes	ye s	ye s				name of accomodation type
Acccommodation_Description	nvarchar	255					null							optional description
Is_Active	bit		1				not null							deactivate rather than delete
<b>Client_Accommodation</b>														
Client_ID	int						not null	yes	ye s	ye s	CASCAD E UPDATE	Client.Client_I D	compound primary key	
Accommodation_ID	nvarchar	100					not null	yes	ye s	ye s	CASCAD E UPDATE	Accommodatio n.Accommoda tion_ID	compound primary key	
Is_Active	bit		1				not null							deactivate rather than delete
<b>Dependent</b>														
Dependent_ID	int			yes	100000	1	not null	yes	ye s	ye s				auto number
Given_Name	nvarchar	50					not null							given name of the dependent
Family_Name	nvarchar	50					not null	yes						family name of the dependent
Middle_Name	nvarchar	100					null							middle name of the dependent
DOB	date						not null							date of birth of the dependent
Gender	nvarchar	20					null							gender of the dependent
Emergency_Contact	nvarchar	100					not null							name of emergency contact
Emergency_Phone	nvarchar	11					not null							phone of emergency contact

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI Q UE	P K	F K	INTEGRIT Y	REFERENCE S	COMMENTS
Is_Active	bit		1				not null							deactivate rather than delete
<b>Dependent_Accommodation</b>														
Dependent_ID	int						not null	yes	ye s	ye s		Cascade UPDATE		auto number
Accommodation_ID	nvarchar	100					not null	yes	ye s	ye s	CASCADE UPDATE	Accommodation_ID		compound primary key
Is_Active	bit		1				not null							deactivate rather than delete
<b>Client_Dependent_Role</b>														
Client_ID	int						not null	yes	ye s	ye s	CASCADE UPDATE	Client.Client_ID		compound primary key
Dependent_ID	int						not null	yes	ye s	ye s	CASCADE UPDATE	Dependent.Dependent_ID		compound primary key
Relationship	nvarchar	100	Legal Guardian'				not null							brief description of guardian/dependent relationship
Is_Active	bit		1				not null							deactivate rather than delete
<b>Notification</b>														
Notificaiton_ID	int			IDENTIT Y	100000	1	not null	yes	ye s	ye s				Auto incrementing id of the notification record
Client_ID	int						not null	yes	ye s	ye s	CASCADE UPDATE	Client.Client_ID		
Title	nvarchar	255	User Notification'				not null							title, header, or subject line
Notification_Body	nvarchar	5000					not null							main text of notification
Time_Sent	datetime			CURRENT_TIMESTAMP			not null							the time the notification was sent
Viewed	bit		0				not null							has client viewed notification
Is_Active	bit		1				not null							deactivate rather than delete
<b>Ticket_Type</b>														
Ticket_Type_ID	nvarchar	50		yes			not null	yes	ye s	ye s				
Type_Description	nvarchar	500					null							

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	Inc	NULLS	INDEX	UNIQUE	PK	FK	INTEGRITY	REFERENCES	COMMENTS
Is_Active	bit	1					not null							deactivate rather than delete
<b>Support_Ticket</b>														
Support_Ticket_ID	int			IDENTITY	100000	1	not null	yes	yes	yes				auto increment
Ticket_Type_ID	nvarchar	50					not null				yes			Ticket_Type(Ticket_Type_ID)
Client_ID	int						not null	yes			yes	CASCADE UPDATE		Ticket_Type.Ticket_Type_ID
Time_Opened	datetime			CURRENT_TIMESTAMP			not null					CASCADE UPDATE		Client.Client_ID
Time_Closed	datetime						null							
Support_Note	nvarchar	3000					null							
Is_Open	bit		1				not null							Ticket is open in support process
Is_Active	bit	1					not null							deactivate rather than delete
<b>Support_Ticket_Employee_Line</b>														
Support_Ticket_ID	int						not null	yes	yes	yes	CASCADE UPDATE	Support_Ticket.Support_Ticket_ID		compound primary key
Employee_ID	int						not null	yes	yes	yes	CASCADE UPDATE	Employee.Employee_ID		compound primary key
Time_Assigned	datetime			CURRENT_TIMESTAMP			not null							
Is_Active	bit	1					not null							
<b>Charter</b>														
Charter_ID	int			IDENTITY	100000	1	not null	yes	yes	yes				auto increment; id of reservation; used for lookup and confirmation emails to clients
Client_ID	int						not null	yes		yes		Client.Client_ID		foreign key; id of client/user

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI QUE	P K	F K	INTEGRI TY	REFERENC E S	COMMENTS
Client_Is_Passenger	bit		1				not null							denotes whether the client (client_id) is also a passenger; 1 = client is a passenger, 0 = client is not a passenger
Rider_Quantity	int						not null							number of riders that are planned to be riding on the charter request
Driver_Needed	bit		1				not null							determines whether a driver is needed; 0 = not needed, 1 = needed
Reviewed_By	int						null			ye s		Employee.Employee_ID		employee_id of the employee who approved/rejected the charter request
Is_Approved	bit						null							denotes whether reservation has been approved; null = not reviewed yet, 1 = approved, 0 = not approved
Date_Request_Start	datetime						not null							date chosen by client of when they want the reservation to start
Date_Request_End	datetime						not null							date chosen by client of when they want the reservation to end
Is_Active	bit		1				not null							denotes whether reservation is active; 1 = active, 0 = not active

Charter_Assignment														
Assignment_ID	int			yes	100000	1	no		ye s	ye s				Id for assigned charter
Driver_ID	int						yes			ye s		Driver.Employee_ID		id for driver assigned to charter
Charter_ID	int								ye s			Charter.Charter_ID		id of the charter assigned
VIN	nvarchar	17					no	yes	ye s	ye s		Vehicle.VIN		id of the vehicle assigned
Vehicle_Type	nvarchar	50					not null			ye s		Vehicle_Type.Vehicle_Type		denotes the idea of what kind of vehicle is on the request (ie. bus, car, van, etc)

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	Inc	NULLS	INDEX	UNIQUE	PK	FK	INTEGRITY	REFERENCES	COMMENTS
Date_Issued	datetime						null							date vehicle left inventory null = vehicle not sent out yet
Date_Returned	datetime						null							date vehicle returned to inventory null = vehicle not returned yet
Is_Active	bit		1				not null							is the vehicle assigned to the charter currently active?

Charter_Accommodation														
Charter_ID	int						not null	yes	yes	yes	yes		Charter.Charter_ID	auto increment; id of reservation; used for lookup and confirmation emails to clients
Accommodation_ID	nvarchar	100					not null	yes	yes	yes	yes		Accommodation.Accommodation_ID	
Is_Active	bit		1				not null							denotes whether reservation is active; 1 = active, 0 = not active

Charter_Stop														
Charter_Stop_ID	int			yes	1	1	not null		yes					pk of the table
Charter_ID	int						not null		yes			Charter.Charter_ID		fk, Charter_ID from the Charter table
Street_Address	nvarchar	255					not null							address of the stop
Zip_Code	nvarchar	5					not null							zipcode of the stop
Latitude	decimal	(8,6)					not null							allows quick mapping of location without real-time geocoding
Longitude	decimal	(9,6)					not null							allows quick mapping of location without real-time geocoding
Duration	int						not null							assume that the time is in minutes
Description	nvarchar	255					null							description of the stop
Stop_Number	int						not null							when a charter has multiple stops, this would define the order in which the stops occur

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	In c	NULLS	INDEX	UNIQUE	P K	F K	INTEGRITY	REFERENCES	COMMENTS
Is_Active	bit		1				not null							activity status of the stop
<b>Charter_Rider</b>														
Charter_ID	int						not null			ye s	ye s		Charter.Charter_ID	
Dependent_ID	int						not null			ye s	ye s		Dependent.Dependent_ID	
Is_Active	bit		1				not null							
<b>Vehicle_Checklist</b>														
Checklist_ID	int			IDENTIT Y	100000	1	no	no	ye s	ye s	no			composite primary key
Employee_ID	int						no	yes	no	ye s	ye s		Employee.Employee_ID	composite primary key
VIN	nvarchar	17					no	yes	no	ye s	ye s		Vehicle.VIN	composite primary key
Date	datetime						no	no	no	no	no			date created
Clean	bit		0				no	no	no	no	no			Is the vehicle clean?
Pedals	bit		0				no	no	no	no	no			Are the pedals functioning correctly?
Dash	bit		0				no	no	no	no	no			Are components of the dash functioning?
Steering	bit		0				no	no	no	no	no			Is the steering working as intended?
AC_Heat	bit		0				no	no	no	no	no			Does the AC/Heat system work?
Mirror	bit		0				no	no	no	no	no			Is the mirror not broken?
Cosmetic	nvarchar	500					yes	no	no	no	no			Field to note any cosmetic damages to a vehicle.
Tire_Pressure_DF	int						no	no	no	no	no			Driver Front tire pressure
Tire_Pressure_PF	int						no	no	no	no	no			Passenger Front tire pressure
Tire_Pressure_DR	int						no	no	no	no	no			Driver Rear tire pressure
Tire_Pressure_PR	int						no	no	no	no	no			Passenger Right tire pressure
Blinker_DF	bit		0				no	no	no	no	no			Driver Front blinker is functioning

FIELD	DATATYPE	LEN GTH	DEFAULT	IDENTITY	START	In c	NULLS	INDE X	UNI Q UE	P K	F K	INTEGRITY	REFERENCE S	COMMENTS
Blinker_PF	bit	0					no	no	no	no	no			Passenger Front blinker is functioning
Blinker_DR	bit	0					no	no	no	no	no			Driver Rear blinker is functioning
Blinker_PR	bit	0					no	no	no	no	no			Passenger Reader blinker is functioning
Breaklight_DF	bit	0					no	no	no	no	no			Driver Front breaklight is functioning
Breaklight_PF	bit	0					no	no	no	no	no			Passenger Front breaklight is functioning
Breaklight_DR	bit	0					no	no	no	no	no			Driver Rear breaklight is functioning
Breaklight_PR	bit	0					no	no	no	no	no			Passenger Rear breaklight is functioning
Headlight_Driver	bit	0					no	no	no	no	no			Driver side headlight is working
Headlight_Passenger	bit	0					no	no	no	no	no			Passenger side headlight working
TailLight_Driver	bit	0					no	no	no	no	no			Driver side taillight is functioning
TailLight_Passenger	bit	0					no	no	no	no	no			Passenger side taillight is functioning
Wiper_Driver	bit	0					no	no	no	no	no			Driver side wiper is functioning
Wiper_Passenger	bit	0					no	no	no	no	no			Passenger side wiper is functioning
Wiper_Rear	bit	0					yes	no	no	no	no			Rear wiper is functioning, if applicable
Seat_Belts	bit	0					no	no	no	no	no			Seat belts are functional
Fire_Extinguisher	bit	0					no	no	no	no	no			If applicable, fire extinguisher is present and not expired
Airbags	bit	0					no	no	no	no	no			Airbags appear functional
First_Aid	bit	0					no	no	no	no	no			First Aid kit is present, not used, and not expired
Emergency_Kit	bit	0					no	no	no	no	no			Emergency Kit is present
Mileage	int						no	no	no	no	no			Current mileage
Fuel_Level	int						no	no	no	no	no			Current fuel level
Breaks	bit	0					no	no	no	no	no			Breaks are functional

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	In c	NULLS	INDEX	UNIQUE	P K	F K	INTEGRITY	REFERENCES	COMMENTS
Accelerator	bit	0					no	no	no	no	no			Accelerator is functional
Clutch	bit	0					no	no	no	no	no			Clutch is functional
Notes	nvarchar	1000					no	no	no	no	no			Any added notes regarding the inspection
Is_Active	bit	1					not null							The status of the report
<b>Driver_Maintenance_Report</b>														
Driver_Maintenance_Report_ID	int			IDENTITY	100000	1	no	no	yes	yes	no			PK ID of a report
Driver_ID	int						no	yes	no	no	yes	Driver.Employee_ID		ID of driver entering report
Date_Time	datetime	now					no	no	no	no				date of report (def: now)
VIN	nvarchar	17					no	yes	no	no	yes	Vehicle.VIN		Vehicle Identification number
Severity	nvarchar	20	"low"				no	no	no	no	no			{low, moderate, high}
Description	nvarchar	250					no	no	no	no	no			description of issue
Is_Active	bit	1					not null	no	no	no	no			Status field
<b>Service</b>														
Service_ID	nvarchar	20					not null		yes	no				
Type	nvarchar	20					not null		no	no				
Is_Active	bit	1					not null							
<b>Service_Assignment</b>														
Service_Assignment_ID	int						not null		yes	no				
VIN	nvarchar	17					not null		no	yes		Vehicle.VIN		
Service_ID	int						not null		no	yes		Service.Service_ID		
Driver_ID	int						not null							
Start_Datetime	datetime													
End_Datetime	datetime													
Is_Active	bit	1					not null							
<b>Ride</b>														
Ride_ID	int						no		yes	yes	no			

FIELD	DATATYPE	LEN GTH	DEFA ULT	IDENTIT Y	START	In c	NULLS	INDE X	U NI Q UE	P K	F K	INTEGRI TY	REFERENC E S	COMMENTS
Client_ID	int						no		no	no	ye s		Client.Client_I D	
Service_ID	int						no		no	no	ye s		Service.Service_Id	
Service_Assignment_ID	int						yes		no	no	ye s		Service_Assignment.Service_Assignment_ID	
Pickup_Location	nvarchar	100					no		no	no	no			
Dropoff_Location	nvarchar	100					no		no	no	no			
Scheduled_Pickup_Time	datetime						yes		no	no	no			
Estimated_Dropoff_Time	datetime						yes		no	no	no			
Actual_Pickup_Time	datetime						yes		no	no	no			
Actual_Dropoff_Time	datetime						yes		no	no	no			
Requested	bit						no		no	no	no			
Is_Active	bit	0					not null		no	no	no			

Source															
Vendor_Id	int						not null		ye s	ye s		Vendor.Vendo r_ID		Vendor for ordering part	
Parts_inventory_id	int						not null		ye s	ye s		Parts_Inventor y.part_id		part to order	
Vendor_Part_Number	nvarchar	100					not null							vendors part number	
Part_Order_Qty	int		1				not null							quantity items can be ordered in, ex. box of 100	
Estimated_delivery_time_days	int						null							typical lead time for part to be delivered	
Part_Price	smallmone y						not null							vendors price for the part	
Minimum_order_Qty	int						null							minimum of part that need to be ordered, ex. you have to order at least 100	

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	Inc	NULLS	INDEX	UNIQUE	PK	FK	INTEGRITY	REFERENCES	COMMENTS
Active	bit		1				not null							Whether or not the sourcing of the vendor is active, e.g., the vendor no longer carries the part or the vendor's price for the part is too high.
<b>dummy</b>														
dummy	dummy								yes	yes			School.School_ID	
<b>Vehicle_Unavailable</b>														
Unavailable_ID	int			IDENTITY	1	1	NOT NULL		yes					
VIN	nvarchar	17					NOT NULL		yes				Vehicle.VIN	
Start_Datetime	datetime						NOT NULL							
End_Datetime	datetime						NOT NULL							
Reason	nvarchar	1000					NOT NULL							
is_active	bit		1				NOT NULL							
<b>Driver_Unavailable</b>														
Unavailable_ID	int			IDENTITY	1	1	NOT NULL		yes					
Driver_ID	int						NOT NULL		yes				Driver.Employee_ID	
Start_Datetime	datetime						NOT NULL							
End_Datetime	datetime						NOT NULL							
Reason	datetime						NOT NULL							
is_active	bit		1				NOT NULL							

Data Dictionary End

## Stored Procedures for Data Operations

The Stored Procedures describe the interfaces by which the application interacts with the database. The Stored Procedures may take parameters from the application and may return data to the application. The application will call Stored Procedures to accomplish data permanence requirements.

## Accommodation Stored Procedure Definitions

### Accommodation

#### **sp\_select\_active\_accommodations**

Table: Accommodation  
Parameter: None  
Returns: Accommodation\_ID, Accommodation\_Description

#### **sp\_select\_accommodation\_by\_id**

Table: Accommodation  
Parameter: @Accommodation\_ID  
Returns: Accommodation\_ID, Accommodation\_Description, Active

#### **sp\_insert\_accommodation**

Tables: Accommodation  
Parameters: @Accommodation\_ID, @Accommodation\_Description  
Returns: int (@@RowsAffected)

#### **sp\_update\_accommodation**

Tables: Accommodation, Client\_Accommodation  
Parameters: @OldAccommodation\_ID, @OldAccommodation\_Description,  
              @NewAccommodation\_Description,  
Returns: int (@@RowsAffected)

#### **sp\_deactivate\_accommodation**

Tables: Accommodation, Client\_Accommodation  
Parameters: @Accommodation\_ID  
Returns: int (@@RowsAffected)

## **Client Accommodation Stored Procedures**

### **sp\_select\_active\_accommodations\_by\_client\_id**

Tables: Client, Client\_Accommodation, Accommodation

Parameters: @Client\_ID

Returns: Client\_Accommodation.Active, Client.ID, Client.Given\_Name,  
Client.Family\_Name, Client.Middle\_Name, Client.DOB,  
Accommodation.Accommodation\_ID, Accommodation.Description

### **sp\_select\_active\_users\_by\_accommodation\_id**

Tables: Client, Client\_Accommodation, Accommodation

Parameters: @Accommodation\_ID

Returns: Client\_Accommodation.Active, Client.Given\_Name,  
Client.Family\_Name, Client.Middle\_Name, Client.DOB,  
Accommodation.Accommodation\_ID, Accommodation.Description

### **sp\_select\_client\_accommodation**

Tables: Client, Client\_Accommodation, Accommodation

Parameters: @Client\_ID, @Accommodation\_ID

Returns: Client\_Accommodation.Active, Client.Given\_Name,  
Client.Family\_Name, Client.Middle\_Name, Client.DOB,  
Accommodation.Accommodation\_ID, Accommodation.Description

### **sp\_insert\_client\_accommodation**

Table: Client\_Accommodation

Parameters: @Client\_ID, @Accommodation\_ID

Returns: Active

### **sp\_update\_client\_accommodation**

Table: Client\_Accommodation

Parameters: @OldClient\_ID, @OldAccommodation\_ID, @NewClient\_ID,  
@NewAccommodation\_ID

Returns: int (@@RowsAffected)

### **sp\_deactivate\_client\_accommodation**

Table: Client\_Accommodation

Parameters: @Client\_ID, @Accommodation\_ID

Returns: int (@@RowsAffected)

## Bid Related Stored Procedures

### **sp\_get\_all\_active\_Bid**

Tables: Bid, special\_service\_detail

Parameters: None

Returns: Bid\_id, Special\_service\_Detial.event\_description,  
vendor\_id,Bid\_Description, amount, is\_approved

### **sp\_get\_all\_approved\_Bid**

Tables: Bid, special\_service\_detail

Parameters: None

Returns: Bid\_id, Special\_service\_Detial.event\_description,  
vendor\_id,Bid\_Description, amount, active

### **sp\_get\_Bid\_by\_bid\_id**

Tables: Bid, special\_service\_detail

Parameters: @bid\_id

Returns: Special\_service\_Detial.event\_description, vendor\_id,  
Bid\_Description, amount, approved, is\_active,  
is\_approved

### **sp\_get\_active\_Bid\_for\_special\_service\_id**

Tables: Bid, special\_service

Parameters: Special\_service\_id

Returns: Special\_service\_Detial.event\_description, vendor\_id,  
Bid\_Description, amount, approved, amount,approved,  
is\_active

### **sp\_get\_Bid\_by\_bid\_id\_for\_foreign\_key\_vendor**

Tables: Bid, Vendor

Parameters: Vendor\_ID

Returns: Special\_service\_Detial.event\_description, vendor\_id,  
amount,Bid\_Description approved, is\_active

**sp\_insert\_bid**

Tables: Bid  
Parameters: @special\_sevice\_id, @vendor\_id, @Date, @amount  
              @Bid\_Description  
Returns: Bid\_ID (scope\_identity())

**sp\_approve\_bid**

Tables: Bid  
Parameters: @bid\_id  
Returns: Int (@@rowsAffected)

**sp\_update\_bid**

Tables: Bid  
Parameters: @bid\_id  
              @old\_special\_sevice\_id, @old\_vendor\_id, @old\_Date,  
              @old\_amount @old\_special\_sevice\_id, @old\_vendor\_id,  
              @old\_Bid\_Description, @old\_Date, @old\_amount,  
              @new\_special\_sevice\_id, @new\_vendor\_id, @new\_Date,  
              @new\_amount @old\_special\_sevice\_id, @new\_vendor\_id,  
              @new\_Bid\_Description, @new\_Date, @new\_amount,  
  
Returns: Int (@@rowwsAffected)

**sp\_deactivate\_bid**

Tables: bid  
Parameters: @bid\_id  
Returns: Int (@@rowsAffecteD)

**sp\_activate\_bid**

Tables: bid

Parameters: @bid\_id

Returns: Int (@@rowsAffected)

**Stored Procedures for Change Order Line Items**  
**Initial Creator: Chris Baenziger**  
**Creation Date: 2023-10-08**

**sp\_insert\_change\_order\_line**

Tables: Change\_Order\_Line  
Parameters: @Change\_Order\_ID, @Parts\_Inventory\_ID,  
@Line\_Item\_Name, @Line\_Item\_Original\_Qty,  
@Line\_Item\_Updated\_Qty, @Line\_Item\_Description  
Returns: Line\_Number

**sp\_select\_change\_order\_line**

Tables: Change\_Order\_Line  
Parameters: @Change\_Order\_ID, @Line\_Number  
Returns: Change\_Order\_ID, Parts\_Inventory\_ID,  
Line\_Item\_Name, Line\_Number,  
Line\_Item\_Original\_Qty, Line\_Item\_Updated\_Qty,  
Line\_Item\_Description

**sp\_update\_change\_order\_line**

Tables: Change\_Order\_Line  
Parameters: @OldChange\_Order\_ID, @OldParts\_Inventory\_ID,  
@OldLine\_Item\_Name, @OldLine\_Number,  
@OldLine\_Item\_Original\_Qty,  
@OldLine\_Item\_Updated\_Qty,  
@OldLine\_Item\_Description @NewChange\_Order\_ID,  
@NewParts\_Inventory\_ID, @NewLine\_Item\_Name,  
@NewLine\_Number, @NewLine\_Item\_Original\_Qty,  
@NewLine\_Item\_Updated\_Qty,  
@NewLine\_Item\_Description  
Returns: int(@@RowsAffected)

**sp\_deactivate\_change\_order\_line**

Tables: Change\_Order\_Line  
Parameters: @Change\_Order\_ID, @Line\_Number  
Returns: int(@@RowsAffected)

**sp\_delete\_change\_order\_line**

Tables: Change\_Order\_Line  
Parameters: @Change\_Order\_ID, @Line\_Number  
Returns: int(@@RowsAffected)

Change Order

**Initial Creator: Chris Baenziger**  
**Creation Date: 2023-10-09**

**sp\_insert\_change\_order**

Tables: Change\_Order  
Parameters: @Vendor\_ID, @Change\_Order\_Date,  
@Delivery\_Address, @Delivery\_Address2,  
@Delivery\_City, @Delivery\_State,  
@Delivery\_Country, @Delivery\_State, @Delivery\_Zip  
Returns: Change\_Order\_ID

**sp\_select\_change\_order\_by\_change\_order\_ID**

Tables: Change\_Order  
Parameters: @Change\_Order\_ID  
Returns: Vendor\_ID, Change\_Order\_Date, Delivery\_Address,  
Delivery\_Address2, Delivery\_City, Delivery\_State,  
Delivery\_Country, Delivery\_State, Delivery\_Zip

**sp\_select\_change\_orders\_by\_date\_range**

Tables: Change\_Order  
Parameters: @Start\_Date, @End\_Date  
Returns: Vendor\_ID, Change\_Order\_Date, Delivery\_Address,  
Delivery\_Address2, Delivery\_City, Delivery\_State,  
Delivery\_Country, Delivery\_State, Delivery\_Zip

**sp\_update\_change\_order**

Tables: Change\_Order  
Parameters: @OldChange\_Order\_ID, @OldVendor\_ID,  
@OldChange\_Order\_Date, @OldDelivery\_Address,  
@OldDelivery\_Address2, @OldDelivery\_City,  
@OldDelivery\_State, @OldDelivery\_Country,  
@OldDelivery\_State, @OldDelivery\_Zip  
@NewChange\_Order\_ID, @NewVendor\_ID,  
@NewChange\_Order\_Date, @NewDelivery\_Address,  
@NewDelivery\_Address2, @NewDelivery\_City,  
@NewDelivery\_State, @NewDelivery\_Country,  
@NewDelivery\_State, @NewDelivery\_Zip  
Returns: int(@@RowsAffected)

**Sp\_deactivate\_change\_order**

Tables: Change\_Order  
Parameters: @Change\_Order\_ID  
Returns: int(@@RowsAffected)

**sp\_delete\_change\_order**

Tables: Change\_Order  
Parameters: @Change\_Order\_ID  
Returns: int(@@RowsAffected)

## **Charter\_Accommodation Stored Procedures**

**Written by: Isabella Rosenbohm 10/16/2023**

**Updated by: Isabella Rosenbohm 10/23/2023**

### **sp\_get\_all\_charter\_accommodations**

**Tables:** Charter\_Accommodation

**Parameters:** None

**Returns:** Charter\_ID, Accommodation\_ID, Is\_Active

### **sp\_get\_charter\_accommodations**

**Tables:** Charter\_Accommodation

**Parameters:** None

**Returns:** Charter\_ID, Accommodation\_ID, Is\_Active

### **sp\_get\_inactive\_charter\_accommodations**

**Tables:** Charter\_Accommadation

**Parameters:** None

**Returns:** Charter\_ID, Accommodation\_ID, Is\_Active

### **sp\_get\_charter\_accommodations\_by\_charter\_id**

**Tables:** Charter\_Accommadation, Charter

**Parameters:** @Charter\_ID

**Returns:** Charter.Charter\_ID, Accommodation\_ID, Is\_Active

### **sp\_get\_charter\_accommodations\_by\_accommodation\_id**

**Tables:** Charter\_Accommadation, Accommodation

**Parameters:** @Accommodation\_ID

**Returns:** Charter\_ID, Accommodation.Accommodation\_ID, Is\_Active

### **sp\_insert\_charter\_accommodation**

**Tables:** Charter\_Accommadation, Charter, Accommodation

**Parameters:** @Accommodation.Accommodation\_ID, @Charter.Charter\_ID, @Is\_Active

**Returns:** int (@@RowsAffected)

**sp\_update\_charter\_accommodation**

Tables: Charter\_Accommodation  
Parameters: @Accommodation\_ID, @Charter\_ID, @Is\_Active  
Returns: int (@@RowsAffected)

**sp\_deactivate\_charter\_accommodation**

Tables: Charter\_Accommodation  
Parameters: @Is\_Active  
Returns: int (@@RowsAffected)

**sp\_activate\_charter\_accommodation**

Tables: Charter\_Accommodation  
Parameters: @Is\_Active  
Returns: int (@@RowsAffected)

**Charter\_Assignment Stored Procedures**

Written by: Isabella Rosenbohm 11/13/23

**sp\_get\_charter\_assignment\_by\_id**

Tables: Charter\_Assignment  
Parameters: @Assignment\_ID  
Returns: Assignment\_ID, Driver\_ID, Charter\_ID, VIN\_Number, Vehicle\_Type\_ID, Date\_Issued, Date\_Returned, Is\_Active

**sp\_get\_charter\_assignments**

Tables: Charter\_Assignment  
Parameters: None  
Returns: Assignment\_ID, Driver\_ID, Charter\_ID, VIN\_Number, Vehicle\_Type\_ID, Date\_Issued, Date\_Returned, Is\_Active

**sp\_get\_inactive\_charter\_assignments**

Tables: Charter\_Assignment  
Parameters: None  
Returns: Assignment\_ID, Driver\_ID, Charter\_ID, VIN\_Number, Vehicle\_Type\_ID, Date\_Issued, Date\_Returned, Is\_Active

**sp\_get\_all\_charter\_assignments**

Tables: Charter\_Assignment  
Parameters: None  
Returns: Assignment\_ID, Driver\_ID, Charter\_ID, VIN\_Number, Vehicle\_Type\_ID, Date\_Issued, Date\_Returned, Is\_Active

**sp\_get\_charter\_assignments\_by\_charter\_id**

Tables: Charter\_Assignment  
Parameters: @Charter\_ID  
Returns: Assignment\_ID, Driver\_ID, Charter\_ID, VIN\_Number, Vehicle\_Type\_ID, Date\_Issued, Date\_Returned, Is\_Active

**sp\_get\_inactive\_charter\_assignments\_by\_charter\_id**

Tables: Charter\_Assignment  
Parameters: @Charter\_ID, @Is\_Active  
Returns: Assignment\_ID, Driver\_ID, Charter\_ID, VIN\_Number, Vehicle\_Type\_ID, Date\_Issued, Date\_Returned, Is\_Active

**sp\_insert\_charter\_assignment**

Tables: Charter\_Assignment  
Parameters: @Assignment\_ID, @Driver\_ID, @Charter\_ID, @VIN\_Number, @Vehicle\_Type\_ID, @Date\_Issued, @Date\_Returned, @Is\_Active  
Returns: Assignment\_ID

**sp\_update\_charter\_assignment**

Tables: Charter\_Assignment  
Parameters: @Assignment\_ID, @Old\_Driver\_ID, @Old\_Charter\_ID, @Old\_VIN\_Number, @Old\_Vehicle\_Type\_ID, @Old\_Date\_Issued, @Old\_Date\_Returned, @Old\_Is\_Active, @New\_Driver\_ID, @New\_Charter\_ID, @New\_VIN\_Number, @New\_Vehicle\_Type\_ID, @New\_Date\_Issued, @New\_Date\_Returned, @New\_Is\_Active  
Returns: int (@@RowsAffected)

**sp\_update\_charter\_assignment\_date\_issued**

Tables: Charter\_Assignment

Parameters: @Date\_Issued, @Assignment\_ID

Returns: int (@@RowsAffected)

#### **sp\_update\_charter\_assignment\_date\_returned**

Tables: Charter\_Assignment

Parameters: @Date\_Returned, @Assignment\_ID

Returns: int (@@RowsAffected)

#### **sp\_update\_charter\_assignment\_driver\_id**

Tables: Charter\_Assignment

Parameters: @Driver\_ID, @Assignment\_ID

Returns: int (@@RowsAffected)

#### **sp\_update\_charter\_assignment\_charter\_ID**

Tables: Charter\_Assignment

Parameters: @Charter\_ID, @Assignment\_ID

Returns: int (@@RowsAffected)

#### **sp\_deactivate\_charter\_assignment**

Tables: Charter\_Assignment

Parameters: @Assignment\_ID, @Is\_Active

Returns: int (@@RowsAffected)

#### **sp\_activate\_charter\_assignment**

Tables: Charter\_Assignment

Parameters: @Assignment\_ID, @Is\_Active

Returns: int (@@RowsAffected)

## **Charter\_Rider Stored Procedures**

**Written by: Isabella Rosenbohm 10/16/2023**

**Updated by: Isabella Rosenbohm 10/23/2023**

### **sp\_get\_all\_charter\_riders**

Tables: Charter\_Rider

Parameters: None

Returns: Charter\_ID, Dependent\_ID, Is\_Active

### **sp\_get\_charter\_riders**

Tables: Charter\_Rider

Parameters: None

Returns: Charter\_ID, Dependent\_ID, Is\_Active

### **sp\_get\_charter\_rider\_by\_charter\_id**

Tables: Charter\_Accommadation, Charter

Parameters: @Charter\_ID

Returns: Charter.Charter\_ID, Dependent\_ID, Is\_Active

### **sp\_get\_charter\_rider\_by\_dependent\_id**

Tables: Charter\_Accommadation, Dependent

Parameters: @Dependent\_ID

Returns: Charter\_ID, Dependent.Dependent\_ID, Is\_Active

### **sp\_insert\_charter\_rider**

Tables: Charter\_Rider, Charter, Dependent

Parameters: @Dependent.Dependent\_ID, @Charter.Charter\_ID, @Is\_Active

Returns: int (@@RowsAffected)

### **sp\_update\_charter\_rider**

Tables: Charter\_Rider

Parameters: @Dependent\_ID, @Charter\_ID, @Is\_Active

Returns: int (@@RowsAffected)

**sp\_deactivate\_charter\_rider**

Tables: Charter\_Rider  
Parameters: @Is\_Active  
Returns: int (@@RowsAffected)

**sp\_activate\_charter\_rider**

Tables: Charter\_Rider  
Parameters: @Is\_Active  
Returns: int (@@RowsAffected)

## **Charter\_Stop Stored Procedures**

**Written by: Isabella Rosenbohm 10/16/23**

**Updated by: Isabella Rosenbohm 10/23/2023**

### **sp\_get\_charter\_stops**

Tables: Charter\_Stop

Parameters: None

Returns: Charter\_Stop\_ID, Street\_Address, Zip\_Code, Latitude, Longitude, Duration, Description, Is\_Active

### **sp\_get\_inactive\_charter\_stops**

Tables: Charter\_Stop

Parameters: None

Returns: Charter\_Stop\_ID, Street\_Address, Zip\_Code, Latitude, Longitude, Duration, Description, Is\_Active

### **sp\_get\_all\_charter\_stops**

Tables: Charter\_Stop

Parameters: None

Returns: Charter\_Stop\_ID, Street\_Address, Zip\_Code, Latitude, Longitude, Duration, Description, Is\_Active

### **sp\_get\_charter\_stop\_by\_id**

Tables: Charter\_Stop

Parameters: @Charter\_Stop\_ID

Returns: Charter\_Stop\_ID, Street\_Address, Zip\_Code, Latitude, Longitude, Duration, Description, Is\_Active

### **sp\_get\_charter\_stop\_by\_street\_address**

Tables: Charter\_Stop

Parameters: @Street\_Address

Returns: Charter\_Stop\_ID, Street\_Address, Zip\_Code, Latitude, Longitude, Duration, Description, Is\_Active

### **sp\_get\_charter\_stop\_by\_zip\_code**

Tables: Charter\_Stop

Parameters: @Zip\_Code  
Returns: Charter\_Stop\_ID, Street\_Address, Zip\_Code, Latitude, Longitude, Duration, Description, Is\_Active

#### **sp\_insert\_charter\_stop**

Tables: Charter\_Stop  
Parameters: @Charter\_Stop\_ID, @Street\_Address, @Zip\_Code, @Latitude, @Longitude, @Duration, @Description, @Is\_Active  
Returns: Charter\_Stop\_ID

#### **sp\_update\_charter\_stop**

Tables: Charter\_Stop  
Parameters: @Charter\_Stop\_ID,  
@Old\_Street\_Address, @Old\_Zip\_Code, @Old\_Latitude,  
@Old\_Longitude, @Old\_Duration, @Old\_Description, @Old\_Is\_Active,  
@New\_Street\_Address, @New\_Zip\_Code, @New\_Latitude,  
@New\_Longitude, @New\_Duration, @New\_Description, @New\_Is\_Active  
Returns: int (@@RowsAffected)

#### **sp\_deactivate\_charter\_stop**

Tables: Charter\_Stop  
Parameters: @Charter\_ID, @Is\_Active  
Returns: int (@@RowsAffected)

#### **sp\_activate\_charter\_stop**

Tables: Charter\_Stop  
Parameters: @Charter\_ID, @Is\_Active  
Returns: int (@@RowsAffected)

## **Charter Stored Procedures**

**Written by: Isabella Rosenbohm 10/2/23**

**Updated by: Isabella Rosenbohm 11/13/23**

### **sp\_get\_charter\_by\_id**

Tables: Charter

Parameters: @Charter\_ID

Returns: Charter\_ID, Client\_ID, Client\_Is\_Passenger, Rider\_Quantity, Driver\_Needed, Reviewed\_By, Is\_Approved, Date\_Request\_Start, Date\_Request\_End, Is\_Active

### **sp\_get\_charters**

Tables: Charter

Parameters: None

Returns: Charter\_ID, Client\_ID, Client\_Is\_Passenger, Rider\_Quantity, Driver\_Needed, Reviewed\_By, Is\_Approved, Date\_Request\_Start, Date\_Request\_End, Is\_Active

### **sp\_get\_inactive\_charters**

Tables: Charter

Parameters: None

Returns: Charter\_ID, Client\_ID, Client\_Is\_Passenger, Rider\_Quantity, Driver\_Needed, Reviewed\_By, Is\_Approved, Date\_Request\_Start, Date\_Request\_End, Is\_Active

### **sp\_get\_all\_charters**

Tables: Charter

Parameters: None

Returns: Charter\_ID, Client\_ID, Client\_Is\_Passenger, Rider\_Quantity, Driver\_Needed, Reviewed\_By, Is\_Approved, Date\_Request\_Start, Date\_Request\_End, Is\_Active

### **sp\_get\_charters\_where\_is\_approved**

Tables: Charter  
Parameters: @Is\_Approved, @Is\_Active  
Returns: Charter\_ID, Client\_ID, Client\_Is\_Passenger, Rider\_Quantity,  
Driver\_Needed, Reviewed\_By, Is\_Approved, Date\_Request\_Start,  
Date\_Request\_End, Is\_Active

#### sp\_get\_inactive\_charters\_where\_is\_approved

Tables: Charter  
Parameters: @Is\_Approved, @Is\_Active  
Returns: Charter\_ID, Client\_ID, Client\_Is\_Passenger, Rider\_Quantity,  
Driver\_Needed, Reviewed\_By, Is\_Approved, Date\_Request\_Start,  
Date\_Request\_End, Is\_Active

#### sp\_get\_all\_charters\_where\_is\_approved

Tables: Charter  
Parameters: @Is\_Approved  
Returns: Charter\_ID, Client\_ID, Client\_Is\_Passenger, Rider\_Quantity,  
Driver\_Needed, Reviewed\_By, Is\_Approved, Date\_Request\_Start,  
Date\_Request\_End, Is\_Active

#### sp\_insert\_charter

Tables: Charter  
Parameters: @Charter\_ID, @Client\_ID, @Client\_Is\_Passenger, @Rider\_Quantity,  
@Driver\_Needed, @Driver\_ID, @Vehicle\_Type\_ID, @Vehicle\_ID,  
@Reviewed\_By, @Is\_Approved, @Date\_Request\_Start,  
@Date\_Request\_End, @Date\_Issued, @Date\_Returned, @Is\_Active  
Returns: Charter\_ID

#### sp\_update\_charter

Tables: Charter  
Parameters: @Charter\_ID,  
@Old\_Client\_ID, @Old\_Client\_Is\_Passenger, @Old\_Rider\_Quantity,  
@Old\_Driver\_Needed, @Old\_Date\_Request\_Start,  
@Old\_Date\_Request\_End, @Old\_Is\_Active, @New\_Client\_ID,  
@New\_Client\_Is\_Passenger, @New\_Rider\_Quantity,  
@New\_Driver\_Needed, @New\_Date\_Request\_Start,  
@New\_Date\_Request\_End, @New\_Is\_Active  
Returns: int (@@RowsAffected)

**sp\_update\_charter\_approval**

Tables: Charter, Employee  
Parameters: @Is\_Approved, @Charter\_ID, @Reviewed\_By (@Employee.Employee\_ID)  
Returns: int (@@RowsAffected)

**sp\_deactivate\_charter**

Tables: Charter  
Parameters: @Charter\_ID, @Is\_Active  
Returns: int (@@RowsAffected)

**sp\_activate\_charter**

Tables: Charter  
Parameters: @Charter\_ID, @Is\_Active  
Returns: int (@@RowsAffected)

## **Client\_Client\_Role Stored Procedures**

**Initial Creator:** Jared Roberts

**Creation Date:** 2023-10-09

### **sp\_update\_client\_client\_role:**

Tables: Client, Client\_Role  
Parameters: @Client\_ID, @Client\_Role\_ID  
Returns: int (@@RowsAffected)  
          Client\_Credential  
          Initial Creator: Jared Roberts  
          Creation Date: 2023-10-09

### **sp\_insert\_client\_credential:**

Tables: Client\_Credential  
Parameters: @License\_Number, @License\_Type,  
              @License\_Expiration, @Client\_ID, @Permission,  
              @Certified, @Certification\_Description,  
              @Certification\_Date  
Returns: int (@@RowsAffected)

### **sp\_select\_client\_credential\_by\_client\_id:**

Tables: Client\_Credential, Client  
Parameters: @Client\_ID  
Returns: License\_Number, License\_Type, License\_Expiration,  
          Client\_ID, Permission, Certified,  
          Certification\_Description, Certification\_Date

### **sp\_update\_client\_credential:**

Tables: Client\_Credential  
Parameters: @Given\_Name, @Family\_Name, @Middle\_Name, @DOB,  
              @email, @Postal\_Code, @City, @Region, @Address,  
              @Text\_Number, @Voice\_Number, @Active  
Returns: int (@@RowsAffected)

### **sp\_delete\_client\_credential:**

Tables: Client\_Credential, Client  
Parameters: @Client\_ID  
Returns: int (@@RowsAffected)  
          Client\_Role

**Initial Creator:** Jared Roberts

**Creation Date:** 2023-10-09

```
sp_insert_client_role:
    Tables:          Client_Role
    Parameters:      @Client_Role_ID, @Role_Description
    Returns:         int (@@RowsAffected)

sp_select_client_role_by_id:
    Tables:          Client_Role
    Parameters:      @Client_ID
    Returns:         Client_Role_ID, Role_Description, Active

sp_update_client_role:
    Tables:          Client_Role
    Parameters:      @Client_Role_ID, @Role_Description, @Active
    Returns:         int (@@RowsAffected)

sp_delete_client_role:
    Tables:          Client_Role
    Parameters:      @Client_Role_ID
    Returns:         int (@@RowsAffected)

sp_deactivate_client_role:
    Tables:          Client_Role
    Parameters:      @Client_Role_ID
    Returns:         int (@@RowsAffected)

sp_activate_client:
    Tables:          Client_Role
    Parameters:      @Client_Role_ID
    Returns:         int (@@RowsAffected)
```

## **Client Stored Procedures**

**Initial Creator:** Jared Roberts

**Creation Date:** 2023-10-08

**sp\_insert\_client:**

Tables: Client  
Parameters: @Given\_Name, @Family\_Name, @Middle\_Name, @DOB,  
@Email, @Postal\_Code, @City, @Region, @Address,  
@Text\_Number, @Voice\_Number  
Returns: int (@@RowsAffected)

**sp\_select\_client\_by\_id:**

Tables: Client  
Parameters: @Client\_ID  
Returns: Client\_ID, Given\_Name, Family\_Name, Middle\_Name,  
DOB, Email, Postal\_Code, City, Region, Address,  
Text\_Number, Voice\_Number, Active

**sp\_update\_client:**

Tables: Client  
Parameters: @Client\_ID, @Given\_Name, @Family\_Name,  
@Middle\_Name, @DOB, @Email, @Postal\_Code, @City,  
@Region, @Address, @Text\_Number, @Voice\_Number,  
@Active  
Returns: int (@@RowsAffected)

**sp\_deactivate\_client:**

Tables: Client  
Parameters: @Client\_ID  
Returns: int (@@RowsAffected)

**sp\_activate\_client:**

Tables: Client  
Parameters: @Client\_ID  
Returns: int (@@RowsAffected)

## Dependent Stored Procedure Definitions

### Dependent & Client Dependent Roll

#### sp\_select\_active\_dependents\_by\_client\_id

Tables: Dependent, Client, Client\_Dependent\_Roll  
Parameters: @Client\_ID  
Returns: Client.Given\_Name, Client.Family\_Name, Client.Middle\_Name,  
Dependent.Given\_Name, Dependent.Family\_Name,  
Dependent.Middle\_Name, Dependent.DOB,  
Dependent.Emergency\_Contact, Dependent.Emergency\_Phone

#### sp\_select\_active\_dependents

Tables: Dependent, Client, Client\_Dependent\_Roll  
Parameters: None  
Returns: Client.Given\_Name, Client.Family\_Name, Client.Middle\_Name,  
Dependent.Given\_Name, Dependent.Family\_Name,  
Dependent.Middle\_Name, Dependent.DOB,  
Dependent.Emergency\_Contact, Dependent.Emergency\_Phone

#### sp\_insert\_dependent

Tables: Dependent, Client\_Dependent\_Roll  
Parameters: @Client\_ID, @Dependent.Given\_Name, @Dependent.Family\_Name,  
@Dependent.Middle\_Name, @Dependent.DOB,  
@Dependent.Emergency\_Contact, @Dependent.Emergency\_Phone  
Returns: int (@@RowsAffected)

#### sp\_update\_dependent

Tables: Dependent, Client\_Dependent\_Roll  
Parameters: @Client\_ID, @Dependent\_ID, @Dependent.OldGiven\_Name,  
@Dependent.OldFamily\_Name, @Dependent.OldMiddle\_Name,  
@Dependent.OldDOB, @Dependent.Emergency\_Contact,  
@Dependent.OldEmergency\_Phone, @Dependent.NewGiven\_Name,  
@Dependent.NewFamily\_Name, @Dependent.NewMiddle\_Name,  
@Dependent.NewDOB, @Dependent.NewEmergency\_Contact,  
@Dependent.NewEmergency\_Phone  
Returns: int (@@RowsAffected)

### **sp\_deactivate\_dependents**

Tables: Dependent, Client\_Dependent\_Roll  
Parameters: @Client\_ID, @Dependent\_ID  
Returns: int (@@RowsAffected)

### **Dependent Accommodation Stored Procedures**

#### **sp\_select\_active\_accommodations\_by\_dependent\_id**

Tables: Dependent, Dependent\_Accommodation, Accommodation  
Parameters: @Dependent\_ID  
Returns: Dependent\_Accommodation.Active, Dependent\_ID,  
Dependent.Given\_Name, Dependent.Family\_Name,  
Dependent.Middle\_Name, Dependent.DOB,  
Accommodation.Accommodation\_ID, Accommodation.Description

#### **sp\_select\_active\_dependents\_by\_accommodation\_id**

Tables: Dependent, Dependent\_Accommodation, Accommodation  
Parameters: @Accommodation\_ID  
Returns: Dependent\_Accommodation.Active, Dependent.Given\_Name,  
Dependent.Family\_Name, Dependent.Middle\_Name, Dependent.DOB,  
Accommodation.Accommodation\_ID, Accommodation.Description

#### **sp\_select\_dependent\_accommodation**

Tables: Dependent, Dependent\_Accommodation, Accommodation  
Parameters: @Dependent\_ID, @Accommodation\_ID  
Returns: Dependent\_Accommodation.Active, Dependent.Given\_Name,  
Dependent.Family\_Name, Dependent.Middle\_Name, Dependent.DOB,  
Accommodation.Accommodation\_ID, Accommodation.Description

#### **sp\_insert\_dependent\_accommodation**

Table: Dependent\_Accommodation  
Parameters: @Dependent\_ID, @Accommodation\_ID  
Returns: Active

#### **sp\_update\_dependent\_accommodation**

Table: Dependent\_Accommodation

Parameters: @OldDependent\_ID, @OldAccommodation\_ID, @NewDependent\_ID,  
@NewAccommodation\_ID  
Returns: int (@@RowsAffected)

#### **sp\_deactivate\_dependent\_accommodation**

Table: Dependent\_Accommodation  
Parameters: @Dependent\_ID, @Accommodation\_ID  
Returns: int (@@RowsAffected)

### **Driver Maintenance Report Stored Procedures**

**Initial Creator:** James Williams

**Creation Date:** 2023-10-06

**Signed Off By:** Jared Roberts

**Sign Off Date:** 2023-10-09

#### **sp\_insert\_driver\_maintenance\_report:**

Tables: Driver\_Maintenance\_Report, Driver  
Parameters: @Driver\_ID, @VIN, @Date, @Severity, @Description  
Returns: Driver\_Maintenance\_Report\_ID

#### **sp\_deactivate\_driver\_maintenance\_report:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Report\_ID  
Returns: int(@@RowsAffected)

#### **sp\_activate\_driver\_maintenance\_report:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Report\_ID  
Returns: int(@@RowsAffected)

#### **sp\_update\_driver\_maintenance\_report\_severity:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Report\_ID, @NewSeverity  
Returns: int(@@RowsAffected)

**sp\_update\_driver\_maintenance\_report\_description:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Report\_ID, @NewDescription  
Returns: int(@@RowsAffected)

**sp\_delete\_driver\_maintenance\_report:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Report\_ID  
Returns: int(@@RowsAffected)

**sp\_get\_all\_driver\_maintenance\_report\_by\_VIN:**

Tables: Driver\_Maintenance\_Report  
Parameters: @VIN  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_all\_driver\_maintenance\_report\_by\_employee\_id:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Employee\_ID  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_all\_driver\_maintenance\_report\_where\_date:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Date  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_all\_driver\_maintenance\_report\_where\_severity:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Severity  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_driver\_maintenance\_report:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Is\_Active = 1  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_all\_driver\_maintenance\_report\_by\_employee\_id\_and\_date:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Employee\_ID, @Date  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_all\_driver\_maintenance\_report\_by\_VIN\_and\_date:**

Tables: Driver\_Maintenance\_Report  
Parameters: @VIN, @Date  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_driver\_maintenance\_report\_where\_employee\_id:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Employee\_ID, @Is\_Active = 1  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_driver\_maintenance\_report\_by\_VIN:**

Tables: Driver\_Maintenance\_Report  
Parameters: @VIN, @Is\_Active = 1  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_driver\_maintenance\_report\_where\_severity:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Severity, @Is\_Active = 1  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_driver\_maintenance\_report\_where\_severity\_and\_VIN:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Severity, @VIN  
Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_all\_driver\_maintenance\_report\_where\_severity\_and\_driver\_id:**

Tables: Driver\_Maintenance\_Report  
Parameters: @Severity, @Driver\_ID

Returns: Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**sp\_get\_all\_driver\_maintenance\_report\_by\_report\_id:**

Tables: Driver\_Maintenance\_Report

Parameters: @Report\_ID

Returns: (One)Report\_ID, Driver\_ID, Date, VIN, Severity, Description, Is\_Active

**DRIVER LICENSE CLASS Stored Procedures**

Sign Off: Chris Baenzger

**sp\_insert\_driver\_license\_class**

Tables: Driver\_License\_Class

Parameters: @Driver\_License\_Class\_ID

Returns: Driver\_License\_Class\_ID

**sp\_deactivate\_driver\_license\_class**

Tables: Driver\_License\_Class

Parameters: @Driver\_License\_Class\_ID

Returns: int (@@RowsAffected)

**sp\_get\_all\_active\_driver\_license\_classes**

Tables: Driver\_License\_Class

Parameters: NonD

Returns: Driver\_License\_Class\_ID

**sp\_delete\_driver\_license\_class**

Tables: Driver\_License\_Class

Parameters: @Drivers\_License\_Class\_ID

Returns: int(@@RowsAffected)

## Driver Unavailable Stored Procedures

### **sp\_select\_unavailable\_drivers**

Tables: Driver\_Unavailable  
Parameters: None  
Returns: (MANY)Unavailable\_ID, Driver\_ID, Start\_Datetime,  
End\_Datetime, Reason, is\_active

### **sp\_select\_inactive\_unavailable\_drivers**

Tables: Driver\_Unavailable  
Parameters: None  
Returns: (MANY)Unavailable\_ID, Driver\_ID, Start\_Datetime,  
End\_Datetime, Reason, is\_active

### **sp\_select\_all\_unavailable\_drivers**

Tables: Driver\_Unavailable  
Parameters: None  
Returns: (MANY)Unavailable\_ID, Driver\_ID, Start\_Datetime,  
End\_Datetime, Reason, is\_active

### **sp\_select\_unavailable\_driver\_by\_ID**

Tables: Driver\_Unavailable  
Parameters: @UnavailableID  
Returns: (ONE)Unavailable\_ID, Driver\_ID, Start\_Datetime,  
End\_Datetime, Reason, is\_active

### **sp\_select\_unavailable\_drivers\_where\_date\_range\_is**

Tables: Driver\_Unavailable  
Parameters: @StartRange, @EndRange

Returns: (MANY)Unavailable\_ID, Driver\_ID, Start\_Datetime,  
End\_Datetime, Reason, is\_active

#### **sp\_select\_unavailable\_times\_by\_driver\_ID**

Tables: Driver\_Unavailable

Parameters: @DriverID

Returns: (MANY)Unavailable\_ID, Driver\_ID, Start\_Datetime,  
End\_Datetime, Reason, is\_active

#### **sp\_insert\_unavailable\_driver**

Tables: Driver\_Unavailable

Parameters: @Driver\_ID, @Start\_Datetime,  
@End\_Datetime, @Reason

Returns: @@ROWCOUNT

#### **Sp\_update\_unavailable\_driver**

Tables: Driver\_Unavailable

Parameters: @Unavailable\_ID, @OldDriver\_ID,  
@OldStart\_Datetime, @OldEnd\_Datetime, @OldReason,  
@OldIs\_active, @NewDriver\_ID, @NewStart\_Datetime,  
@NewEnd\_Datetime, @NewReason, @NewIs\_active

Returns: @@ROWCOUNT

#### **sp\_deactivate\_unavailable\_driver**

Tables: Driver\_Unavailable

Parameters: @Unavailable\_ID

Returns: @@ROWCOUNT

### **DRIVER CRUD Procedures**

Sign Off: Chris Baenzger

#### **sp\_insert\_driver**

Tables: Driver  
Parameters: @Employee\_ID, @Driver\_License\_Class\_Id,  
@Is\_Active  
Returns: Driver\_ID

#### **sp\_update\_driver**

Tables: Driver  
Parameters: @Driver\_ID, @Employee\_ID, @Driver\_License\_Class,  
@Is\_Active  
Returns: Driver\_ID, Employee\_ID, License\_Class, Is\_Active

#### **sp\_deactivate\_driver**

Tables: Driver  
Parameters: @Driver\_ID  
Returns: int (@@RowsAffected)

#### **sp\_get\_driver\_by\_id**

Tables: Driver  
Parameters: @Driver\_ID  
Returns: Driver\_ID, Employee\_ID, Driver\_License\_Class\_Id,  
Is\_Active

#### **sp\_get\_all\_active\_driver**

Tables: Driver  
Parameters: None  
Returns: Driver\_ID, Employee\_ID, Driver\_License\_Class\_Id,  
Is\_Active

#### **sp\_delete\_driver**

Table: Driver

Parameters:      @Driver\_ID, @Employee\_ID, @Driver\_License\_Class,  
                  @Is\_Active

Returns:          int(@@RowsAffected)

## **Employee Role Stored Procedures Definitions**

### **sp\_insert\_employee\_role**

Tables: employee\_role  
Parameters: @Employee\_ID, @Role\_ID  
Returns: Int(@@rowsAffected)

### **sp\_update\_employee\_role**

Tables: Employee\_Role  
Parameters: @Employee\_ID, @Role\_ID  
Returns: Int (@@rowsaffected)

### **sp\_get\_employee\_role\_by\_employee\_id**

Tables: Employee\_Role  
Parameters: @Employee\_ID  
Returns: Role\_ID

### **sp\_get\_all\_employee\_role**

Tables: Employee\_Role  
Parameters: None  
Returns: Employee\_ID, Role\_ID

### **sp\_deactivate\_employee\_role**

Tables: Employee\_Role  
Parameters: @Employee\_ID, @Role\_ID  
Returns: Employee\_ID, Role\_ID, Is\_Active

## **Employee Stored Procedures**

**Written by: Isabella Rosenbohm 10/4/2023**

**Updated by: Isabella Rosenbohm 10/23/2023**

### **sp\_select\_employee\_by\_id**

Tables: Employee

Parameters: @Employee\_ID

Returns: Given\_Name, Family\_Name, Address, Address2, City, State, Country, Zip, Phone\_Number, Email, Position, Is\_Active

### **sp\_select\_employees**

Tables: Employee

Parameters: None

Returns: Employee\_ID, Given\_Name, Family\_Name, Address, Address2, City, State, Country, Zip, Phone\_Number, Email, Position, Is\_Active

### **sp\_select\_inactive\_employees**

Tables: Employee

Parameters: None

Returns: Employee\_ID, Given\_Name, Family\_Name, Address, Address2, City, State, Country, Zip, Phone\_Number, Email, Position, Is\_Active

### **sp\_select\_all\_employees**

Tables: Employee

Parameters: None

Returns: Employee\_ID, Given\_Name, Family\_Name, Address, Address2, City, State, Country, Zip, Phone\_Number, Email, Position, Is\_Active

### **sp\_select\_employee\_by\_position**

Tables: Employee

Parameters: @Position

Returns: Employee\_ID, Given\_Name, Family\_Name, Address, Address2, City, State, Country, Zip, Phone\_Number, Email, Is\_Active

**sp\_select\_employee\_by\_email**

Tables: Employee  
Parameters: @Email  
Returns: Employee\_ID, Given\_Name, Family\_Name, Address, Address2, City, State, Country, Zip, Phone\_Number, Position, Is\_Active

**sp\_insert\_employee**

Tables: Employee  
Parameters: @Given\_Name, @Family\_Name, @Address, @Address2, @City, @State, @Country, @Zip, @Phone\_Number, @Email, @Position  
Returns: Employee\_ID

**sp\_update\_employee**

Tables: Employee  
Parameters: @Employee\_ID,  
@Old\_Given\_Name, @Old\_Family\_Name, @Old\_Address,  
@Old\_Address2, @Old\_City, @Old\_State, @Old\_Country, @Old\_Zip,  
@Old\_Phone\_Number, @Old\_Email, @Old\_Position, @Old\_Is\_Active,  
@New\_Given\_Name, @New\_Family\_Name, @New\_Address,  
@New\_Address2, @New\_City, @New\_State, @New\_Country, @New\_Zip,  
@New\_Phone\_Number, @New\_Email, @New\_Position, @New\_Is\_Active  
Returns: int (@@RowsAffected)

**sp\_deactivate\_employee**

Tables: Employee  
Parameters: @Employee\_ID  
Returns: int (@@RowsAffected)

## Inspection Report Stored Procedures

Initial Creator: Ben Collins

Creation Date: 2023-10-16

### sp\_insert\_inspection\_report

Tables: Inspection\_Report

Parameters: @Driver\_ID, @VIN, @Service\_Order\_ID, @Oil\_Level, @Tire\_Pressure,  
@Front\_Left\_Turn\_Signal, @Front\_Right\_Turn\_Signal,  
@Rear\_Left\_Turn\_Signal, @Rear\_Right\_Turn\_Signal,  
@Left\_Brake\_Light, @Right\_Brake\_Light, @Windshield\_Washer\_Fluid,  
@Problem\_Description, @Is\_Active

Returns: Inspection\_Report\_ID

### sp\_select\_inspection\_report\_by\_Inspection\_Report\_ID

Tables: Inspection\_Report

Parameters: @Inspection\_Report\_ID

Returns: Driver\_ID, VIN, Service\_Order\_ID, Oil\_Level, Tire\_Pressure,  
Front\_Left\_Turn\_Signal, Front\_Right\_Turn\_Signal,  
Rear\_Left\_Turn\_Signal, Rear\_Right\_Turn\_Signal, Left\_Brake\_Light,  
Right\_Brake\_Light, Windshield\_Washer\_Fluid, Problem\_Description,  
Is\_Active

### sp\_select\_inspection\_report\_by\_Driver\_ID

Tables: Inspection\_Report

Parameters: @Driver\_ID

Returns: Inspection\_Report\_ID, VIN, Service\_Order\_ID, Oil\_Level, Tire\_Pressure,  
Front\_Left\_Turn\_Signal, Front\_Right\_Turn\_Signal,  
Rear\_Left\_Turn\_Signal, Rear\_Right\_Turn\_Signal, Left\_Brake\_Light,  
Right\_Brake\_Light, Windshield\_Washer\_Fluid, Problem\_Description,  
Is\_Active

### **sp\_select\_inspection\_report\_by\_VIN**

Tables:      Inspection\_Report  
Parameters:    @VIN  
Returns:      Inspection\_Report\_ID, Driver\_ID, Service\_Order\_ID, Oil\_Level,  
                 Tire\_Pressure, Front\_Left\_Turn\_Signal, Front\_Right\_Turn\_Signal,  
                 Rear\_Left\_Turn\_Signal, Rear\_Right\_Turn\_Signal, Left\_Brake\_Light,  
                 Right\_Brake\_Light, Windshield\_Washer\_Fluid, Problem\_Description,  
                 Is\_Active

### **sp\_select\_inspection\_report\_by\_Service\_Order\_ID**

Tables:      Inspection\_Report  
Parameters:    @Service\_Order\_ID  
Returns:      Inspection\_Report\_ID, Driver\_ID, VIN, Oil\_Level, Tire\_Pressure,  
                 Front\_Left\_Turn\_Signal, Front\_Right\_Turn\_Signal,  
                 Rear\_Left\_Turn\_Signal, Rear\_Right\_Turn\_Signal, Left\_Brake\_Light,  
                 Right\_Brake\_Light, Windshield\_Washer\_Fluid, Problem\_Description,  
                 Is\_Active

### **sp\_update\_inspection\_report**

Table:      Inspection\_Report  
Parameters:    @New\_Inspection\_Report\_ID, @New\_Driver\_ID, @New\_VIN,  
                 @New\_Service\_Order\_ID, @New\_Oil\_Level, @New\_Tire\_Pressure,  
                 @New\_Front\_Left\_Turn\_Signal, @New\_Front\_Right\_Turn\_Signal,  
                 @New\_Rear\_Left\_Turn\_Signal, @New\_Rear\_Right\_Turn\_Signal,  
                 @New\_Left\_Brake\_Light, @New\_Right\_Brake\_Light,  
                 @New\_Windshield\_Washer\_Fluid, @New\_Problem\_Description,  
                 @New\_Is\_Active,  
                 @Old\_Inspection\_Report\_ID, @Old\_Driver\_ID, @Old\_VIN,  
                 @Old\_Service\_Order\_ID, @Old\_Oil\_Level, @Old\_Tire\_Pressure,  
                 @Old\_Front\_Left\_Turn\_Signal, @Old\_Front\_Right\_Turn\_Signal,  
                 @Old\_Rear\_Left\_Turn\_Signal, @Old\_Rear\_Right\_Turn\_Signal,  
                 @Old\_Left\_Brake\_Light, @Old\_Right\_Brake\_Light,  
                 @Old\_Windshield\_Washer\_Fluid, @Old\_Problem\_Description,  
                 @Old\_Is\_Active  
Returns:      int (@@RowsAffected)

**sp\_deactivate\_inspection\_report**

Table: Inspection\_Report  
Parameters: @Inspection\_Report\_ID  
Returns: int (@@RowsAffected)

**sp\_delete\_inspection\_report**

Table: Inspection\_Report  
Parameters: @Inspection\_Report\_ID  
Returns: int (@@RowsAffected)

## **Login Stored Procedures**

**Initial Creator:** Jared Roberts  
**Creation Date:** 2023-10-09

### **sp\_insert\_login:**

Tables: Login, Client  
Parameters: @Username, @Password\_Hash, @Client\_ID,  
@Security\_Question1, @Security\_Response1,  
@Security\_Question2, @Security\_Response2,  
@Security\_Question3, @Security\_Response3  
Returns: int (@@RowsAffected)

### **sp\_select\_login\_by\_username:**

Tables: Login, Client\_ID  
Parameters: @Username  
Returns: Username, Password\_Hash, Client\_ID,  
@Security\_Question1, @Security\_Response1,  
@Security\_Question2, @Security\_Response2,  
@Security\_Question3, @Security\_Response3

### **sp\_select\_login\_by\_client\_id:**

Tables: Login, Client\_ID  
Parameters: @Client\_ID  
Returns: Username, Password\_Hash, Client\_ID  
@Security\_Question1, @Security\_Response1,  
@Security\_Question2, @Security\_Response2,  
@Security\_Question3, @Security\_Response3

### **sp\_update\_login:**

Tables: Client\_ID, Client  
Parameters: @Username, @Password\_Hash, @Client\_ID,  
@Security\_Question1, @Security\_Response1  
@Security\_Question2, @Security\_Response2,  
@Security\_Question3, @Security\_Response3  
Returns: int (@@RowsAffected)

### **sp\_delete\_login:**

Tables: Client\_ID, Client  
Parameters: @Username, @Password\_Hash, @Client\_ID  
Returns: int (@@RowsAffected)

## Maintenance Schedule Stored Procedures

### Jacob Rugger

#### **sp\_get\_all\_active\_scheduled\_maintenance**

Tables: Maintenance\_Schedule, Model\_Lookup, Service\_Type  
Parameters: None  
Returns: Maintenance\_Schedule\_ID,  
Maintenance\_Schedule.Model\_Lookup\_ID,  
Model\_Lookup.VIN, Service\_Type.Service\_Type\_ID,  
Schedule\_Date, Is\_Completed, Is\_Active

#### **sp\_get\_scheduled\_maintenance\_by\_id**

Tables: Maintenance\_Schedule, Model\_Lookup, Service\_Type  
Parameters: @Maintenance\_Schedule\_ID  
Returns: Maintenance\_Schedule\_ID,  
Maintenance\_Schedule.Model\_Lookup\_ID,  
Model\_Lookup.VIN, Service\_Type.Service\_Type\_ID,  
Schedule\_Date, Is\_Completed, Is\_Active

#### **sp\_get\_scheduled\_maintenance\_by\_date**

Tables: Maintenance\_Schedule, Model\_Lookup, Service\_Type  
Parameters: @Schedule\_Date  
Returns: Maintenance\_Schedule\_ID,  
Maintenance\_Schedule.Model\_Lookup\_ID,  
Model\_Lookup.VIN, Service\_Type.Service\_Type\_ID,  
Schedule\_Date, Is\_Completed, Is\_Active

#### **sp\_get\_scheduled\_maintenance\_by\_model**

Tables: Maintenance\_Schedule, Model\_Lookup, Service\_Type  
Parameters: @Model\_Lookup\_ID  
Returns: Maintenance\_Schedule\_ID,  
Maintenance\_Schedule.Model\_Lookup\_ID,  
Model\_Lookup.VIN, Service\_Type.Service\_Type\_ID,  
Schedule\_Date, Is\_Completed, Is\_Active

#### **sp\_get\_scheduled\_maintenance\_by\_service\_type**

Tables: Maintenance\_Schedule, Model\_Lookup, Service\_Type  
Parameters: @Maintenance\_Type\_ID  
Returns: Maintenance\_Schedule\_ID,  
Maintenance\_Schedule.Model\_Lookup\_ID,  
Model\_Lookup.VIN, Service\_Type.Service\_Type\_ID,  
Schedule\_Date, Is\_Completed, Is\_Active

#### **sp\_insert\_scheduled\_maintenance**

Tables: Maintenance\_Schedule  
Parameters: @Maintenance\_Schedule\_ID, @Model\_Lookup\_ID,  
@Service\_Type\_ID, @Schedule\_Date, @Is\_Completed  
Returns: Maintenance\_Schedule\_ID

#### **sp\_update\_maintenance\_schedule**

Tables: Maintenance\_Schedule  
Parameters: @Maintenance\_Schedule\_ID, @Model\_Lookup\_ID,  
@Service\_Type\_ID, @Schedule\_Date, @Is\_Completed  
Returns: int (@@RowsAffected)

#### **sp\_mark\_scheduled\_maintenance\_complete**

Tables: Maintenance\_Schedule  
Parameters: @Maintenance\_Schedule\_ID, @Is\_Completed  
Returns: int (@@RowsAffected)

#### **sp\_deactivate\_scheduled\_maintenance**

Tables: Maintenance\_Schedule  
Parameters: @Maintenance\_Schedule\_ID  
Returns: int (@@RowsAffected)

#### **sp\_delete\_scheduled\_maintenance**

Tables: Maintenance\_Schedule  
Parameters: @Maintenance\_Schedule\_ID  
Returns: int (@@RowsAffected)

## Model Compatibility Stored Procedures

### sp\_select\_compatible\_model

Tables: Model\_Lookup\_ID  
Parameters: @Model\_Lookup.Model\_Lookup\_ID,  
@Parts\_Inventory.Parts\_Inventory\_ID  
Returns: Parts\_ID, Model\_ID

### sp\_select\_VIN

Tables: Model\_Lookup, Vehicle  
Parameters: @Model\_Lookup\_ID  
Returns: (Scalar) Model\_LookupID, VIN

### sp\_insert\_compatible\_model

Tables: Model\_Compatibility  
Parameters: @Model\_Lookup\_ID, @Parts\_Inventory\_ID  
Returns: Model\_Compatibility (scope identity())

### sp\_update\_compatible\_model

Table: Model\_Compatibility  
Parameters: @Model\_Lookup\_ID, @Parts\_Inventory\_ID  
Returns: int (@@RowsAffected)

### sp\_deactivate\_compatible\_model

Table: Model\_Compatibility  
Parameters: @Model\_Compatibility  
Returns: int (@@RowsAffected)

### sp\_delete\_compatible\_model

Table: Model\_Compatibility  
Parameters: None  
Returns: Model\_Lookup\_ID, Part\_Inventory\_ID, Model\_Lookup.is\_Active

**Model\_Lookup Stored Procedures**  
**Initial Creation by Steven Sanchez**  
**Date Created: 2023/10/08**

**sp\_insert\_model\_lookup**

Tables: Model\_Lookup  
Parameters: @Model\_Lookup\_ID, @VIN  
Returns: [int](SCOPE\_IDENTITY())

**sp\_update\_model\_lookup**

Tables: model\_lookup  
Parameters: @Model\_Lookup\_ID, @VIN, @oldModel\_Lookup\_ID,  
@oldVIN  
Returns: int(@@RowsAffected)

**sp\_deactivate\_model\_lookup**

Tables: Model\_Lookup  
Parameters: @Model\_Lookup\_ID  
Returns: int (@@RowsAffected)

**sp\_get\_model\_lookup\_by\_id**

Tables: Model\_Lookup  
Parameters: @Model\_Lookup\_ID  
Returns: Model\_Lookup\_ID, VIN, is\_active

**sp\_get\_all\_active\_model\_lookup**

Tables: Model\_Lookup  
Parameters: None  
Returns: Model\_Lookup\_ID, VIN, is\_active  
**Notification-Related Stored Procedures**

**sp\_select\_non\_viewed\_notifications**

Tables: Notification  
Parameters: None  
Returns: Notification\_ID, Client\_ID, Title, Notification\_Body, Time\_Sent

**sp\_select\_notification\_by\_id**

Tables: Notification  
Parameters: @Notification\_ID  
Returns: Notification\_ID, Client\_ID, Title, Notification\_Body, Time\_Sent,

Viewed, Active

#### **sp\_insert\_notification**

Tables: Notification  
Parameters: @Client\_ID, @Title, @Notification\_Body  
Returns: Notification\_ID

#### **sp\_update\_notification**

Table: Notification  
Parameters: @Client\_ID, @Title, @Notification\_Body, @Viewed  
Returns: int (@@RowsAffected)

#### **sp\_notification\_update\_as\_viewed**

Table: Notification  
Parameters: @Notification\_ID  
Returns: int (@@RowsAffected)

#### **sp\_deactivate\_notification**

Table: Notification  
Parameters: @Noification\_ID  
Returns: int (@@RowsAffected)

#### **sp\_delete\_notification**

Table: Notification  
Parameters: @Notificaion\_ID  
Returns: int (@@RowsAffected)

## Packing Slip Line Items Stored Procedure Definitions

### sp\_get\_lines\_by\_slip\_ID

Tables: Packing\_Slip\_Line\_Items  
Parameters: @slip\_id (can be incomplete)  
Returns: (MANY)Packing\_Slip\_ID, Qty\_Recieved,  
Parts\_Inventory\_ID

### sp\_get\_lines\_by\_parts\_inventory\_id

Tables: Packing\_Slip\_Line\_Items  
Parameters: @part\_id  
Returns: (MANY)Packing\_Slip\_ID, Qty\_Recieved,  
Parts\_inventory\_ID

### sp\_insert\_slip\_line\_item

Tables: Packing\_Slip\_Line\_Items  
Parameters: @packing\_slip\_id, @qty\_recieved,  
@part\_id  
Returns: @@rowcount

### sp\_deactivate\_slip\_line\_item

Tables: Packing\_Slip\_Line\_Items  
Parameters: @packing\_slip\_id  
Returns: @@rowcount

### sp\_delete\_slip\_line\_item

Tables: Packing\_Slip\_Line\_Items  
Parameters: @packing\_slip\_id, @parts\_inventory\_id  
Returns: @@rowcount

### sp\_update\_slip\_line\_item

Tables: Packing\_Slip\_Line\_Items  
Parameters: @packing\_slip\_id, @old\_parts\_inventory\_id,  
@old\_qty\_recieved, @new\_parts\_inventory\_id, @new\_qty\_recieved

Returns: @@rowcount

**sp\_get\_line\_by\_parts\_id\_and\_slip\_id**

Tables: Packing\_Slip\_Line\_Items

Parameters: @packing\_slip\_id, @parts\_inventory\_id,

Returns: (ONE)Packing\_slip\_id, parts\_inventory\_id,

Qty\_recieved

**Ride Stored Procedures**  
**Sign Off: Chris Baenzger**

**sp\_insert\_ride**

Tables: Ride

Parameters: @Ride\_ID, @User\_ID, @Service\_ID, @Scheduled\_Time,  
@Service\_VIN, @Pickup\_Location,  
@Dropoff\_Location, @Scheduled\_Pickup\_Time,  
@Estimated\_Dropoff\_Time, @Actual\_Pickup\_Time  
@Actual\_Dropoff\_Time, @Canceled, @Requested

Returns: Ride\_ID

**sp\_update\_ride**

Tables: Ride

Parameters: @Ride\_ID, @User\_ID, @Service\_ID,  
@Scheduled\_Time, @Service\_VIN,  
@Pickup\_Location, @Dropoff\_Location,  
@Scheduled\_Pickup\_Time,  
@Estimated\_Dropoff\_Time,  
@Actual\_Pickup\_Time @Actual\_Dropoff\_Time,  
@Canceled,@Requested

Returns: Ride\_ID, User\_ID, Service\_ID, Scheduled\_Time,  
Service\_VIN, Pickup\_Location,  
Dropoff\_Location,  
Scheduled\_Pickup\_Time, Estimated\_Dropoff\_Time,  
Actual\_Pickup\_Time Actual\_Dropoff\_Time, Canceled,  
Requested

**sp\_deactivate\_ride**

Tables: Ride

Parameters: @Ride\_ID

Returns: int (@@RowsAffected)

#### **sp\_get\_ride\_by\_id**

Tables: Ride

Parameters: @Ride\_ID

Returns: Ride\_ID, User\_ID, Service\_ID, Scheduled\_Time,  
Service\_VIN, Pickup\_Location,  
Dropoff\_Location, Scheduled\_Pickup\_Time  
Estimated\_Dropoff\_Time, Actual\_Pickup\_Time  
Actual\_Dropoff\_Time, Canceled, Requested

#### **sp\_get\_all\_active\_rides**

Tables: Ride

Parameters: None

Returns: Ride\_ID, User\_ID, Service\_ID, Scheduled\_Time,  
Service\_VIN, Pickup\_Location,  
Dropoff\_Location, Scheduled\_Pickup\_Time,  
Estimated\_Dropoff\_Time,  
Actual\_Pickup\_Time Actual\_Dropoff\_Time, Canceled,  
Requested

#### **sp\_get\_rides\_by\_time\_and\_location**

Tables: Ride, Client, Vehicle

Parameters: @Service\_VIN, @Scheduled\_Pickup\_time

Returns: Ride\_ID, User\_ID, Service\_ID,  
Pickup\_Location, Dropoff\_Location

sp\_

#### **sp\_delete\_ride**

Tables: Ride

Parameters: @Ride\_ID

Returns: int(@@RowsAffected)

## Role Stored Procedures Definitions

### p\_insert\_driver

Tables: Role  
Parameters: @Role\_ID, @Is\_Active  
Returns: Role\_ID

### sp\_update\_driver

Tables: Driver  
Parameters: @Role\_ID, @Is\_Active  
Returns: Driver\_ID, Is\_Active

### sp\_deactivate\_role

Tables: Role  
Parameters: @Role\_ID  
Returns: int (@@RowsAffected)

### sp\_get\_role\_by\_id

Tables: Role  
Parameters: @Role\_ID  
Returns: Role\_ID, Is\_Active

### sp\_get\_all\_active\_role

Tables: Role  
Parameters: None  
Returns: Role\_ID, Is\_Active

## route\_stop Stored Procedures Bill Tobias 2023/10/30 [revised]

### sp\_insert\_route\_stop

TABLES: [route\_stop]  
 PARAMETERS: (1) @route\_id int  
               (2) @stop\_id int  
               (3) @route\_stop\_number int  
               (4) @start\_offset int  
               (5) @is\_active bit  
 RETURNS: SCOPE\_IDENTITY() [route\_id] int

#### **sp\_update\_route\_stop**

TABLES: [route\_stop]  
 PARAMETERS: (1) @route\_id int  
               (2) @stop\_id int  
               (3) @route\_stop\_number int  
               (4) @start\_offset int  
               (5) @is\_active bit  
               (6) @old\_route\_id int  
               (7) @old\_stop\_id int  
               (8) @old\_route\_stop\_number int  
               (9) @old\_start\_offset int  
               (10) @old\_is\_active bit  
 RETURNS: @@ROWCOUNT [rows\_affected] int

#### **sp\_select\_route\_stop\_by\_route\_id**

TABLES: (1) route\_stop  
               (2) stop  
 PARAMETERS: @route\_id int  
 RETURNS: [route\_id],  
               [stop\_id],  
               [route\_stop\_number],  
               [start\_offset],  
               [route\_stop\_is\_active],  
               [street\_address],

[zip\_code],  
[latitude],  
[longitude],  
[stop\_is\_active]

**sp\_select\_route\_stop\_by\_route\_id\_stop\_id**

TABLES: (1) route\_stop

(2) stop

PARAMETERS: (1) @route\_id int

(2) @stop\_id int

RETURNS: [route\_id],

[stop\_id],  
[route\_stop\_number],  
[start\_offset],  
[route\_stop\_is\_active],  
[street\_address],  
[zip\_code],  
[latitude],  
[longitude],  
[stop\_is\_active]

**sp\_select\_route\_stop\_by\_stop\_id**

TABLES: (1) route\_stop

(2) stop

PARAMETERS: (1) @stop\_id int

RETURNS: [route\_id],

[stop\_id],  
[route\_stop\_number],  
[start\_offset],  
[route\_stop\_is\_active],  
[street\_address],  
[zip\_code],

[latitude],  
[longitude],  
[stop\_is\_active]

#### **sp\_deactivate\_route\_stop\_by\_route\_id**

TABLES: (1) route\_stop  
PARAMETERS: (1) @route\_id int  
RETURNS: @@ROWCOUNT [rows\_affected] int

#### **sp\_deactivate\_route\_stop\_by\_stop\_id**

TABLES: (1) route\_stop  
PARAMETERS: (1) @stop\_id int  
RETURNS: @@ROWCOUNT [rows\_affected] int

#### **sp\_deactivate\_route\_stop\_by\_route\_id\_stop\_id**

TABLES: (1) route\_stop  
PARAMETERS: (1) @route\_id int  
RETURNS: @@ROWCOUNT [rows\_affected] int

#### **sp\_activate\_route\_stop\_by\_route\_id**

TABLES: (1) route\_stop  
PARAMETERS: (1) @route\_id int  
RETURNS: @@ROWCOUNT [rows\_affected] int

#### **sp\_activate\_route\_stop\_by\_stop\_id**

TABLES: (1) route\_stop  
PARAMETERS: (1) @stop\_id int  
RETURNS: @@ROWCOUNT [rows\_affected] int

**sp\_activate\_route\_stop\_by\_route\_id\_stop\_id**

TABLES: (1) route\_stop

PARAMETERS: (1) @route\_id int

RETURNS: @@ROWCOUNT [rows\_affected] int

**route [v. 3.1] Stored Procedures**

Bill Tobias 2023/10/30 [revised]

**sp\_deactivate\_route\_by\_id**

TABLES: [route]

PARAMETERS: @route\_id int

RETURNS: @@ROWCOUNT [rows\_affected]

**sp\_activate\_route\_by\_id**

TABLES: [route]

PARAMETERS: @route\_id int

RETURNS: @@ROWCOUNT [rows\_affected]

**sp\_get\_all\_routes**

TABLES: [route]

PARAMETERS: NONE

RETURNS: route\_id,

route\_name,

route\_start\_time,

route\_cycle,

route\_end\_time,

days\_of\_service,

is\_active

**sp\_get\_route\_by\_id**

TABLES: [route]

PARAMETERS: @route\_id int  
RETURNS: route\_id,  
route\_name,  
route\_start\_time,  
route\_cycle,  
route\_end\_time,  
days\_of\_service,  
is\_active

#### **sp\_get\_routes**

TABLES: [route]  
PARAMETERS: NONE  
RETURNS: route\_id,  
route\_name,  
route\_start\_time,  
route\_cycle,  
route\_end\_time,  
days\_of\_service,  
is\_active

#### **sp\_get\_inactive\_routes**

TABLES: [route]  
PARAMETERS: NONE  
RETURNS: route\_id,  
route\_name,  
route\_start\_time,  
route\_cycle,  
route\_end\_time,  
days\_of\_service,  
is\_active

#### **sp\_get\_active\_routes**

TABLES: [route]

PARAMETERS: NONE

RETURNS:

```
route_id,  
route_name,  
route_start_time,  
route_cycle,  
route_end_time,  
days_of_service,  
is_active
```

### **sp\_insert\_route**

TABLES:

```
[route]
```

PARAMETERS:

```
(1) @route_name  
(2) @route_start_time  
(3) @route_cycle  
(4) @route_end_time  
(5) @days_of_service  
(6) @is_active
```

RETURNS:

```
SCOPE_IDENTITY() [route_id] int
```

### **sp\_update\_route**

TABLES:

```
[route]
```

PARAMETERS:

```
(1) @route_id  
(2) @route_name  
(3) @route_start_time  
(4) @route_cycle  
(5) @route_end_time  
(6) @days_of_service  
(7) @is_active  
(8) @old_route_name  
(9) @old_route_start_time  
(10) @old_route_cycle  
(11) @old_route_end_time
```

(12) @old\_days\_of\_service

(13) @old\_is\_active

RETURNS:  
@@ROWCOUNT [rows\_affected] int

## **Service Vehicle Stored Procedures**

**Sign Off: Chris Baenzger**

### **sp\_insert\_service\_vehicle**

Tables: Service\_Vehicle  
Parameters: @Service\_VIN, @VIN, @Service\_ID  
Returns: Service\_VIN

### **sp\_update\_service\_vehicle**

Tables: Service\_Vehicle  
Parameters: @Service\_VIN, @VIN, @Service\_ID  
Returns: Service\_VIN, VIN, Service\_ID

### **sp\_deactivate\_service\_vehicle**

Tables: Service\_Vehicle  
Parameters: @Service\_VIN  
Returns: int (@@RowsAffected)

### **sp\_get\_service\_vehicle\_by\_id**

Tables: Service\_Vehicle  
Parameters: @Service\_VIN  
Returns: Service\_VIN, VIN, Service\_ID

### **sp\_get\_all\_active\_service\_vehicles**

Tables: Service\_Vehicle  
Parameters: None  
Returns: Service\_VIN, VIN, Service\_ID

### **sp\_delete\_service\_vehicle**

Tables: Service\_Vehicle

Parameters: @Service\_VIN

Returns: int(@@RowsAffected)

## **Service Stored Procedures**

**Sign Off: Chris Baenzger**

### **sp\_insert\_service**

Tables: Service  
Parameters: @Service\_ID, @Type  
Returns: Service\_ID

### **sp\_update\_service**

Tables: Service  
Parameters: @Service\_ID, @Type  
Returns: Service\_ID, Type

### **sp\_deactivate\_service**

Tables: Service  
Parameters: @Service\_ID  
Returns: int (@@RowsAffected)

### **sp\_get\_service\_by\_id**

Tables: Service  
Parameters: @Service\_ID  
Returns: Service\_ID, Type

### **sp\_get\_all\_active\_services**

Tables: Service  
Parameters: None  
Returns: Service\_ID, Type

### **sp\_delete\_service**

Tables: Service  
Parameters: @Service\_ID

Returns: `int(@@RowsAffected)`

## Special Inspection-Related Stored Procedures

### **Sp\_get\_all\_active\_special\_inspection**

Tables:                   Special\_Inspection, service\_orders  
Parameters:               None  
Returns:                  Special\_inspection\_id,  
                            service\_orders.Service\_details,  
                            inspection\_description,  
                            date, employee\_ID

### **Sp\_get\_special\_inspection\_by\_special\_inspection\_id**

Tables:                   Special\_Inspection, service\_orders  
Parameters:               @special\_inspection\_id  
Returns:                  Special\_inspection\_id,  
                            service\_orders.Service\_details,  
                            inspection\_description,  
                            date, employee\_id, is\_active

### **Sp\_get\_active\_special\_inspection\_by\_special\_service\_id**

Tables:                   Special\_inspection, special\_service  
Parameters:               @special\_service\_id  
Returns:                  Special\_inspection\_id, special\_service\_id,  
                            inspection\_description, date,  
                            Employee\_id

### **Sp\_get\_active\_special\_inspection\_by\_employee\_id**

Tables:                   Special\_inspection, employee  
Parameters:               @employee\_id  
Returns:                  Special\_inspection\_id, special\_service\_id,  
                            inspection\_description, date,  
                            Employee\_id

**Sp\_insert\_special\_inspection**

Tables:           Special\_inspection  
Parameters:       @special\_service\_id, @inspection\_description,  
                    @date, @employee\_id  
Returns:           Special\_inspection\_id(scope identity())

**Sp\_update\_special\_inspection**

Tables:           Special\_inspection  
Parameters:       @Special\_inspection\_id  
                    @special\_service\_id, @inspection\_description,  
                    @date, @employee\_id  
                    @special\_service\_id, @inspection\_description,  
                    @date, @employee\_id  
  
Returns:           Int (@@rowsAffected)

**Sp\_deactivate\_special\_inspection**

Tables:           Special\_inspection  
Parameters:       @special\_inspeciton\_id  
Returns:           Int (@@rowsaffected)

**Sp\_activate\_special\_inspection**

Tables:           Special\_inspection  
Parameters:       @special\_inspeciton\_id  
Returns:           Int (@@rowsaffected)

## **Special\_Service\_detail Related Stored Procedures**

### **sp\_get\_all\_active\_special\_service\_detail**

Tables: Special\_Service\_Detail, Service\_Orders

Parameters: None

Returns: Special\_service\_ID,

Service\_orders.Service\_details,  
event\_description, priority

### **sp\_get\_special\_service\_order\_by\_special\_service\_id**

Tables: Special\_Service\_Detail, Service\_Orders

Parameters: @Special\_service\_detail\_ID

Returns: Special\_service\_ID, Service.Service\_details,  
event\_description, priority, is\_active

### **sp\_get\_active\_special\_service\_order\_for\_service\_order**

Tables: Special\_Service\_detail, Service\_orders

Parameters: Service\_order\_id

Returns: Special\_service\_detial\_id, service\_order\_id,  
event\_descritpion, priority

### **sp\_insert\_special\_service\_order**

Tables: Special\_Service\_Detail

Parameters: @Service\_Order\_ID, @event\_description, @priority

Returns: Social\_service\_detail\_ID (scope\_identity())

### **sp\_update\_special\_service\_order**

Tables: Special\_Service\_Order, Special\_service\_detail,  
Special\_inspection, bid

Parameters: @Special\_Service\_detail\_ID

@oldService\_Order\_ID, @oldevent\_description,  
@oldpriority

@newService\_Order\_ID, @newevent\_description,  
@newpriority

Returns: Int (@@RowsAffected)

**sp\_deactivate\_special\_service\_order**

Tables: Special\_Service\_Order

Parameters: @special\_service\_detail\_ID

Returns: Int (@@RowsAffected)

**sp\_activate\_special\_service\_order**

Tables: Special\_Service\_Order

Parameters: @special\_service\_detail\_ID

Returns: Int (@@RowsAffected)

## Special Work Order Related Stored Procedures

### **sp\_get\_active\_special\_work\_order**

Tables: Special\_work\_orders, service\_orders,  
special\_approval

Parameters: None

Returns: Special\_work\_order\_id,  
service\_orders.service\_order\_details,  
special\_approval.approval\_descritpion,  
work\_order\_description, date

### **sp\_get\_special\_work\_order\_by\_special\_work\_order\_id**

Tables: Special\_work\_orders, service\_orders,  
special\_approval

Parameters: @special\_work\_order\_id

Returns: service\_orders.service\_order\_details,  
special\_approval.approval\_descritpion,  
work\_order\_description, date, is\_active

### **sp\_get\_active\_special\_work\_order\_for\_special\_approval\_id**

Tables: Special\_approval, special\_work\_order

Parameters: @Special\_approval\_id

Returns: Special\_work\_order\_id, special\_approval\_id,  
work\_order\_description, date, is\_active

### **sp\_insert\_special\_work\_order**

Tables: Special\_work\_order

Parameters: @Special\_approval\_id, @work\_order\_description,  
@date

Returns: Special\_work\_order\_id (scope\_identity())

### **sp\_update\_special\_work\_order**

Tables:

Parameters: @Special\_work\_order\_id

```
@oldSpecial_approval_id, @oldwork_order_description,  
@olddate  
@newSpecial_approval_id, @newwork_order_description,  
@newdate  
Returns: Int (@@rowsAffected)
```

#### **sp\_deactivate\_special\_work\_order**

```
Tables: Special_work_order  
Parameters: @Special_work_order_id  
Returns: Int (@@rowsAffected)
```

#### **sp\_activate\_special\_work\_order**

```
Tables: Special_work_order  
Parameters: @Special_work_order_id  
Returns: Int (@@rowsAffected)
```

## Stop Stored Procedures

Bill Tobias    2023/10/30 [revised]

### sp\_deactivate\_stop\_by\_id

TABLES: [stop]  
PARAMETERS: @stop\_id int  
RETURNS: @@ROWCOUNT [rows\_affected]

### sp\_activate\_stop\_by\_id

TABLES: [stop]  
PARAMETERS: @stop\_id int  
RETURNS: @@ROWCOUNT [rows\_affected]

### sp\_get\_all\_stops

TABLES: [stop]  
PARAMETERS: NONE  
RETURNS: stop\_id,  
             street\_address,  
             zip\_code,  
             latitude,  
             longitude,  
             is\_active

### sp\_get\_inactive\_stops

TABLES: [stop]  
PARAMETERS: NONE  
RETURNS: stop\_id,  
             street\_address,  
             zip\_code,  
             latitude,  
             longitude,  
             is\_active

**sp\_get\_active\_stops**

TABLES: [stop]

PARAMETERS: NONE

RETURNS:  
stop\_id,  
street\_address,  
zip\_code,  
latitude,  
longitude,  
is\_active**sp\_get\_stop\_by\_id**

TABLES: [stop]

PARAMETERS: @stop\_id int

RETURNS:  
stop\_id,  
street\_address,  
zip\_code,  
latitude,  
longitude,  
is\_active**sp\_insert\_stop**

TABLES: [stop]

PARAMETERS: (1) @street\_address

(2) @zip\_code  
(3) @latitude  
(4) @longitude  
(5) @is\_active

RETURNS: SCOPE\_IDENTITY() [stop\_id]

**sp\_update\_stop**

TABLES: [stop]

PARAMETERS: @stop\_id int,

@street\_address [nvarchar](250),  
@zip\_code [nvarchar](20),  
@latitude decimal(8,6),

```
@longitude           decimal(9,6),
@is_active          [bit] = 1,
@old_street_address [nvarchar](250),
@old_zip_code       [nvarchar](20),
@old_latitude       decimal(8,6),
@old_longitude      decimal(9,6),
@old_is_active      [bit] = 1
```

RETURNS:

```
@@ROWCOUNT [rows_affected]
```

## UML – Data Domain Models

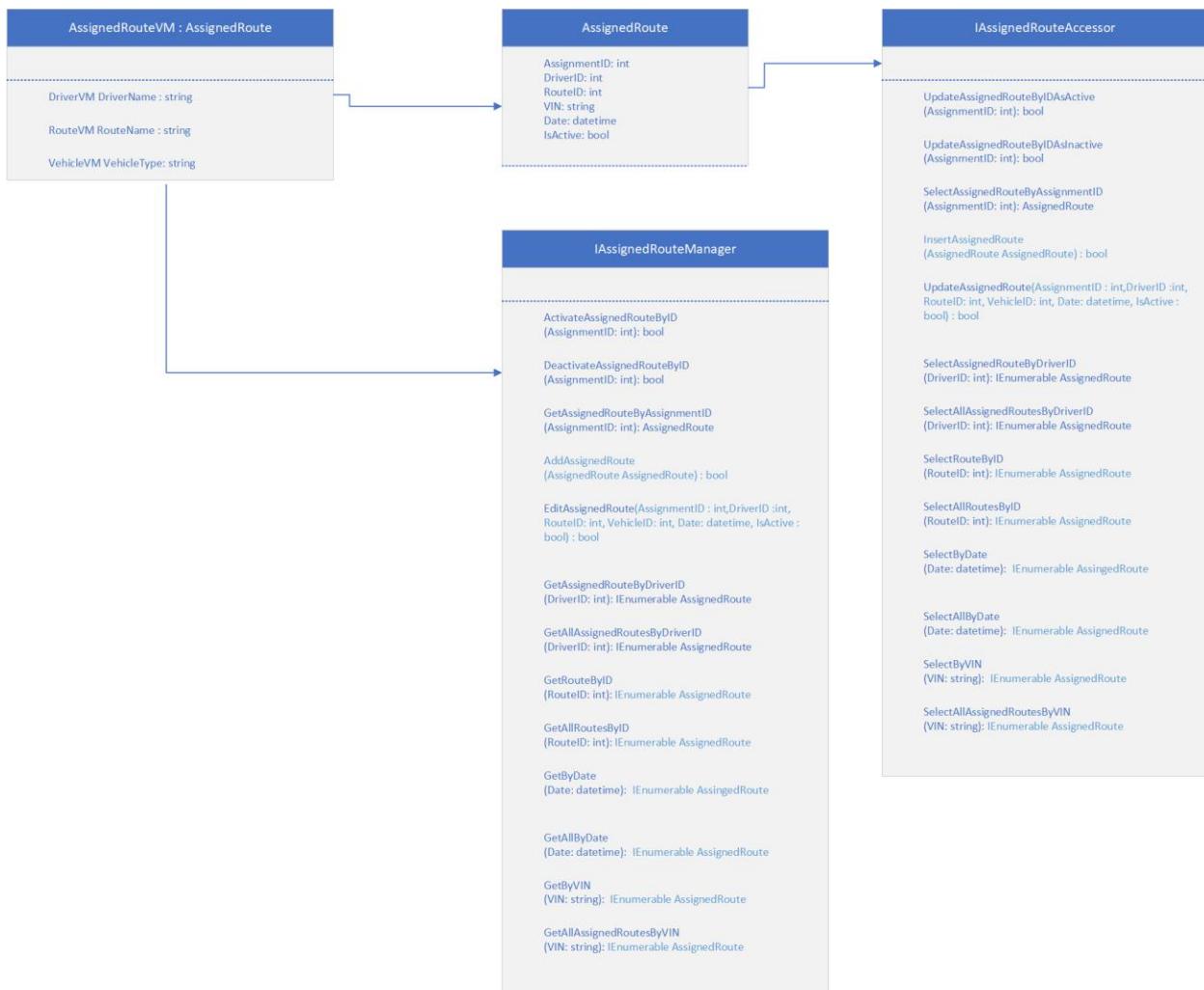
The Data Domain Models represent the in-memory representations of data objects which the application will use. The Storage Models contain properties matching exactly the fields of a specific table in the database. The View Models contain all the same properties of the associated Storage Models as well as additional data which will only be relevant from the perspective of the application logic as well as data which does not exist in the database but will be displayed to the user.

# Class Interface Diagrams

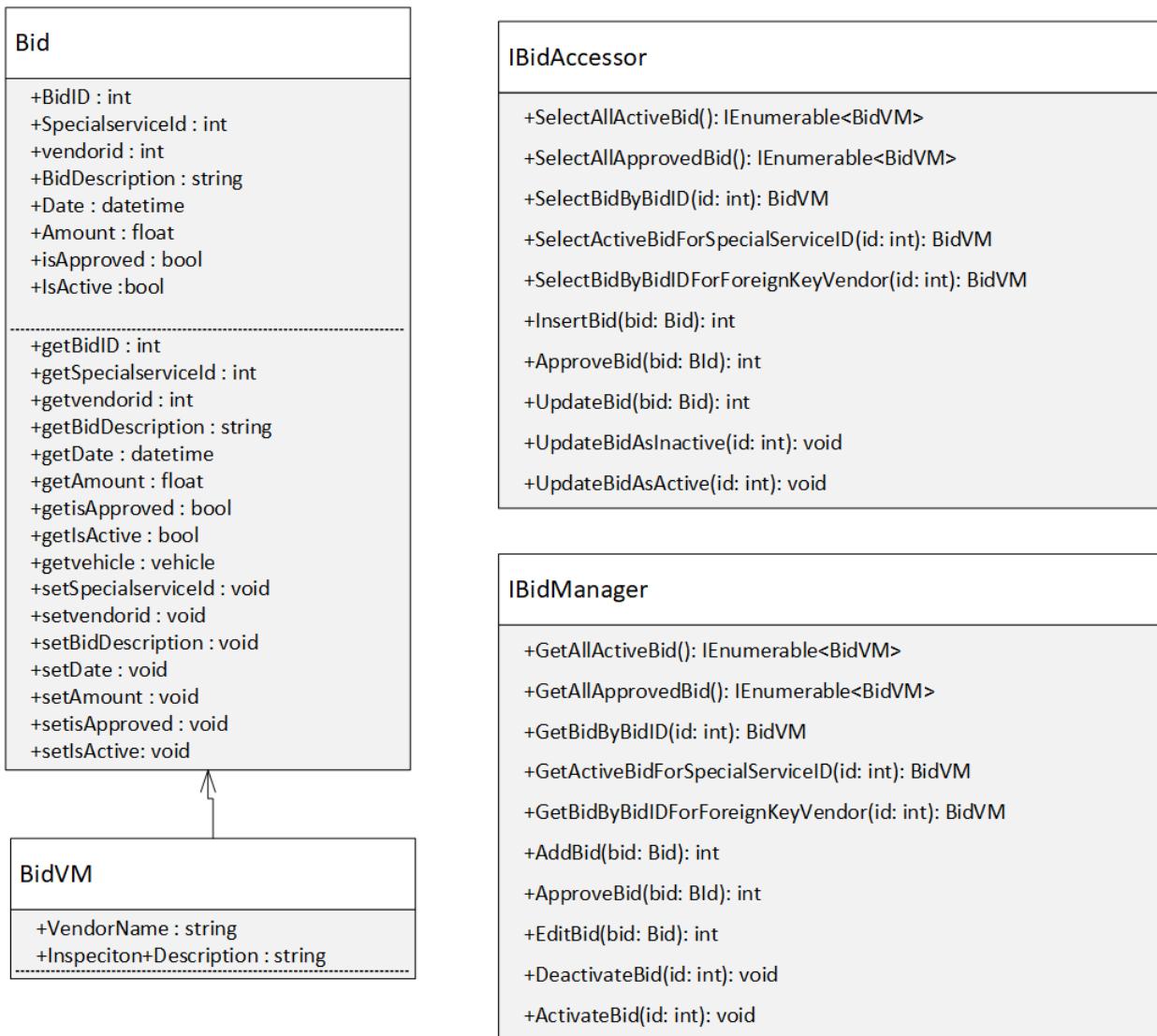
## Accommodation



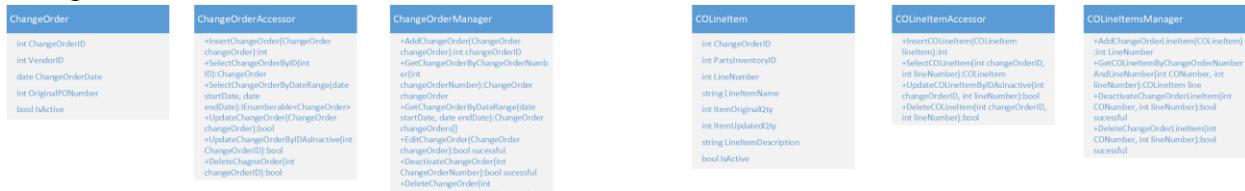
## Assigned Route



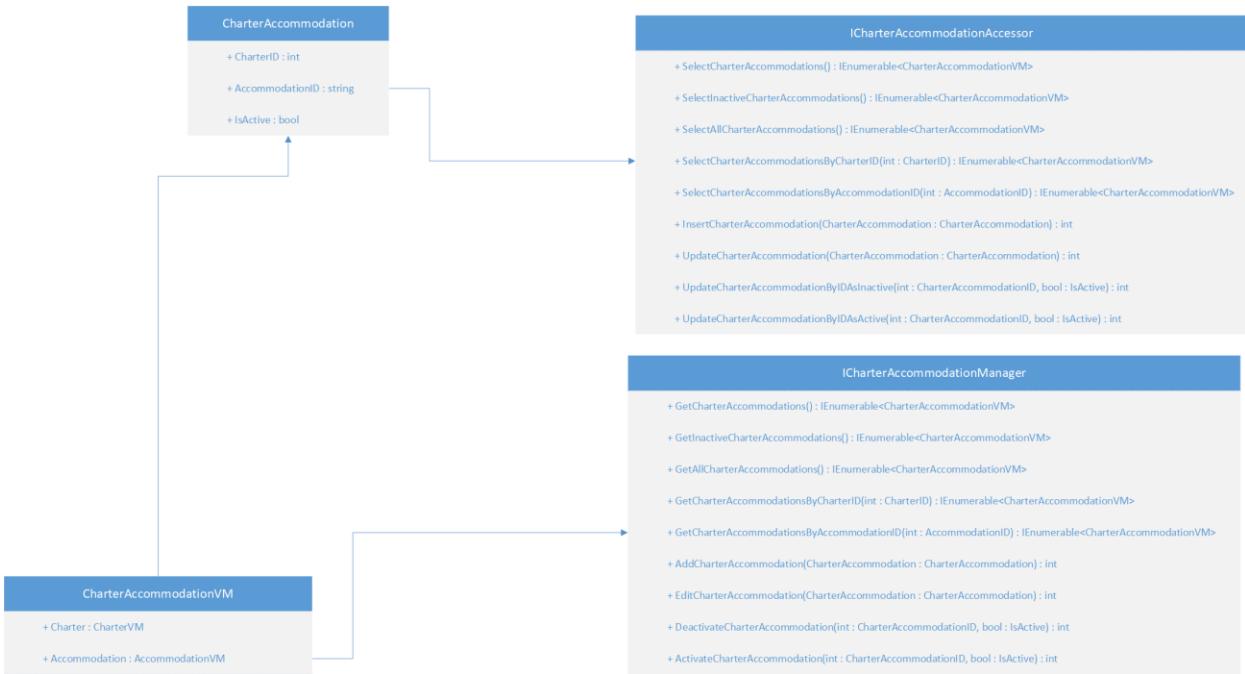
## Bid



## Change Order



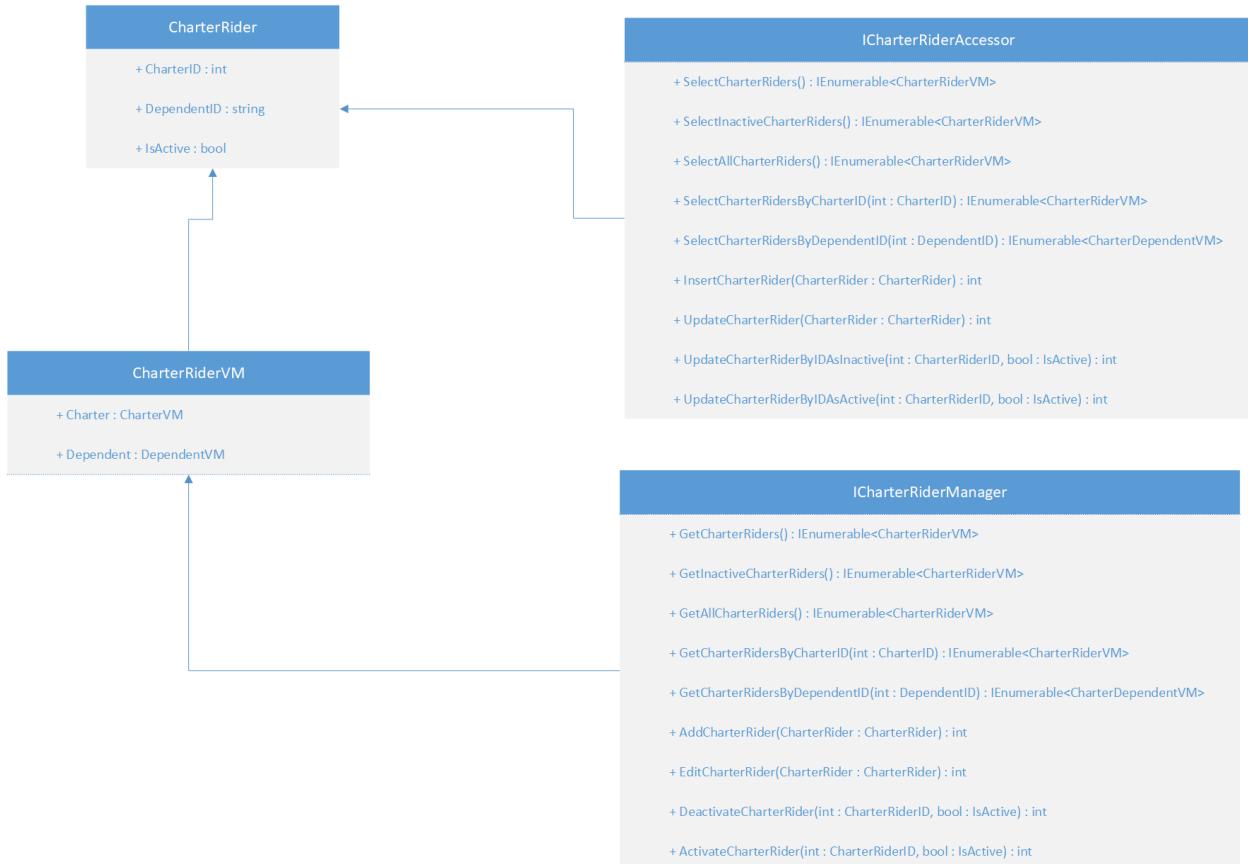
## Charter Accommodation



## Charter



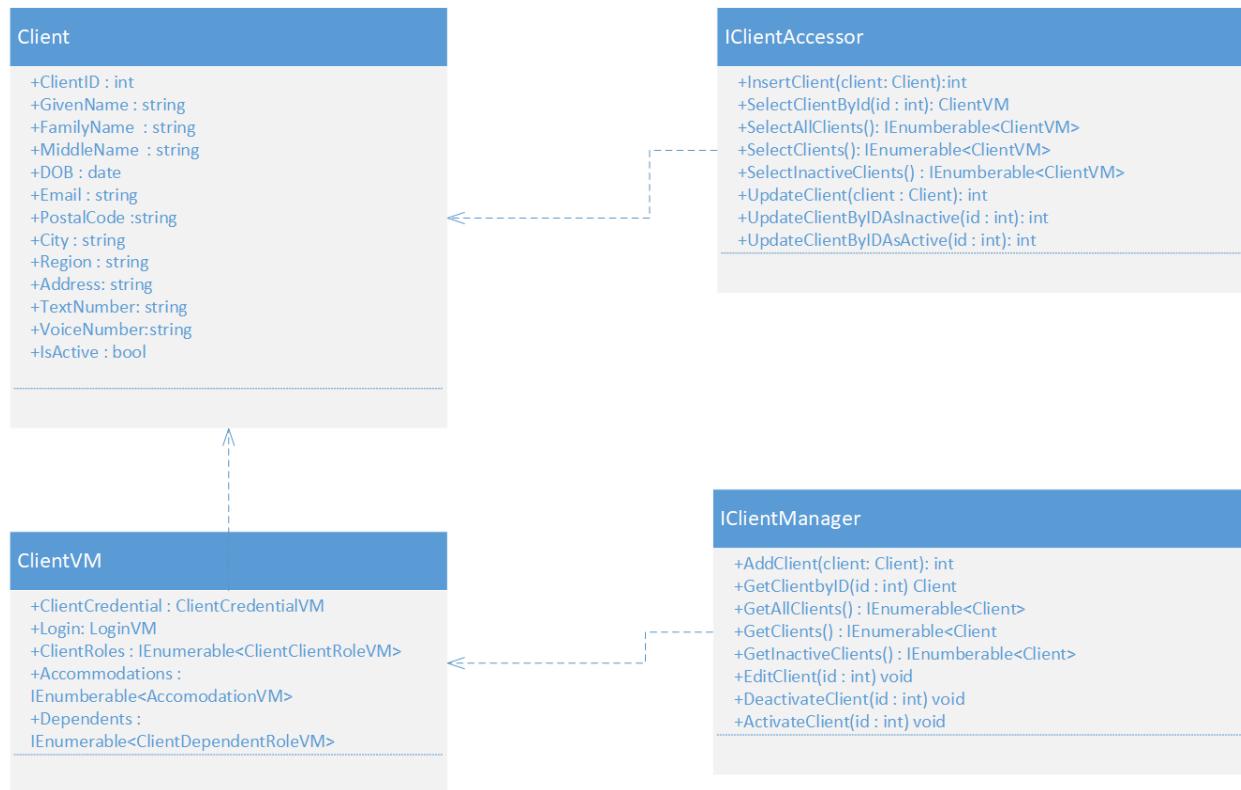
## Charter Rider



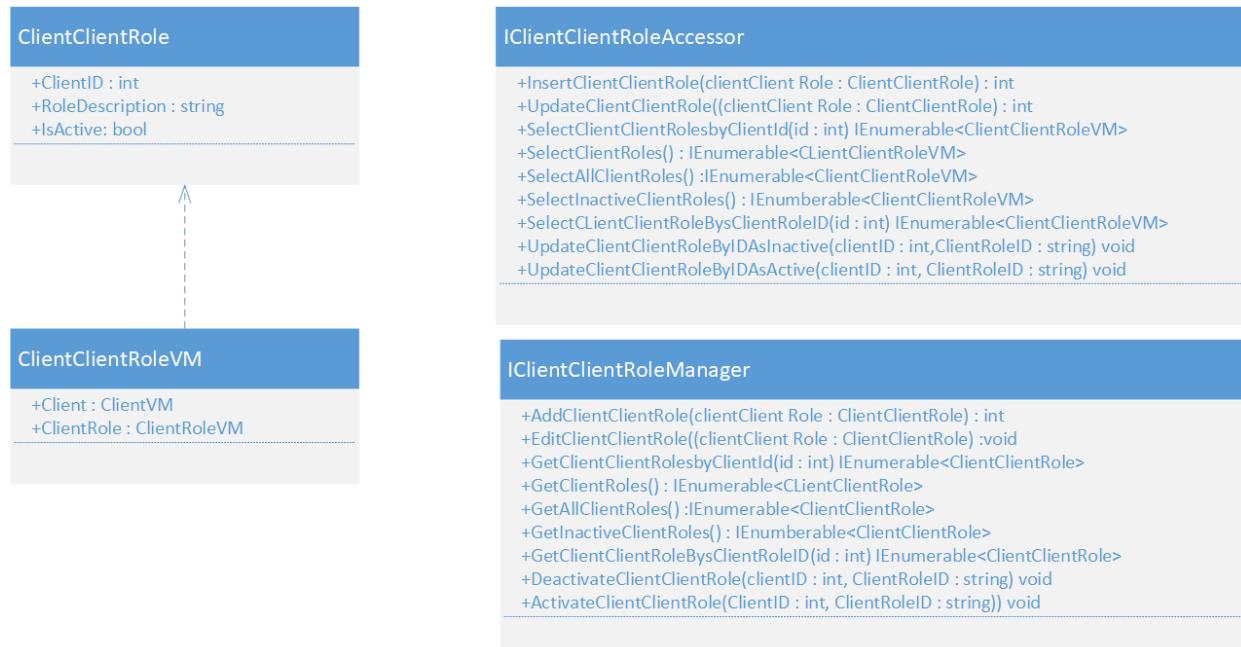
## Charter Stop



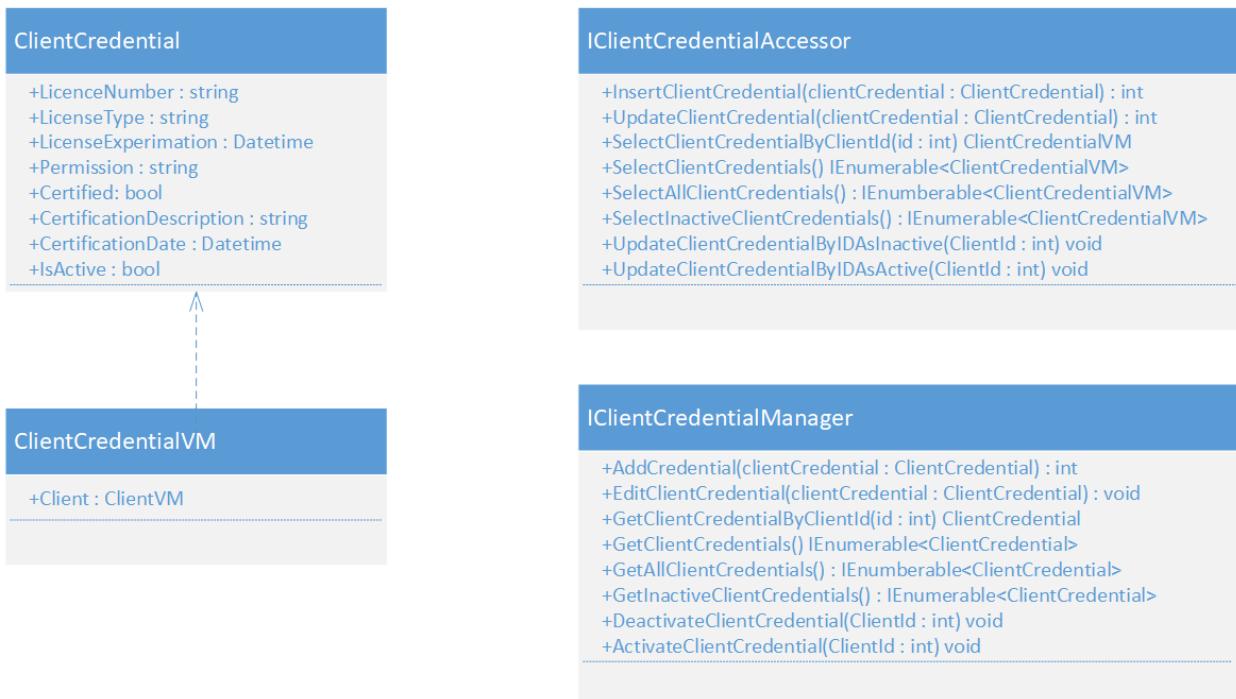
## Client



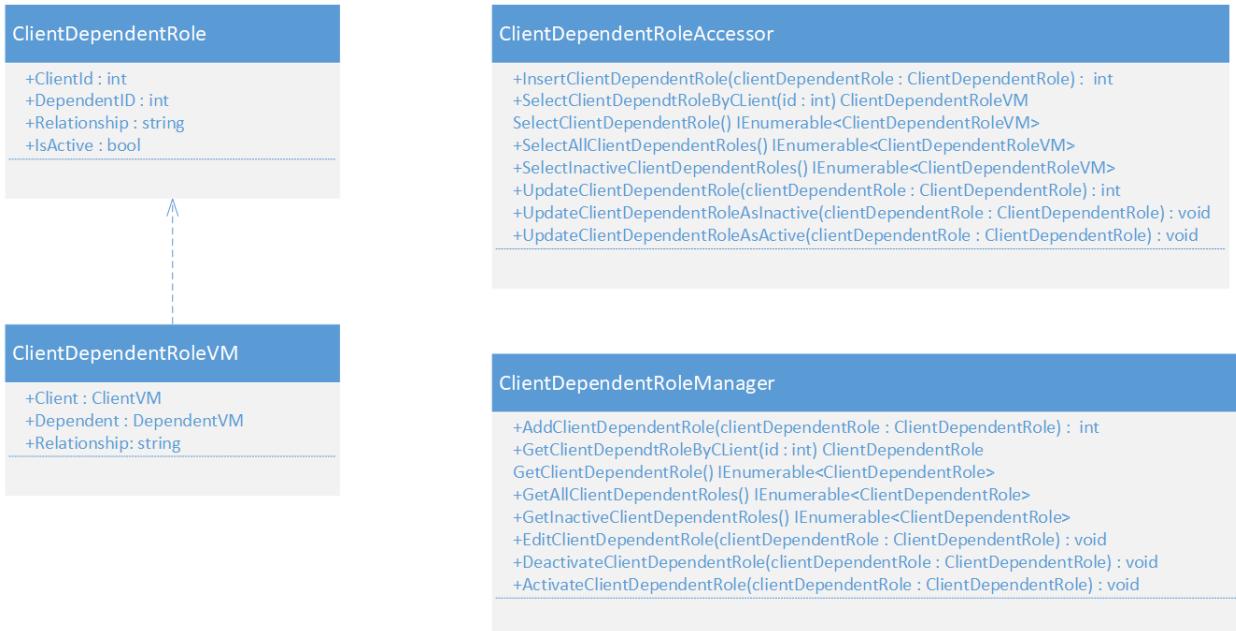
## Client Client-Role



## Client Credential



## Client Dependent-Role



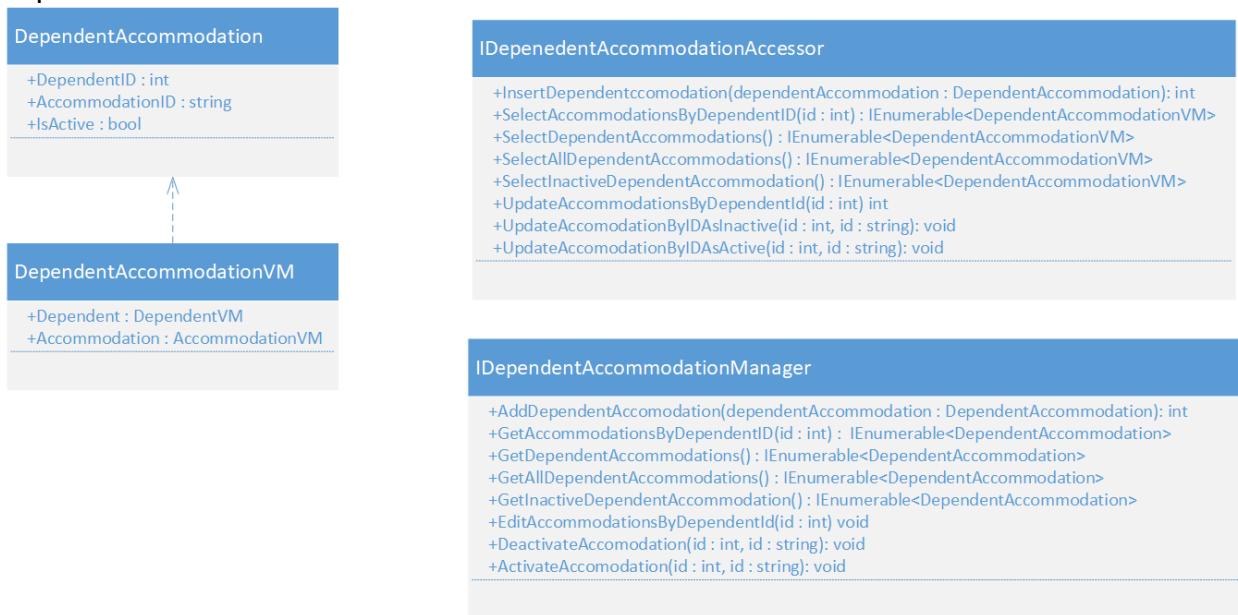
## Client Role



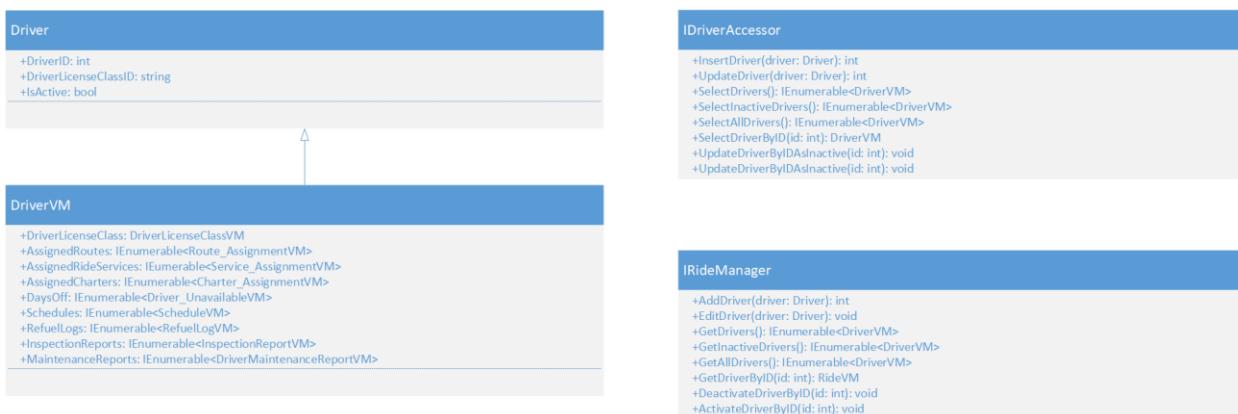
## Dependent



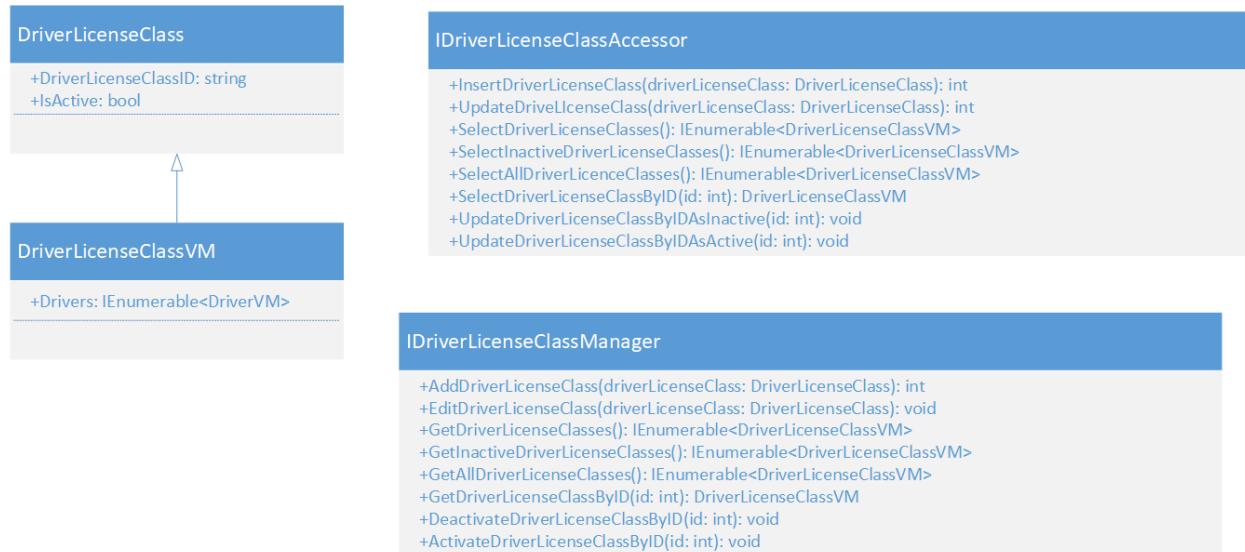
## Dependent Accommodation



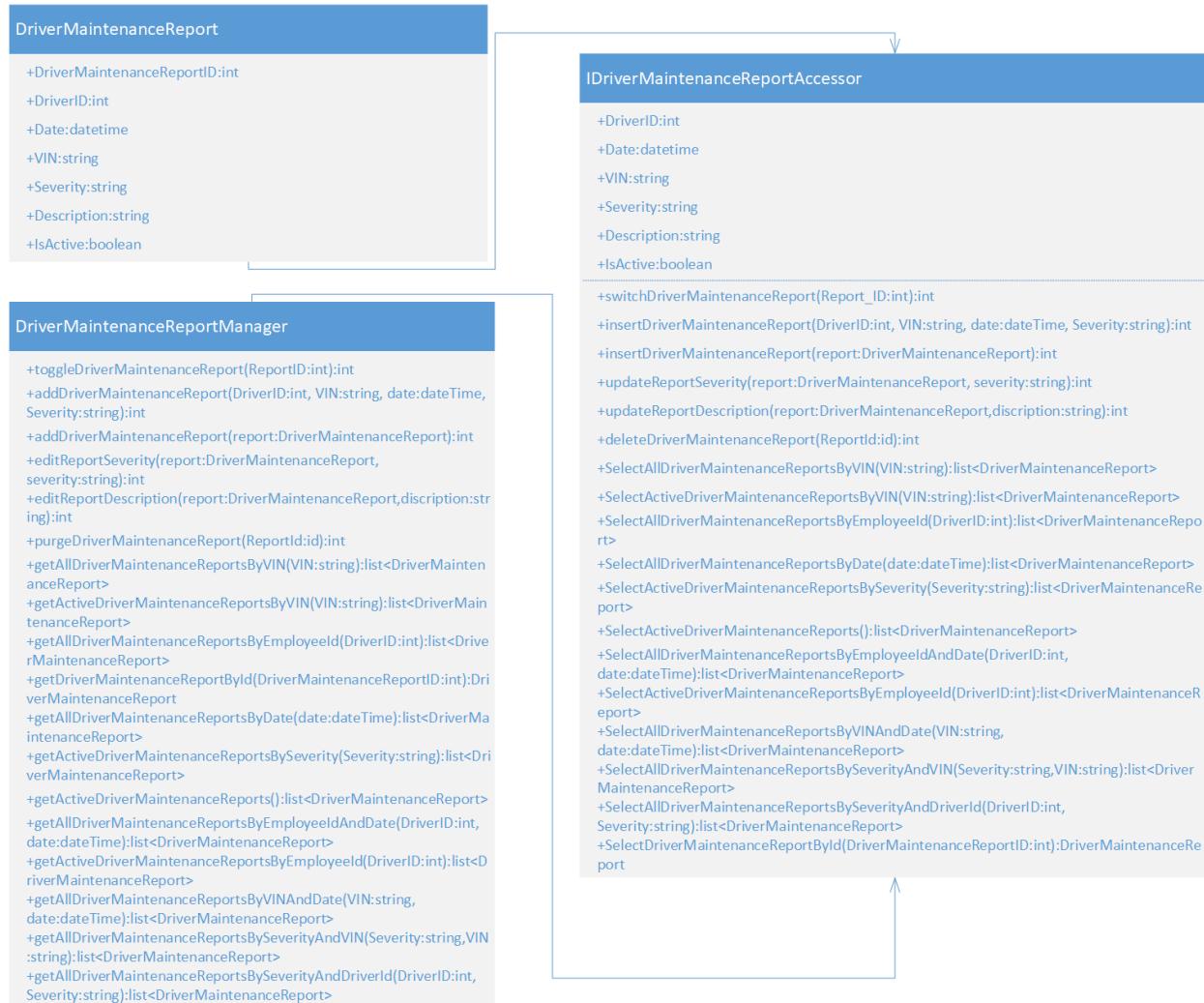
## Driver



## Driver License



## Driver Maintenance Report



## Employee

### Employee

```
+ int EmployeeID  
+ string GivenName  
+ string FamilyName  
+ string Address  
+ string Address2  
+ string City  
+ string State  
+ string Country  
+ string Zip  
+ string PhoneNumber  
+ string Email  
+ string Position  
+ bool IsActive
```

### EmployeeVM

```
+ IEnumerable<EmployeeRoleVM>  
EmployeeRole  
  
+ IEnumerable<DriverVM> Driver  
  
+ IEnumerable<SafetyReportVM>  
SafetyReport  
  
+ IEnumerable<ServiceDetail> Creator  
  
+ IEnumerable<ServiceDetail> Servicer  
  
+ IEnumerable<SpecialInspection>  
SpecialInspection  
  
+ IEnumerable<PartsRequest>  
PartsRequest  
  
+ IEnumerable<SupportTicketEmployeeLine> SupportTicketEmployeeLine  
  
+ EmployeeRoleVM  
CreateEmployeeRole(RoleVM role) --  
  
+ SupportTicketEmployeeLine  
CreateSupportTicketEmployeeLine  
(SupportTicketVM supportTicket)
```

<<Interface>>

### IEmployeeAccessor

```
+ SelectEmployeeByID(id: int):  
EmployeeVM  
  
+ SelectActiveEmployees():  
IEnumerable<EmployeeVM>  
  
+ SelectEmployees():  
IEnumerable<EmployeeVM>  
  
+ InsertEmployee(employee: Employee):  
int  
  
+ UpdateEmployeeByID(employee: Employee): int  
  
+ DeleteEmployeeByID(id: int): int  
  
+ UpdateEmployeeByIDAsInactive(id:  
int): int  
  
+ UpdateEmployeeByIDAsActive(id: int):  
int
```

<<Interface>>

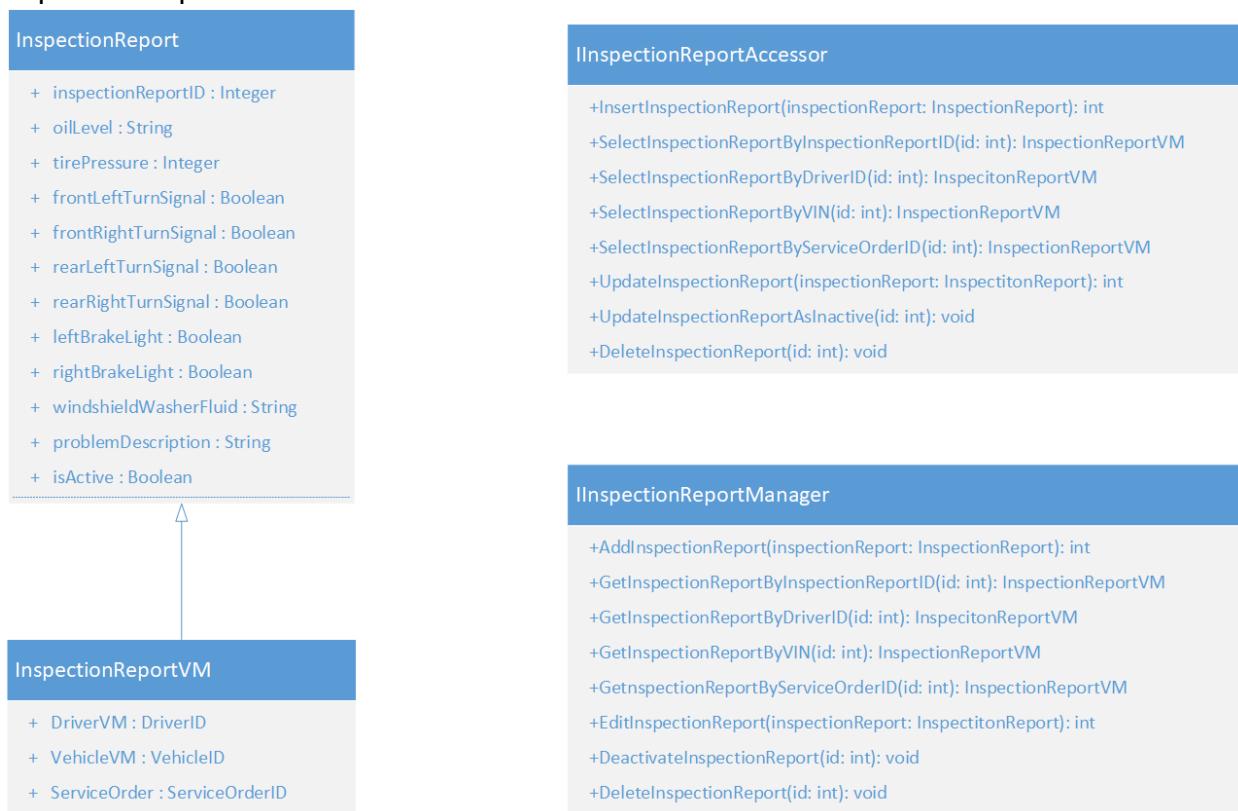
### IEmployeeManager

```
+ GetEmployeeByID(id: int):  
EmployeeVM  
  
+ GetActiveEmployees():  
IEnumerable<EmployeeVM>  
  
+ GetEmployees():  
IEnumerable<EmployeeVM>  
  
+ AddEmployee(employee: Employee):  
int  
  
+ EditEmployee(empooyee: Employee):  
int  
  
+ DeleteEmployeeByID(id: int): int  
  
+ DeactivateEmployeeByID(id: int): int  
  
+ ActivateEmployeeByID(id: int): int
```

## Employee Role



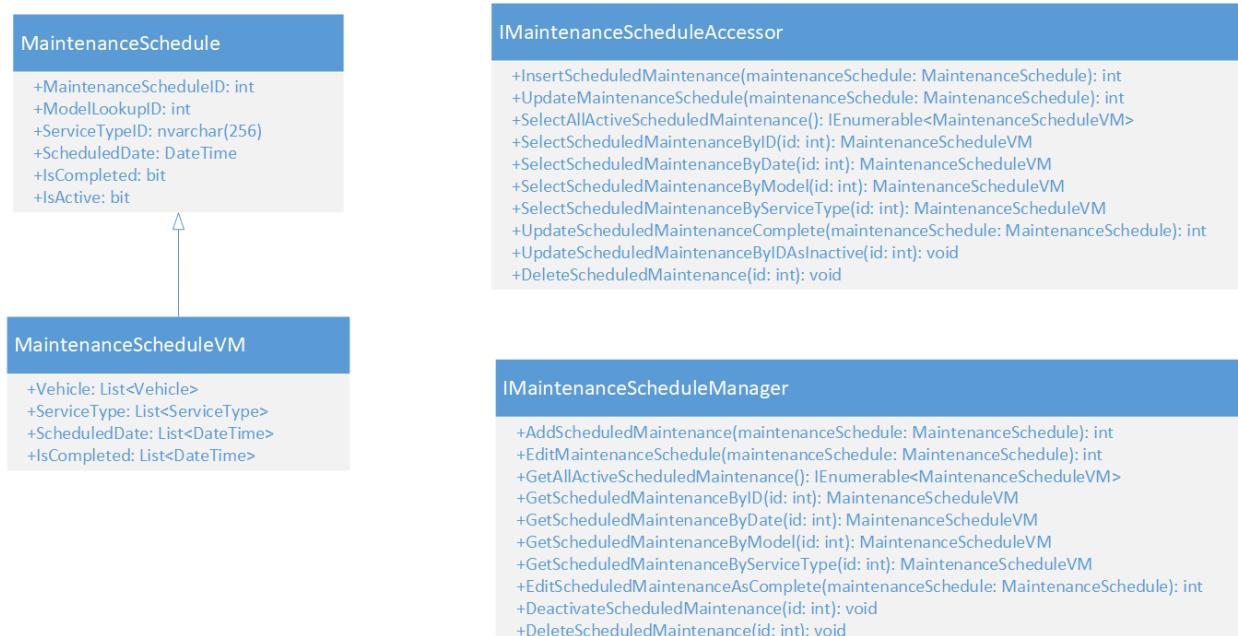
## Inspection Report



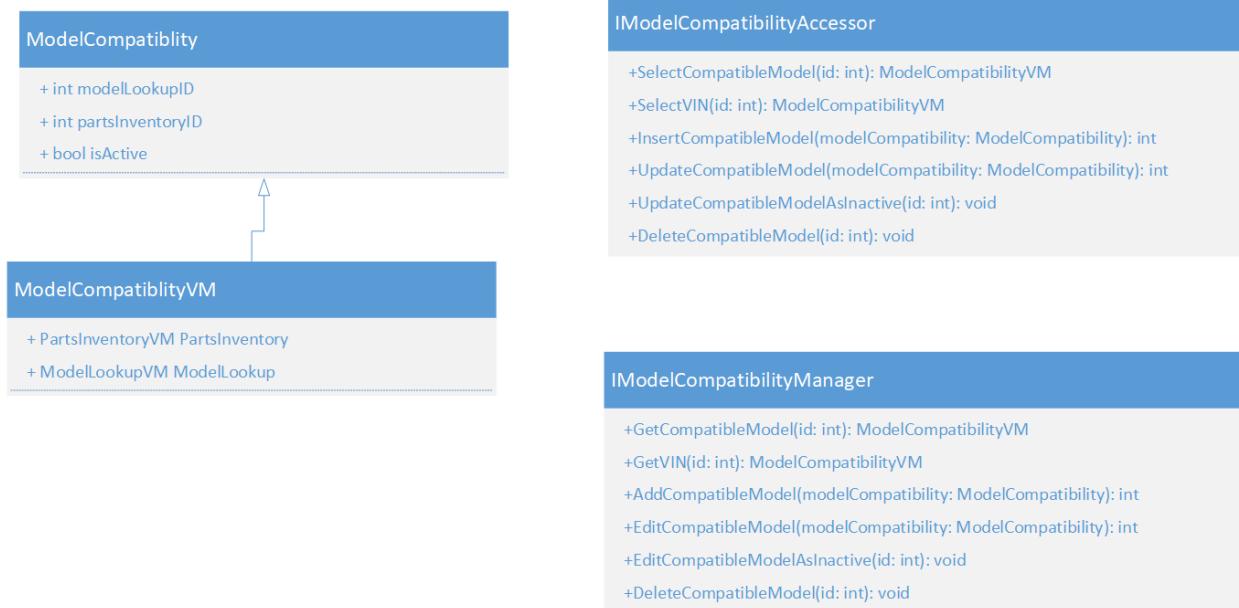
## Login



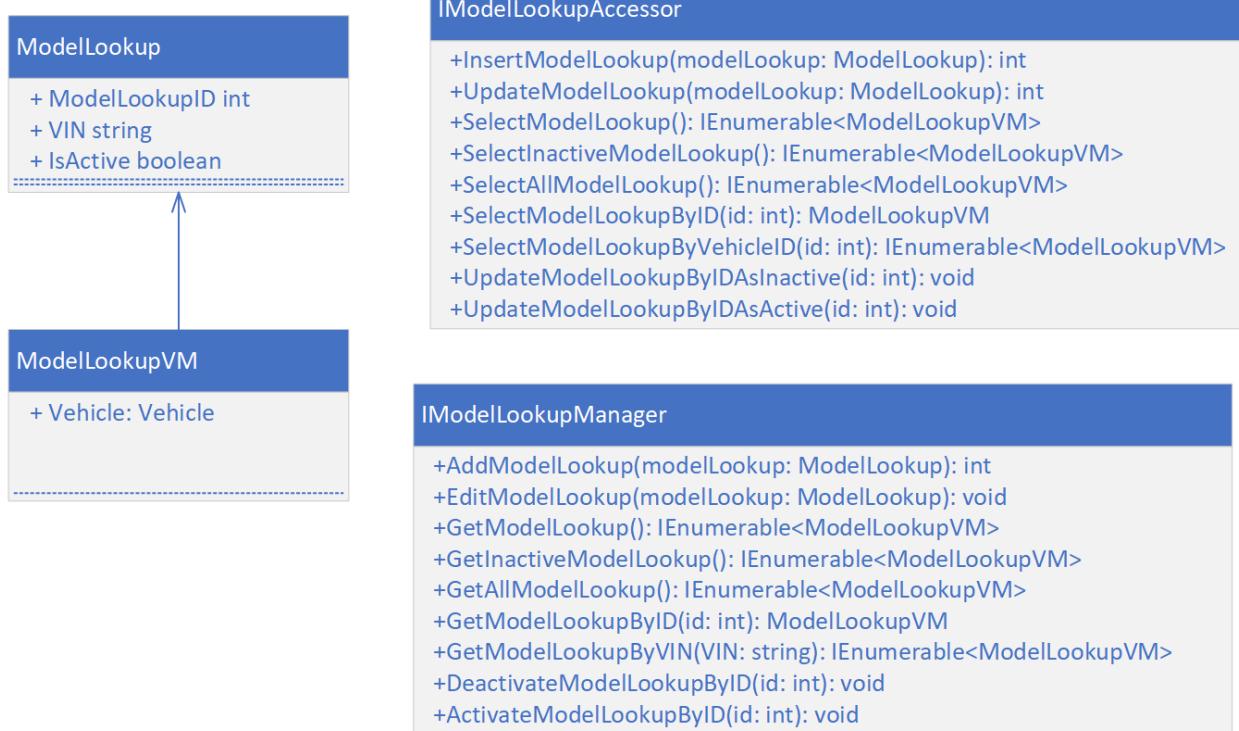
## Maintenance Schedule



## Model Compatibility



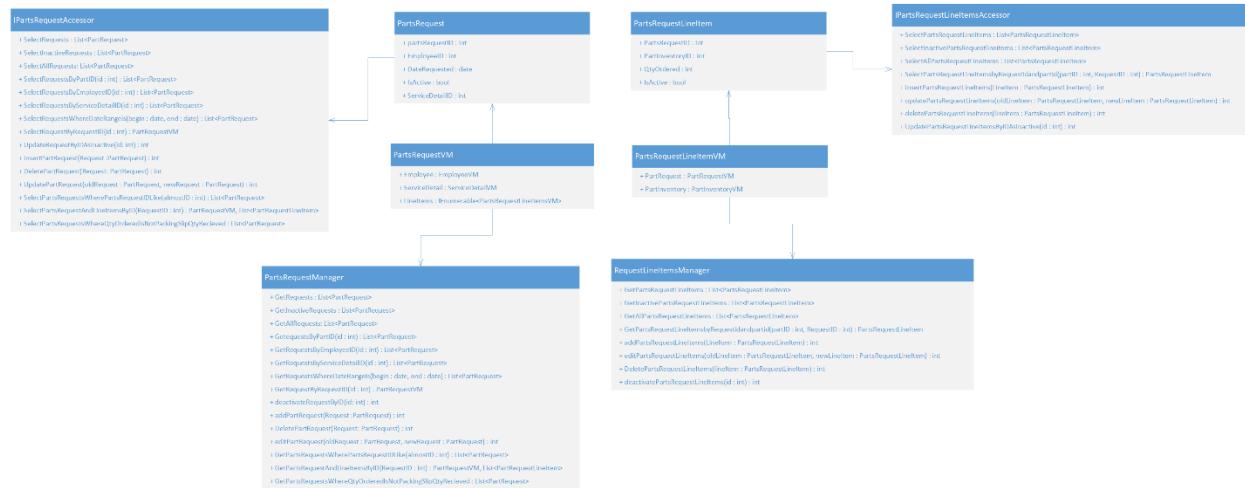
## Model Lookup



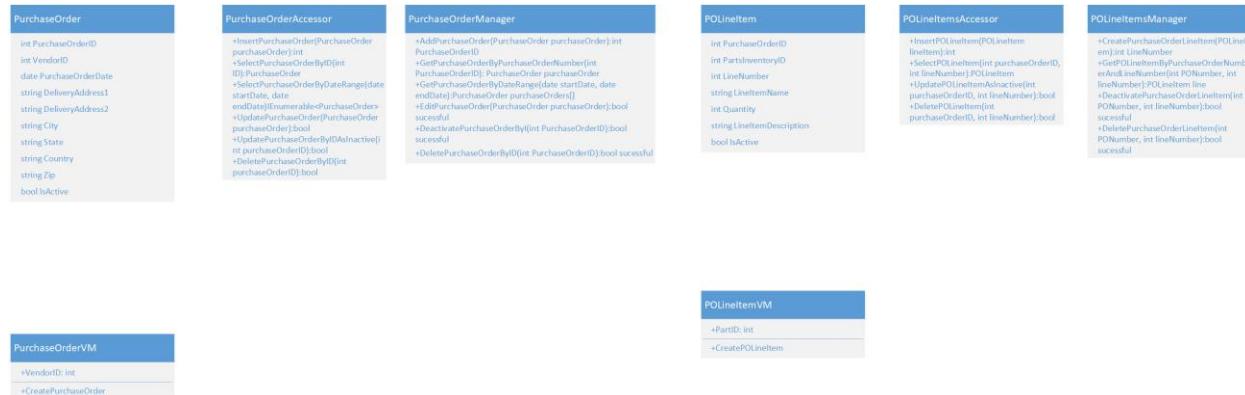
## Packing Slip



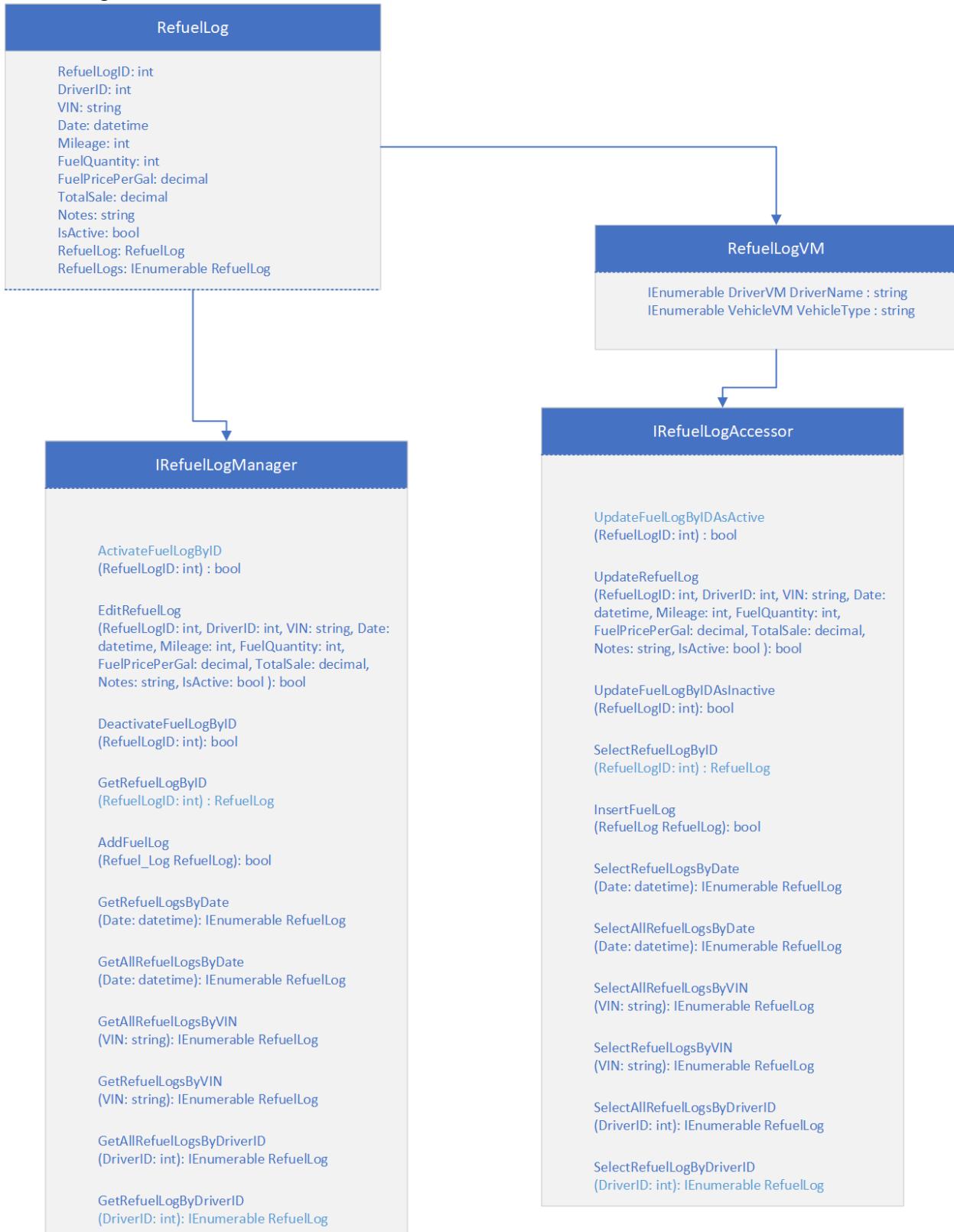
## Parts Request



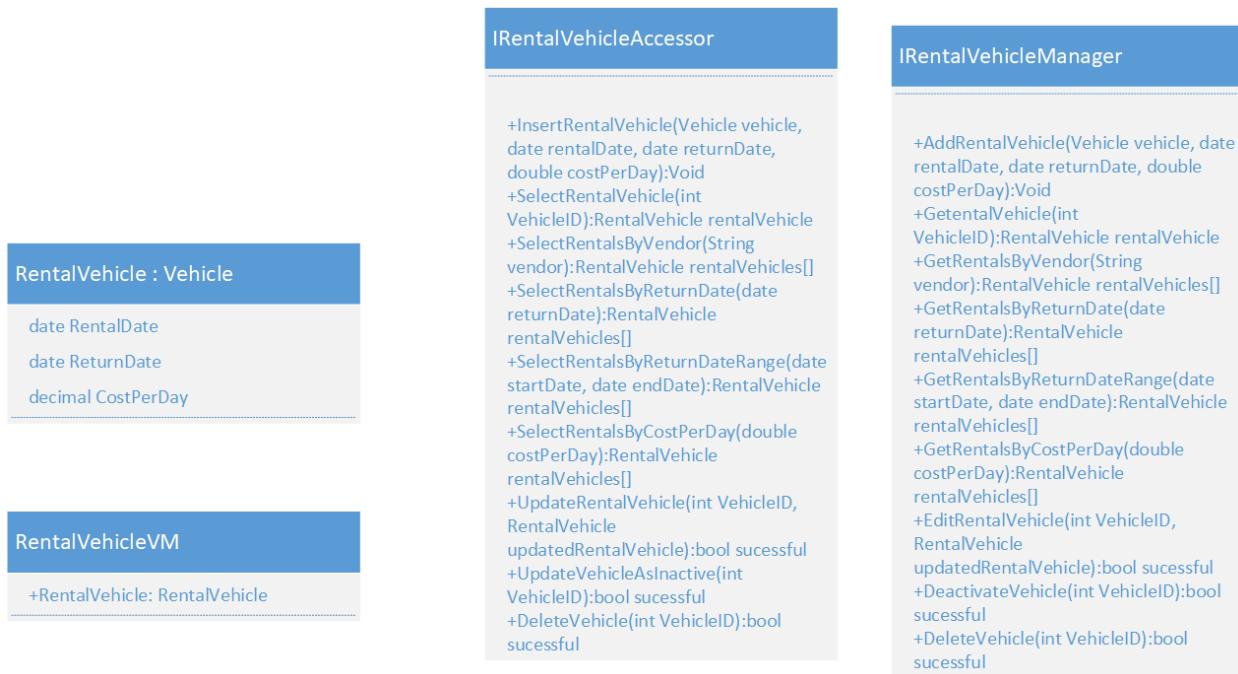
## Purchase Order



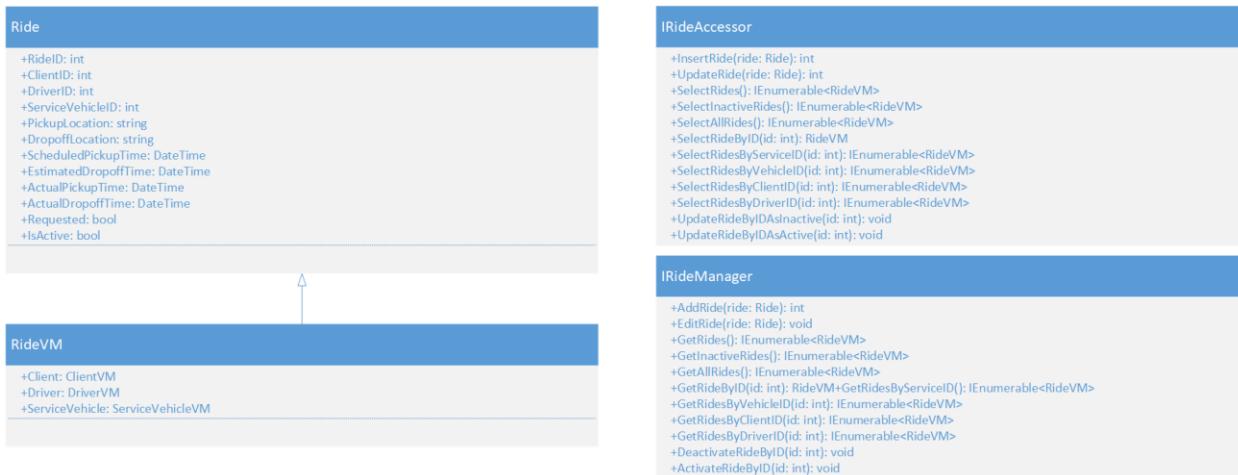
## Refuel Log



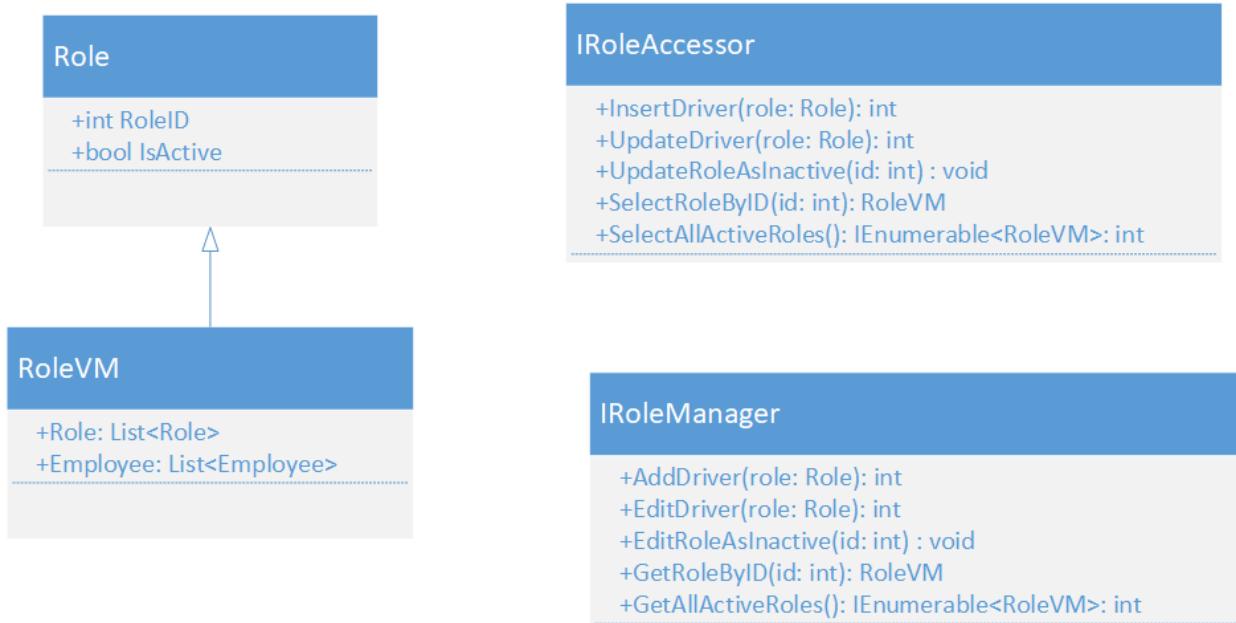
## Rental



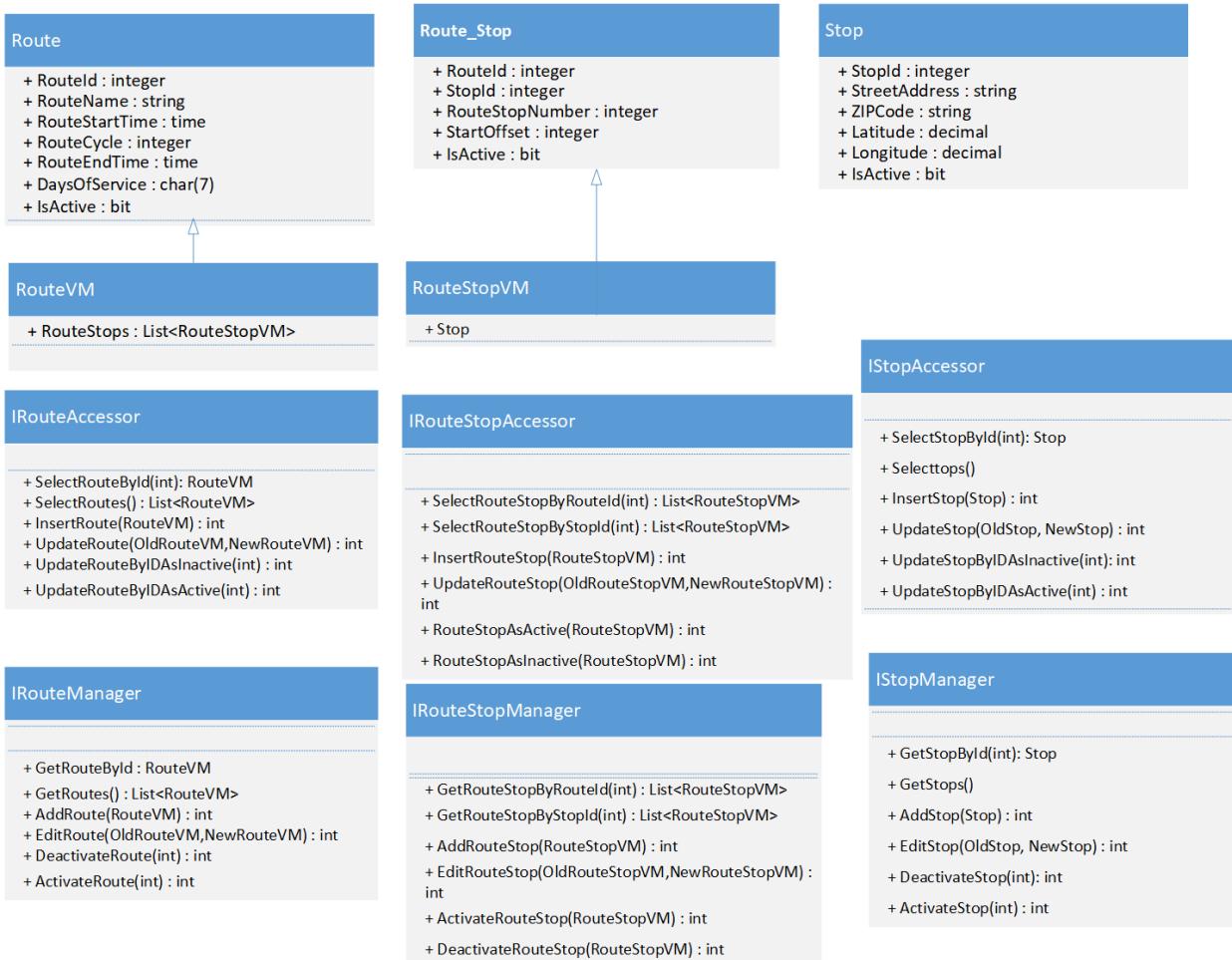
## Ride



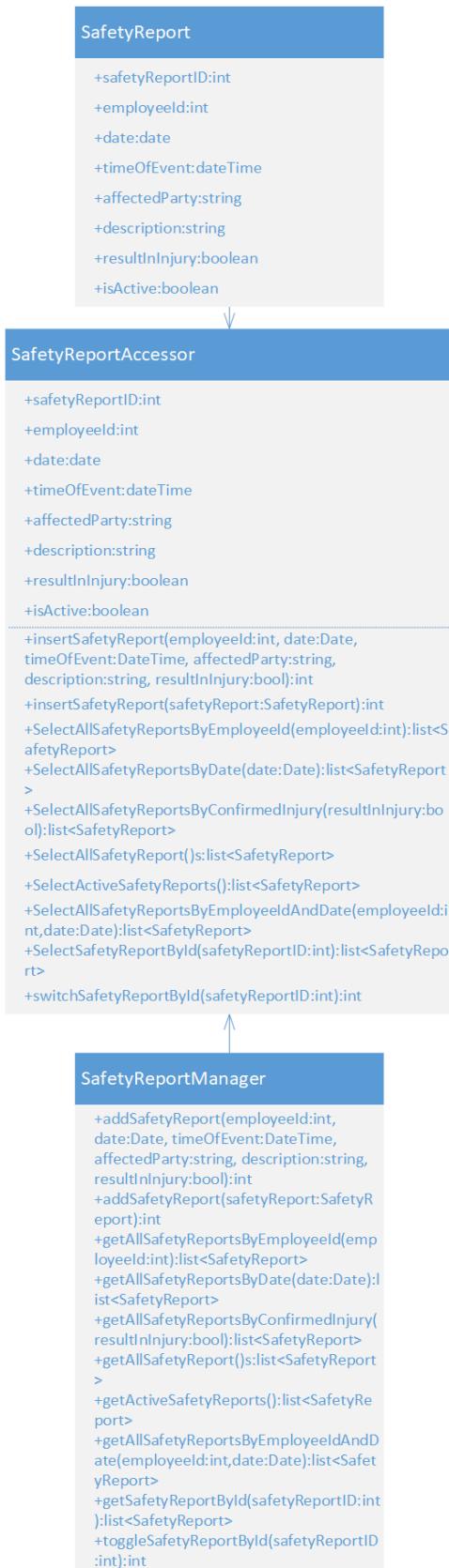
## Role



## Route Stop Route-Stop

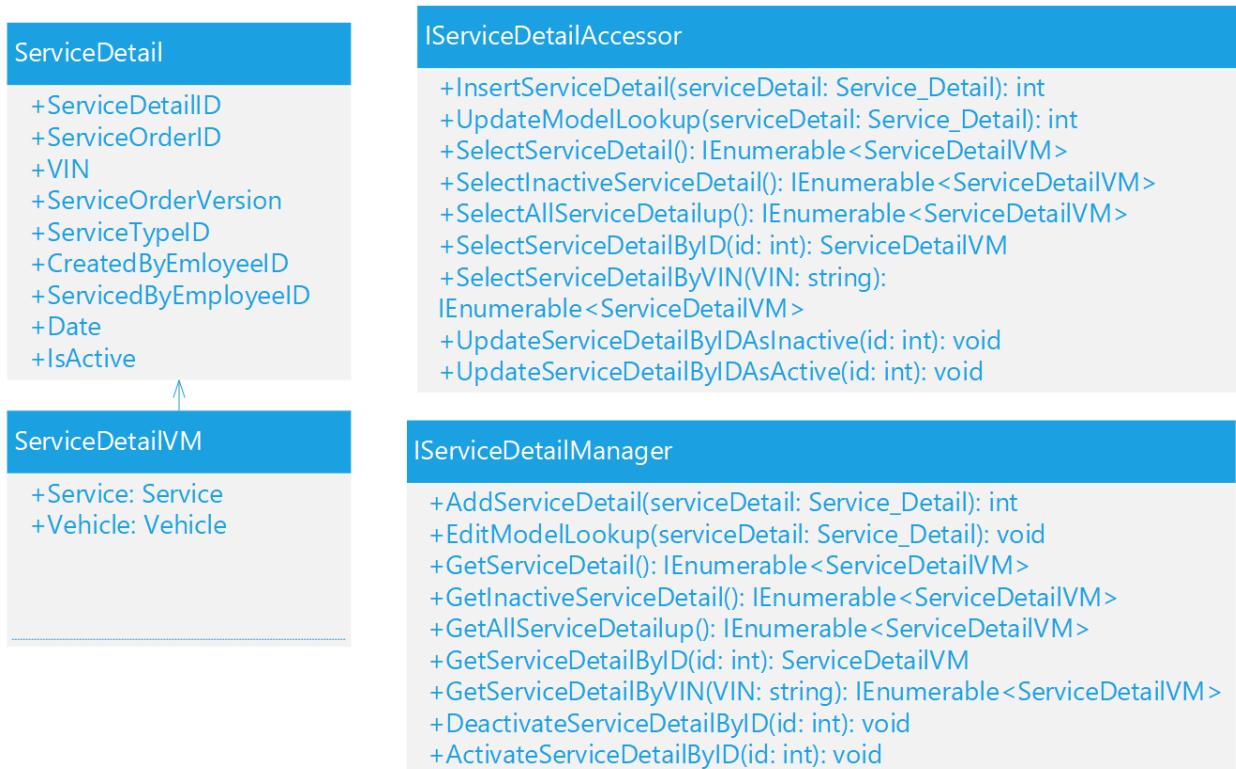


## Safety Report Schedule





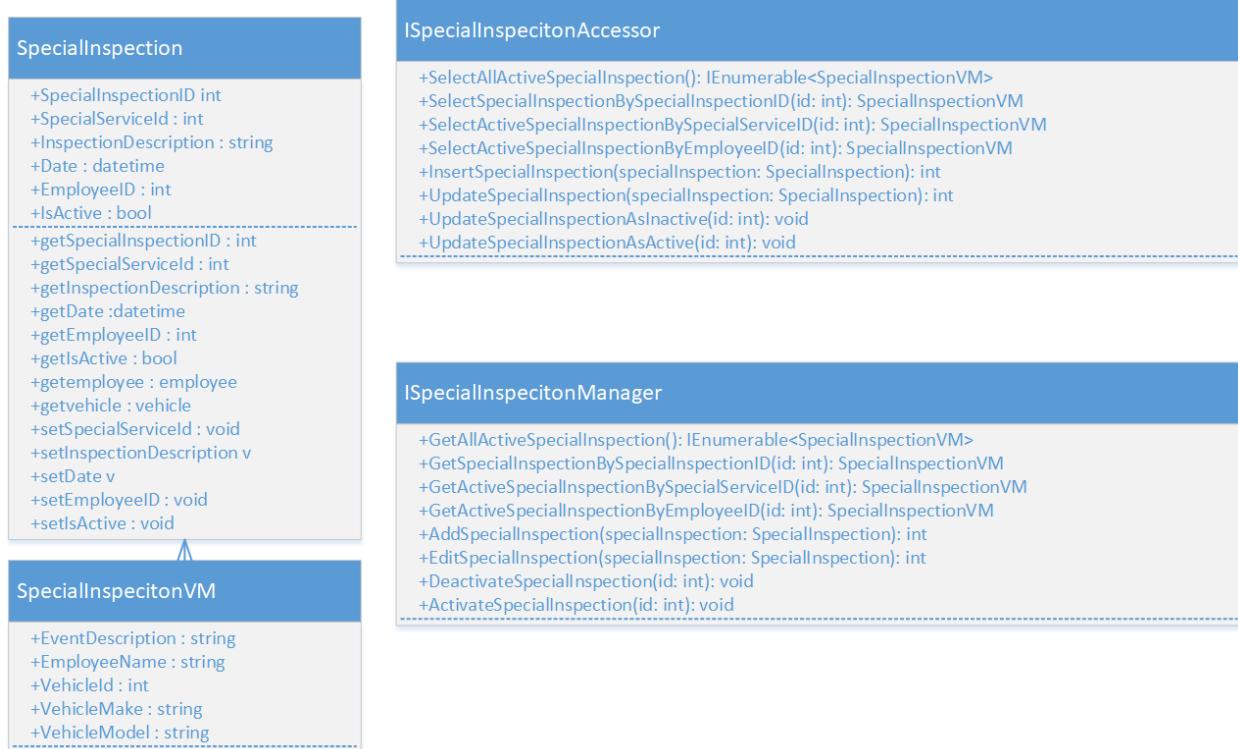
## Service Detail



## Service Vehicle



## Special Inspection



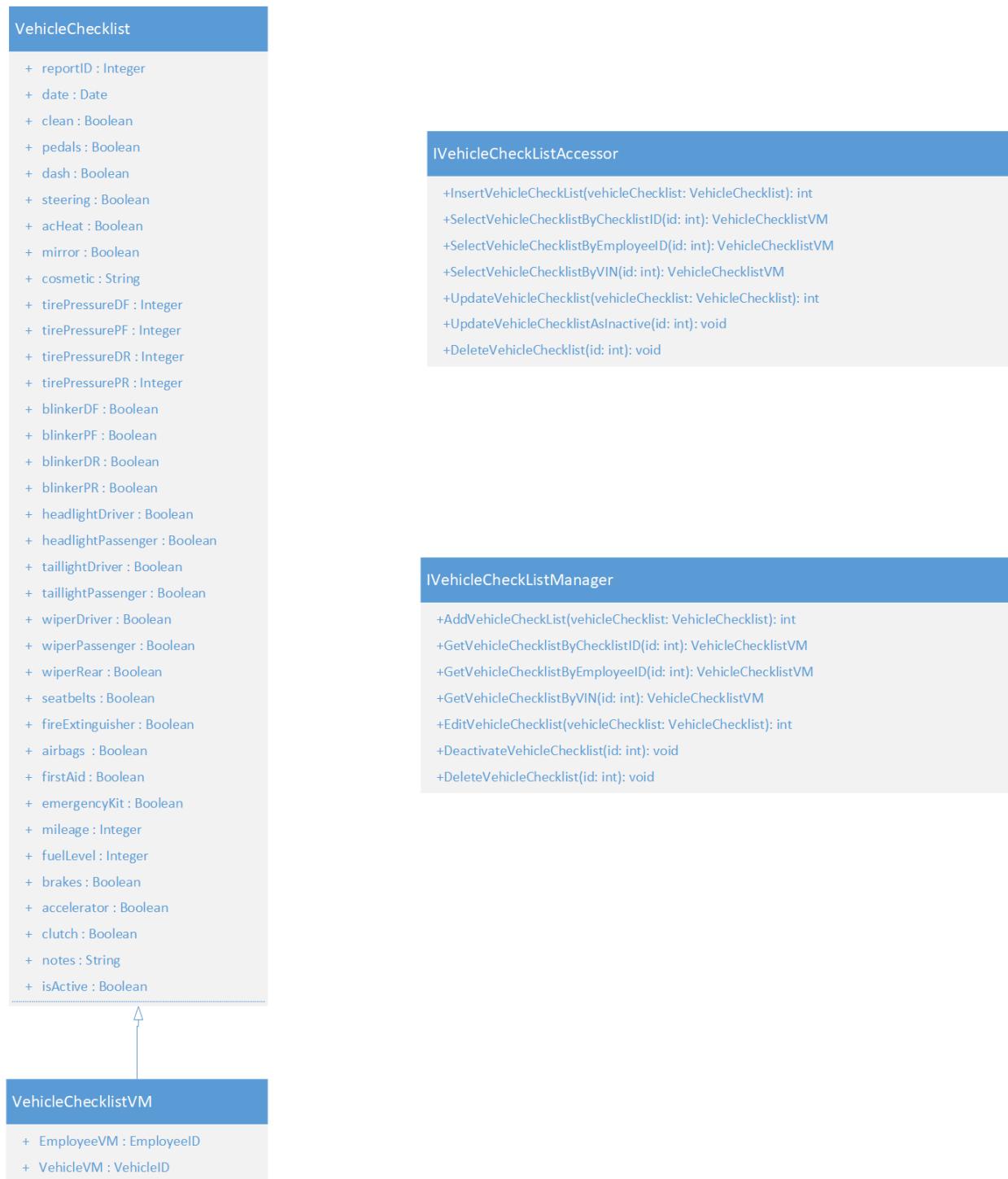
## Special Service Detail



## Special Work Order



## Vehicle Checklist



## Vehicle

### Vehicle

```
+VIN: string
+ModelLookupID: int
+Mileage: int
+LicensePlate: string
+VehicleType: string
+DateEntered: date
+Rental: bool
+IsActive: bool
```



### VehicleVM

```
+AssignedWorkOrders:
IEnumerable<Work_Order_VM>
+AssignedRouteAssignments:
IEnumerable<Route_Assignment_VM>
+AssignedRideServices:
IEnumerable<Service_Assignment_VM>
+OtherUnavailable:
IEnumerable<Vehicle_Unavailable_VM>
+AssignedCharters:
IEnumerable<Charter_Assignment_VM>
+Model: ModelVM
```

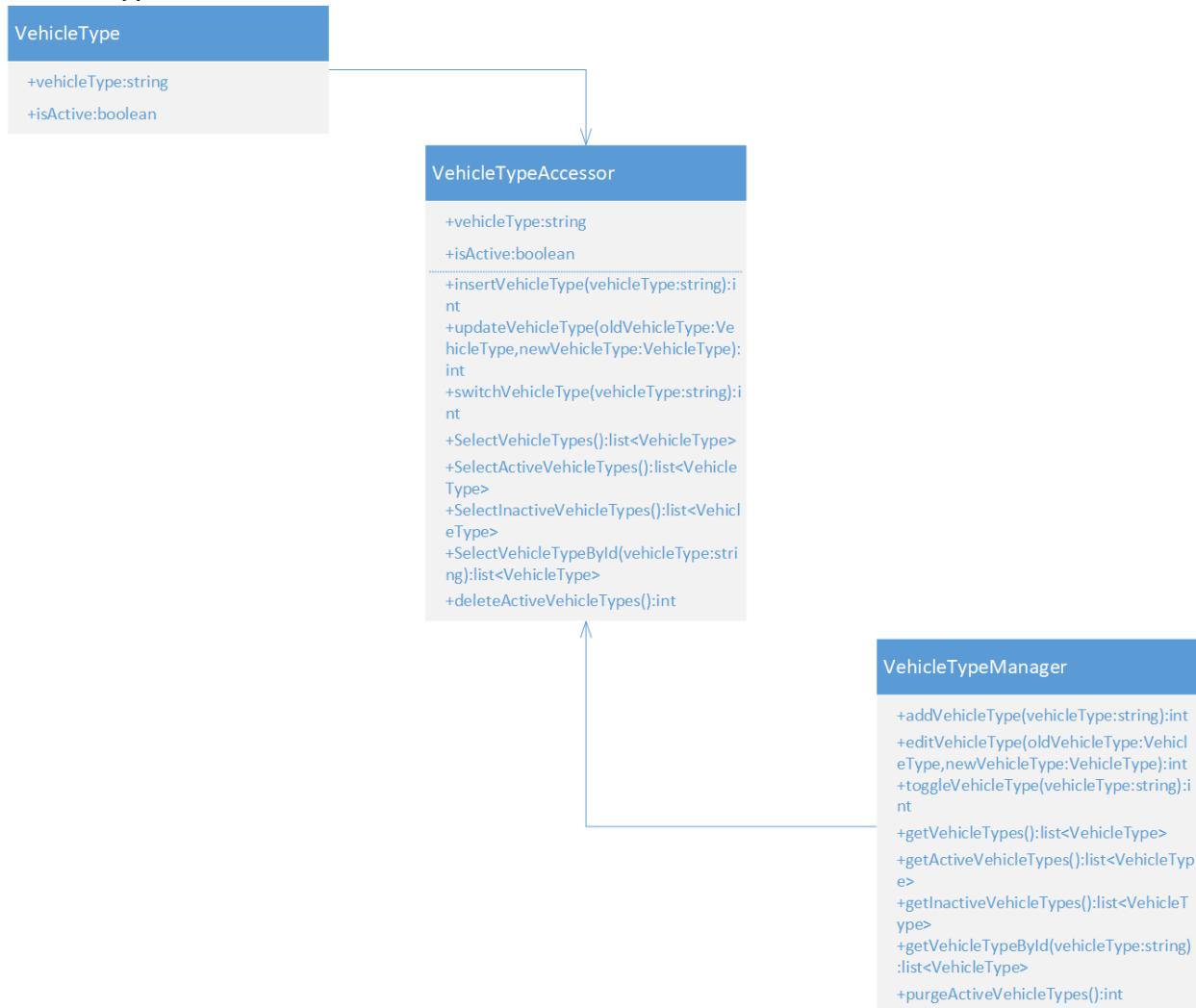
### IVehicleAccessor

```
+InsertVehicle(Vehicle vehicle):int VehicleID
+SelectVehiclesByMakeModel(String make, String model):Vehicle vehicle[]
+SelectRentalVehicles():Vehicle vehicles[]
+SelectActiveVehicles():Vehicle vehicles[]
+SelectInactiveVehicles():Vehicle vehicles[]
+UpdateVehicle(Vehicle vehicleOld, Vehicle vehicleNew):bool sucessful
+SelectAllVehicles():Vehicle vehicles[]
+UpdateVehicleAsInactive(VehicleID):bool sucessful
+DeleteVehicle(VehicleID):bool sucessful
+SelectVehiclesByType(String type):Vehicle vehicles[]
+SelectVehiclesByMileageRange(int mileageLow, int mileageHigh):Vehicle vehicle[]
+SelectVehicleByVehicleNumber(int VehicleID):int VehicleID
```

### IVehicleManager

```
+AddVehicle(Vehicle vehicle):int VehicleID
+GetVehiclesByMakeModel(String make, String model):Vehicle vehicle[]
+GetRentalVehicles():Vehicle vehicles[]
+GetActiveVehicles():Vehicle vehicles[]
+GetInactiveVehicles():Vehicle vehicles[]
+EditVehicle(Vehicle vehicleOld, Vehicle vehicleNew):bool sucessful
+GetAllVehicles():Vehicle vehicles[]
+DeactivateVehicle(VehicleID):bool sucessful
+DeleteVehicle(VehicleID):bool sucessful
+GetVehiclesByType(String type):Vehicle vehicles[]
+GetVehiclesByMileageRange(int mileageLow, int mileageHigh):Vehicle vehicle[]
+GetVehicleByVehicleNumber(int VehicleID):int VehicleID
```

## Vehicle Type



# UML – Data Access Class Interface Diagrams

The Data Access Class Interface Diagrams describe the public methods representing the application-level version of the Stored Procedures from the database.

# UML - Data Manager Class Interface Diagrams

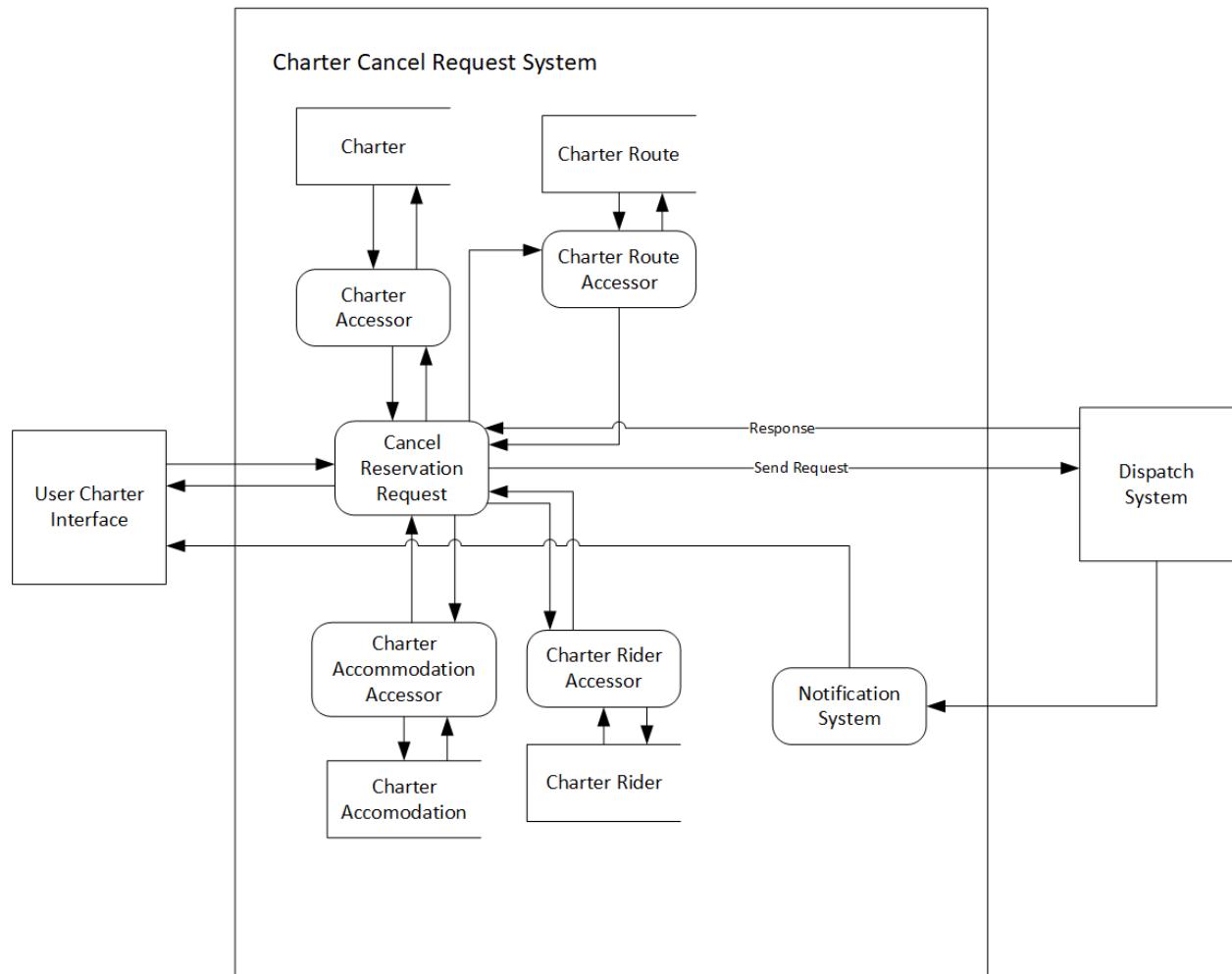
The Data Manager Class Interface Diagrams describe the highest-level logical interfaces from which the application interacts with the system functionality. The Data Manager classes provide the presentation layer of the application, an interface to the data access layer and a domain of automation.

## Data Flow Diagrams by Subsystem

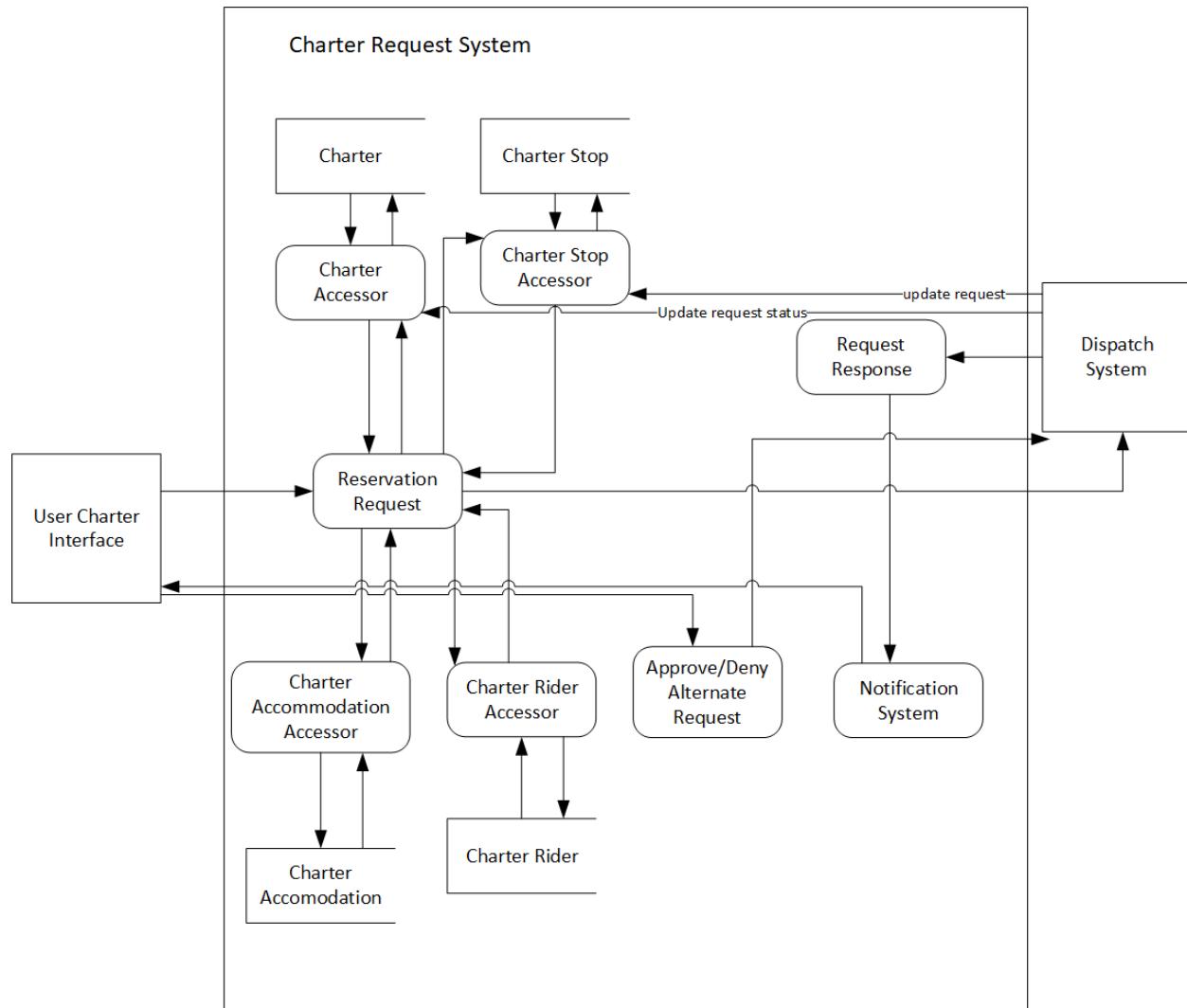
The Data Flow Diagrams describe how data is passed, used, manipulated, accessed, stored, and interpreted between the user, the database, and logical processes within the application.

# Charter Subsystem

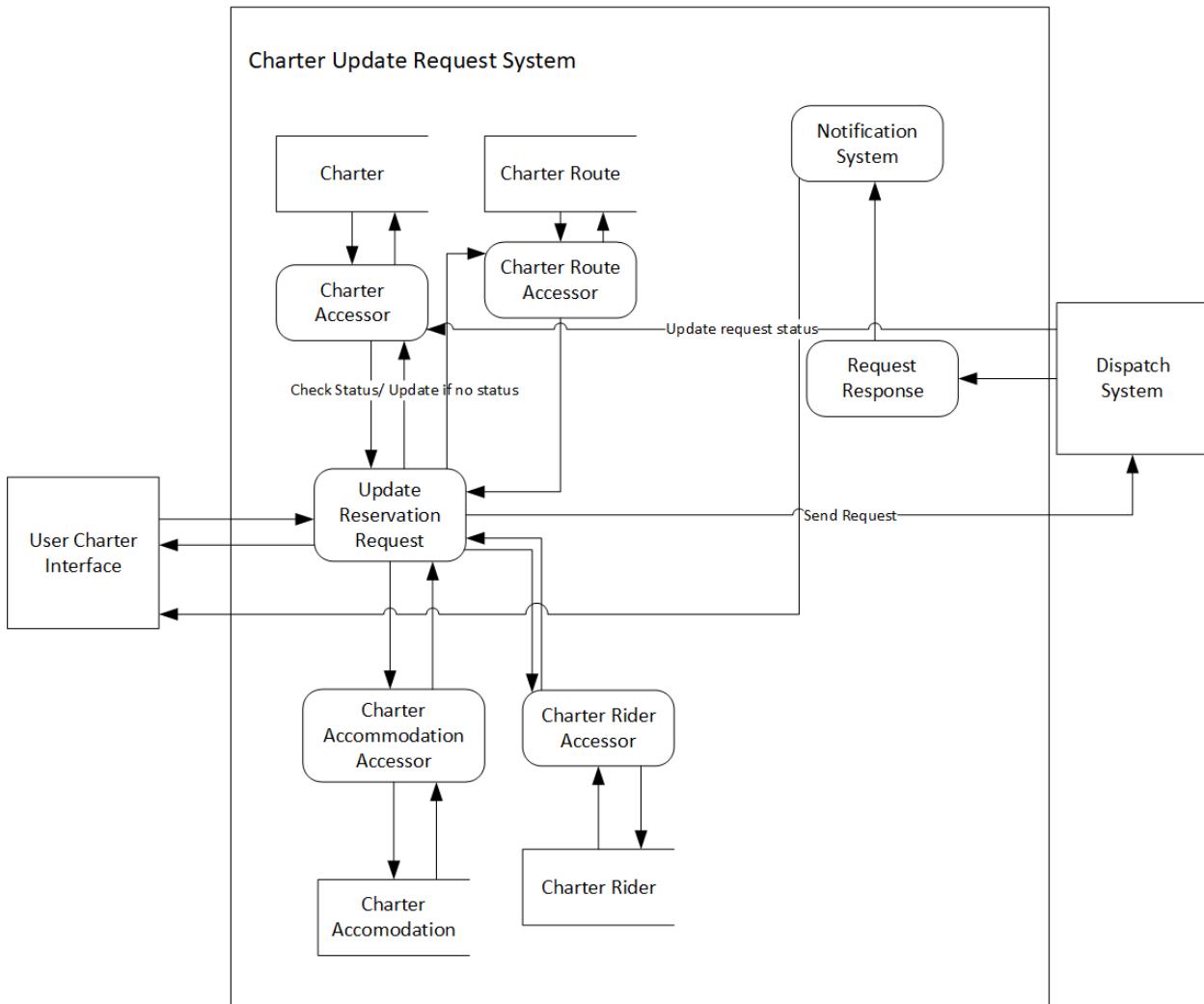
## Cancel Request



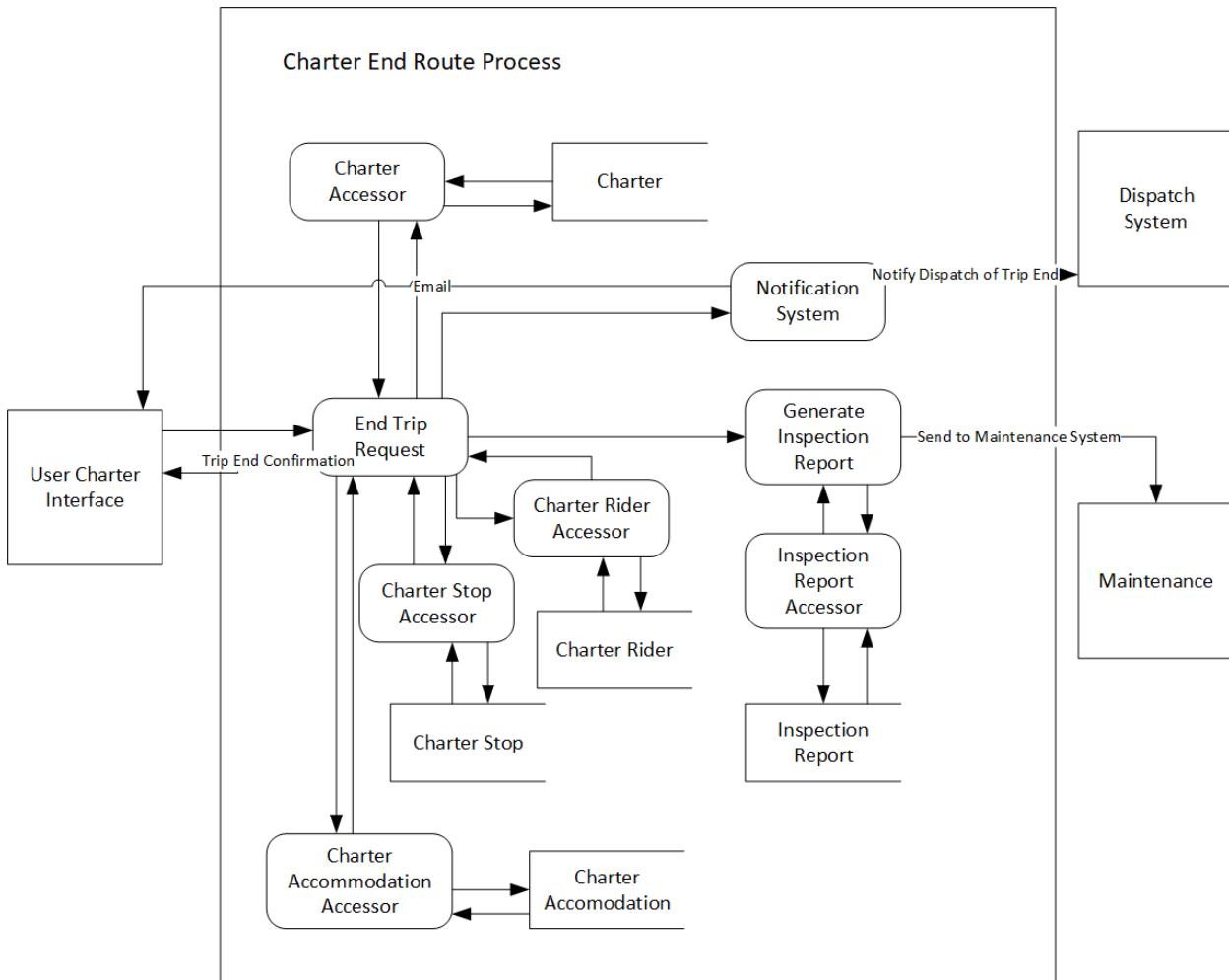
## Requesting a Charter



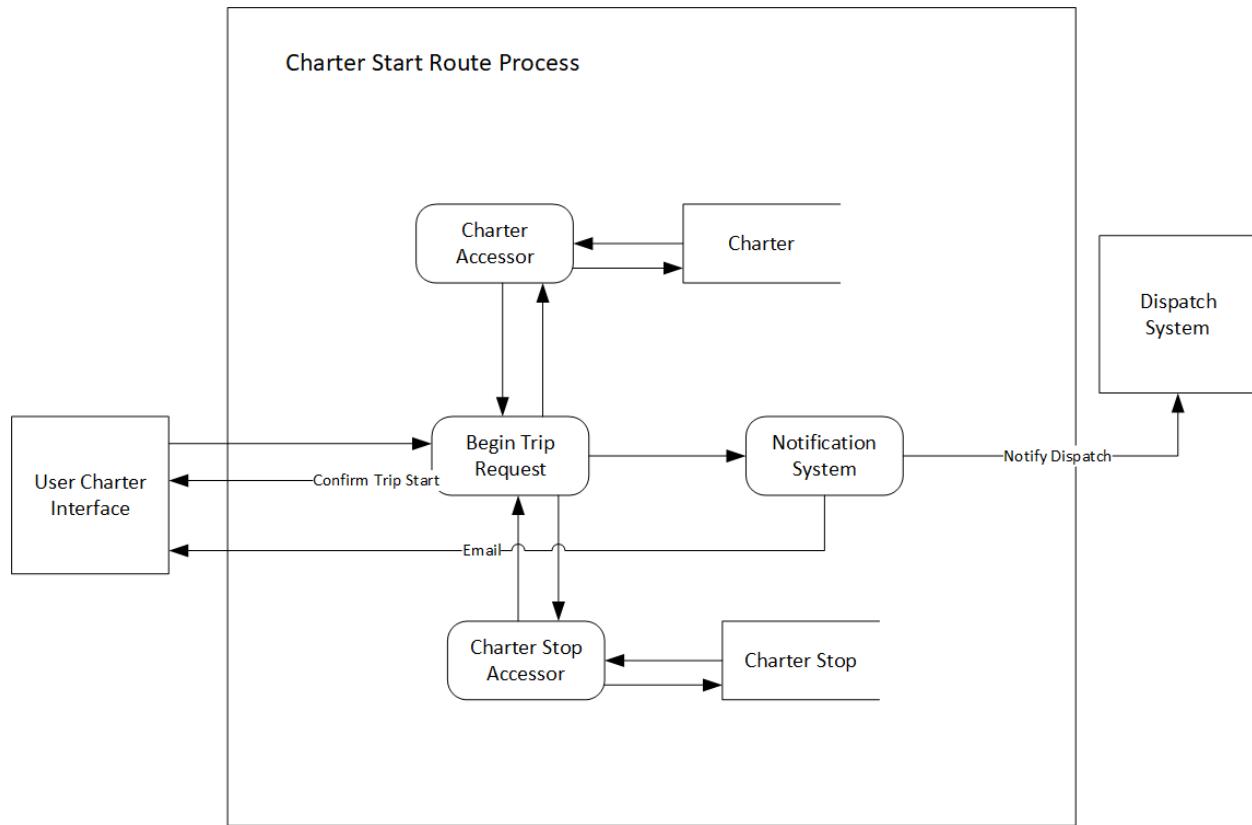
## Updating a Charter Request



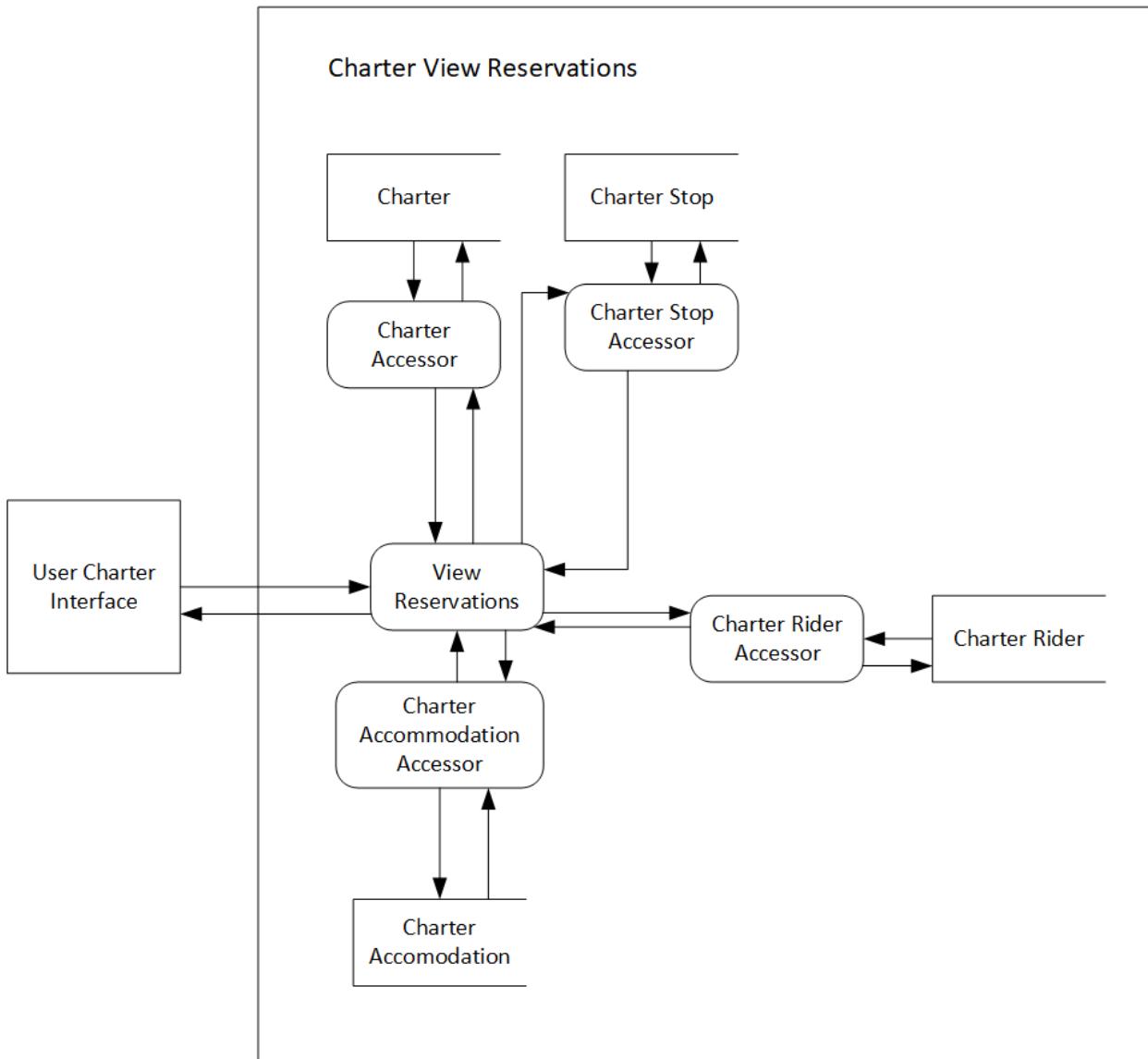
## Verify Route End



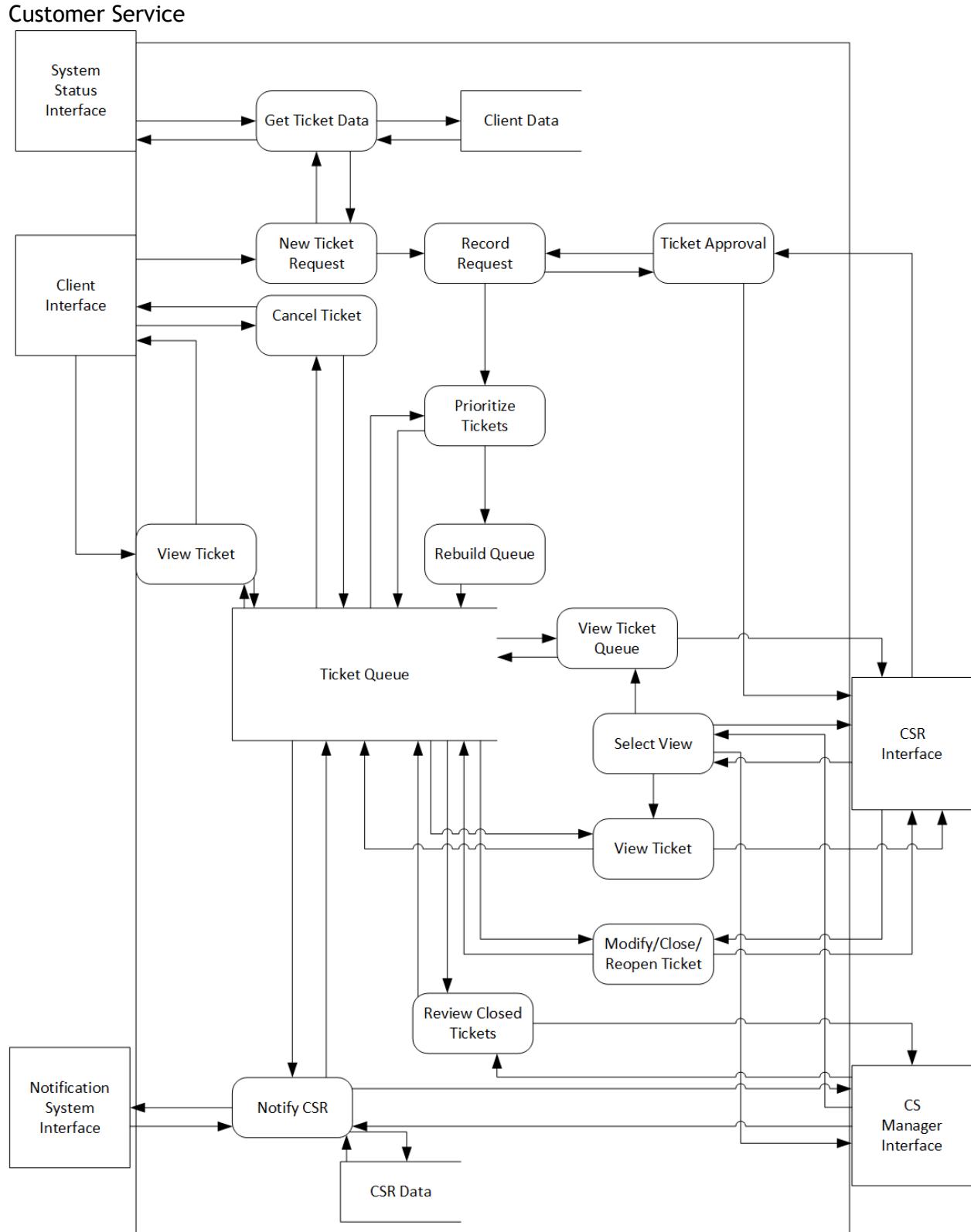
## Verify Route Start



## View Reservations

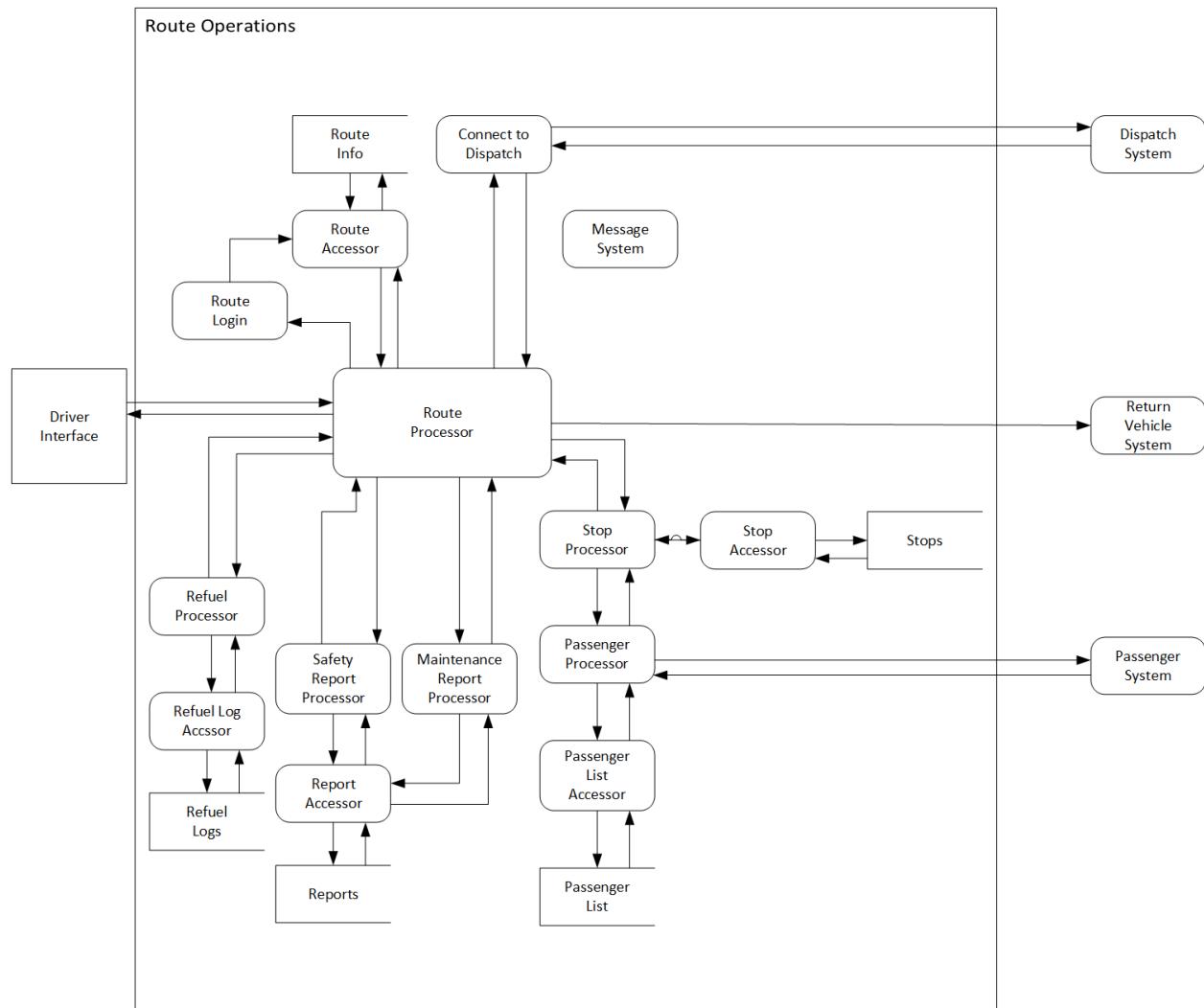


# Customer Service Subsystem



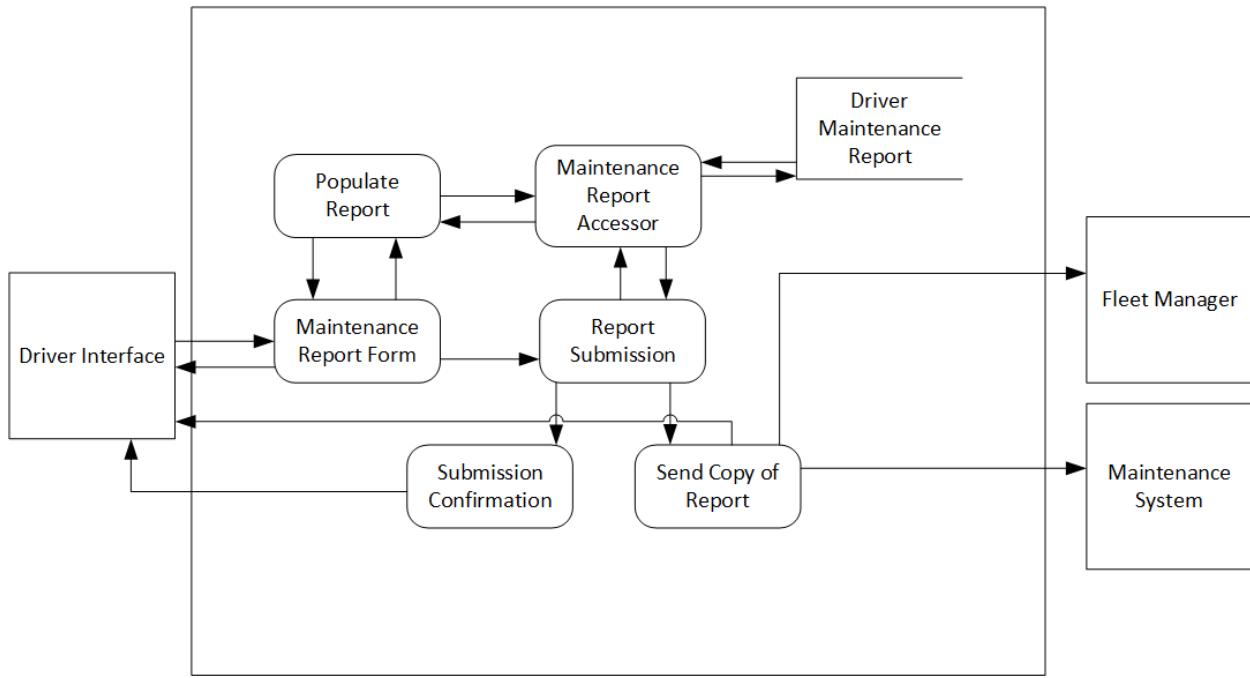
# Driver Subsystem

## General

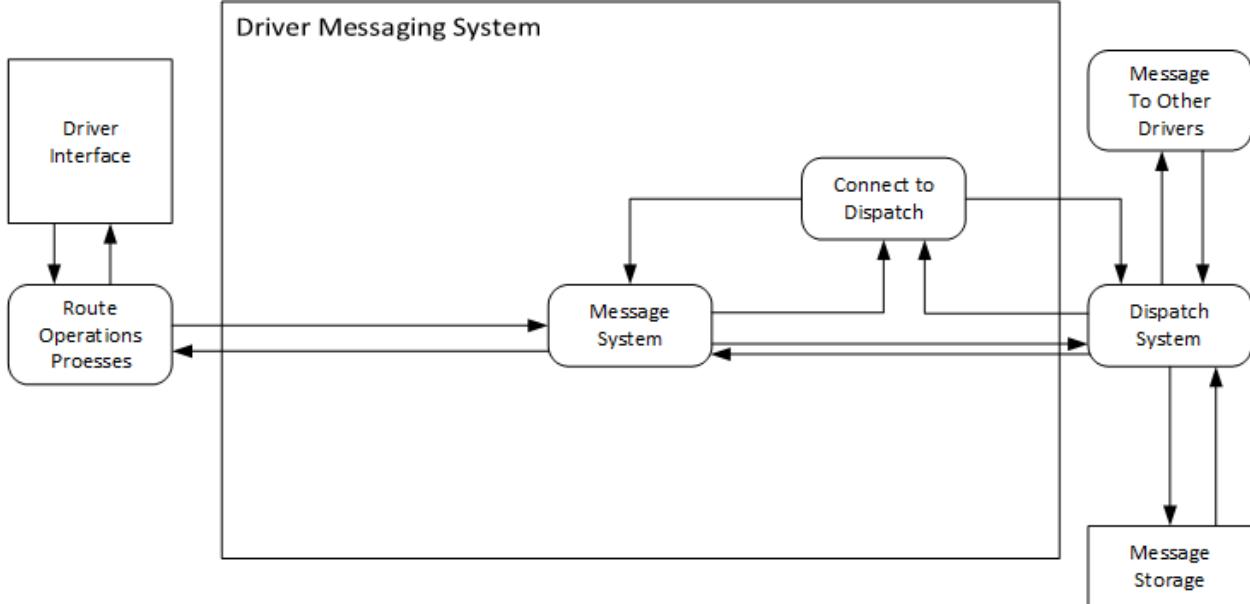


## Maintenance Report Support

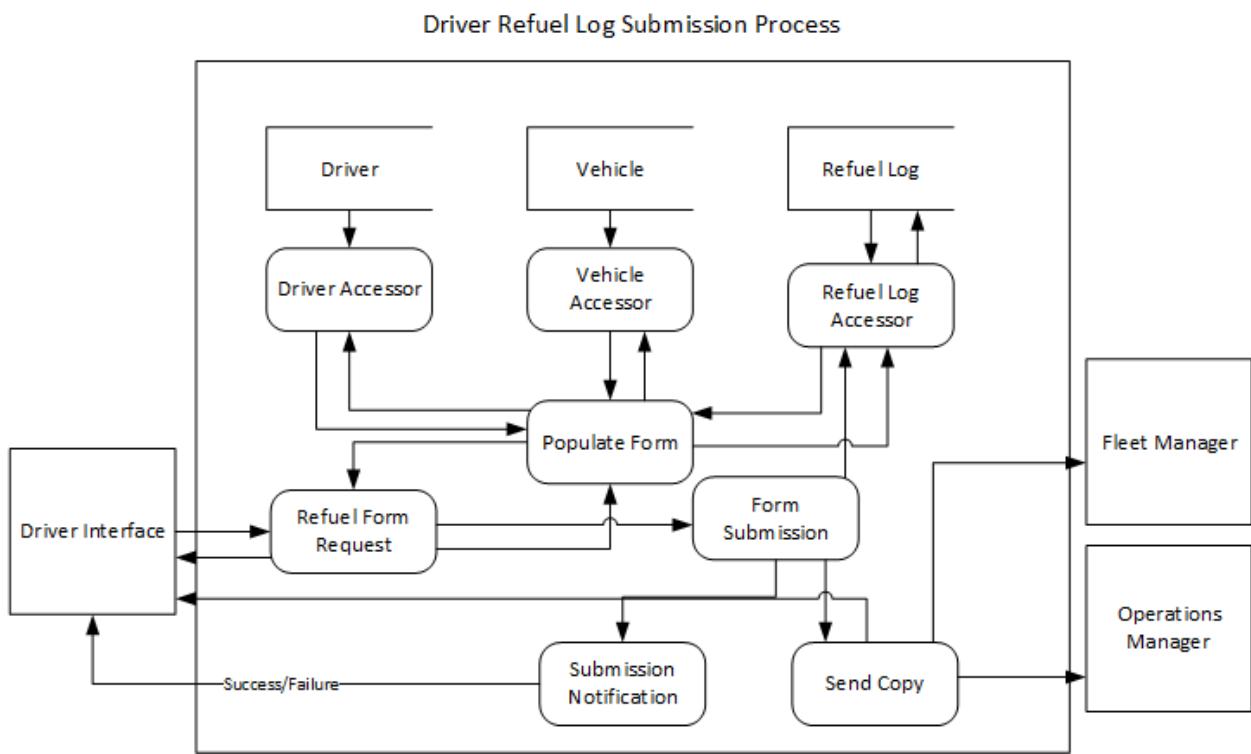
Driver Submit Maintenance Report Process



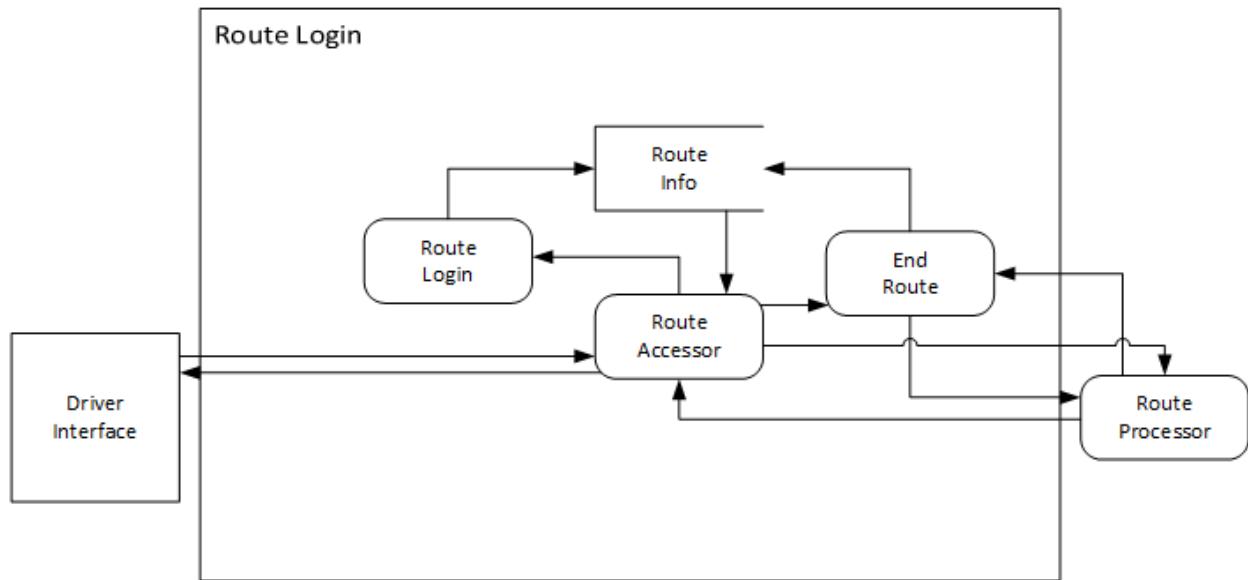
Messaging



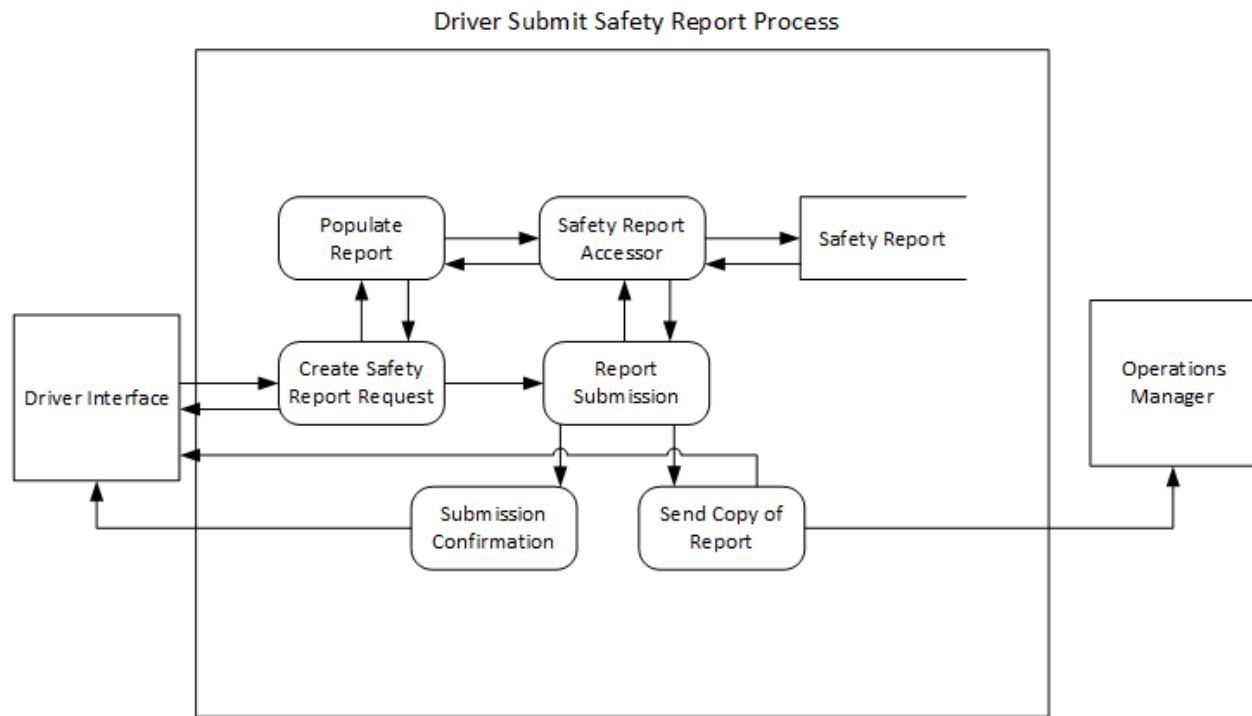
## Driver Refueling



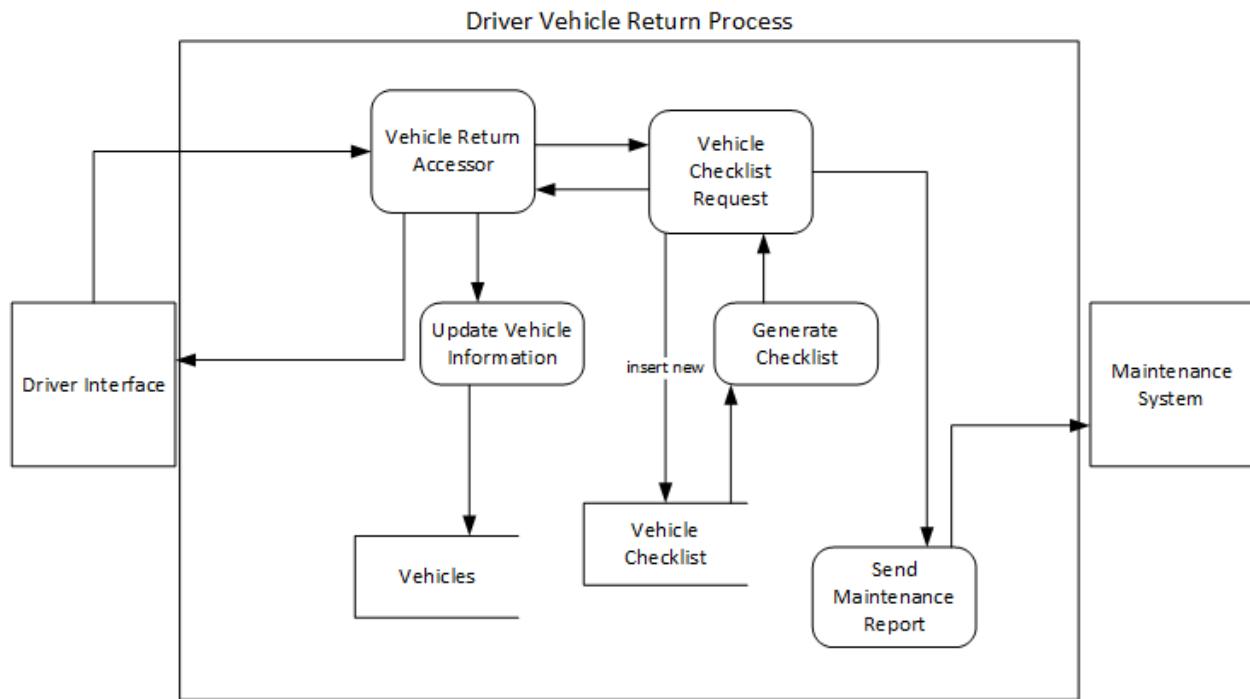
## Driver Route Login



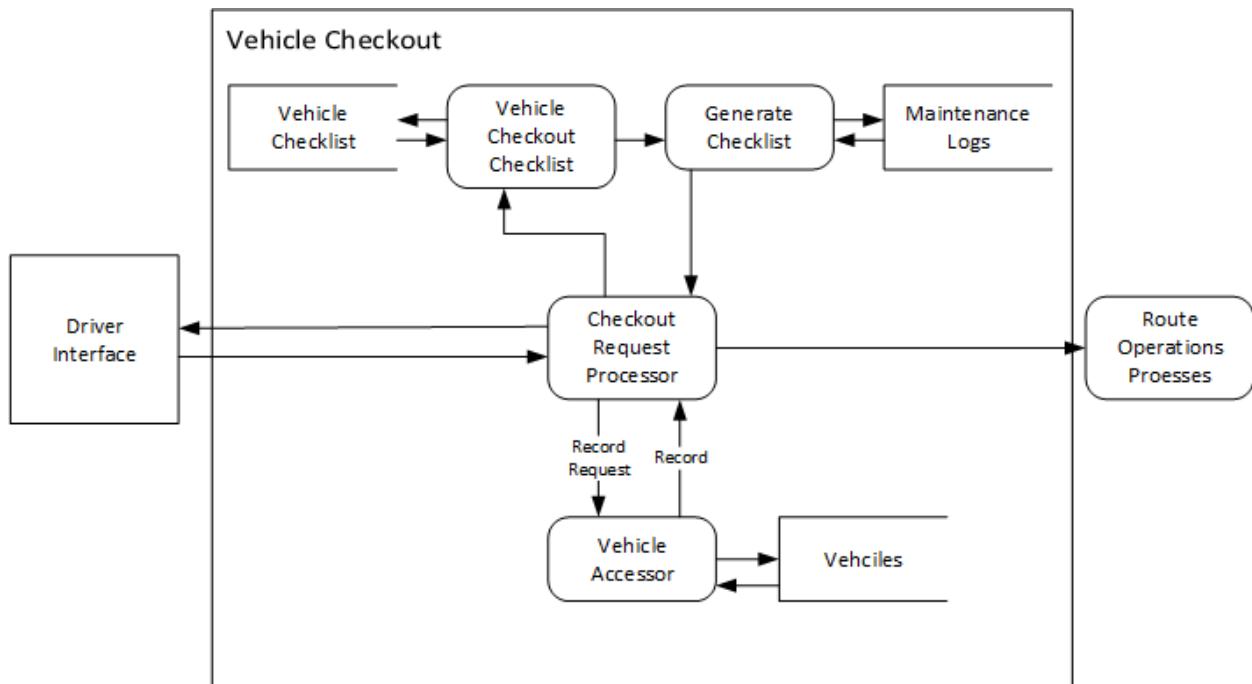
## Safety Reporting



## Vehicle Check-In

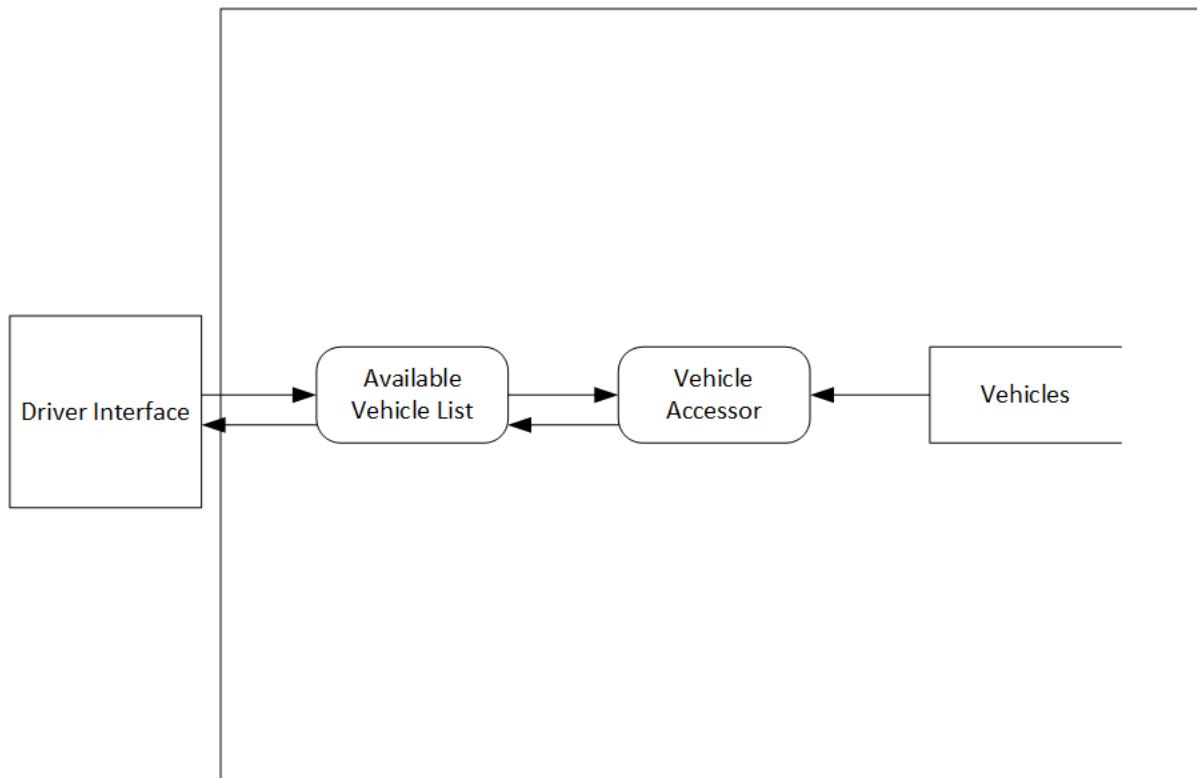


## Vehicle Check-Out



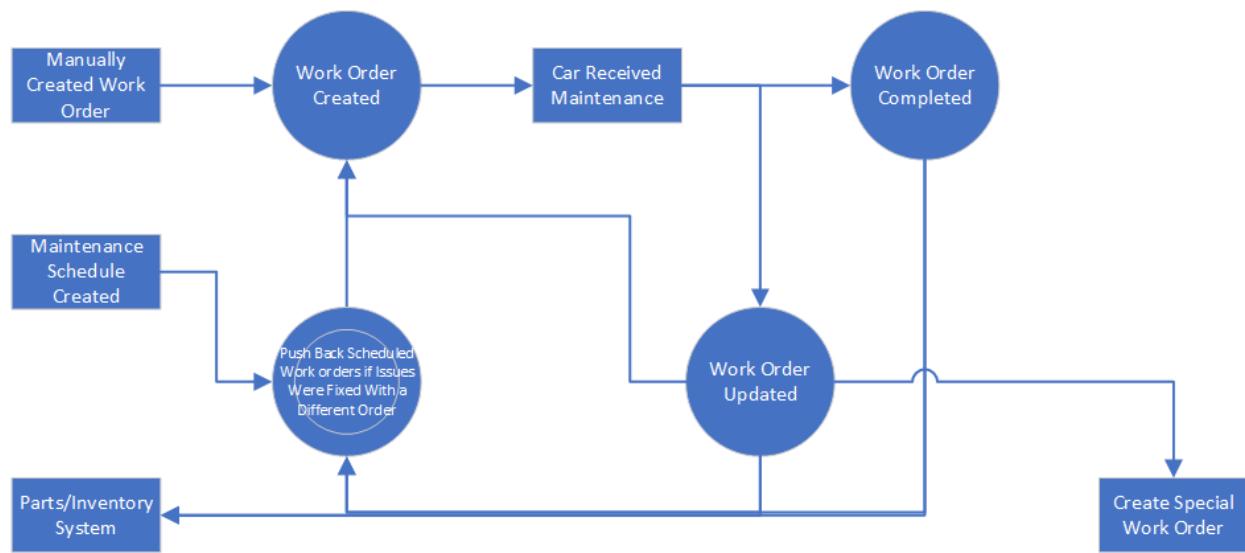
## View Available Vehicles

Driver View Available Vehicles Process



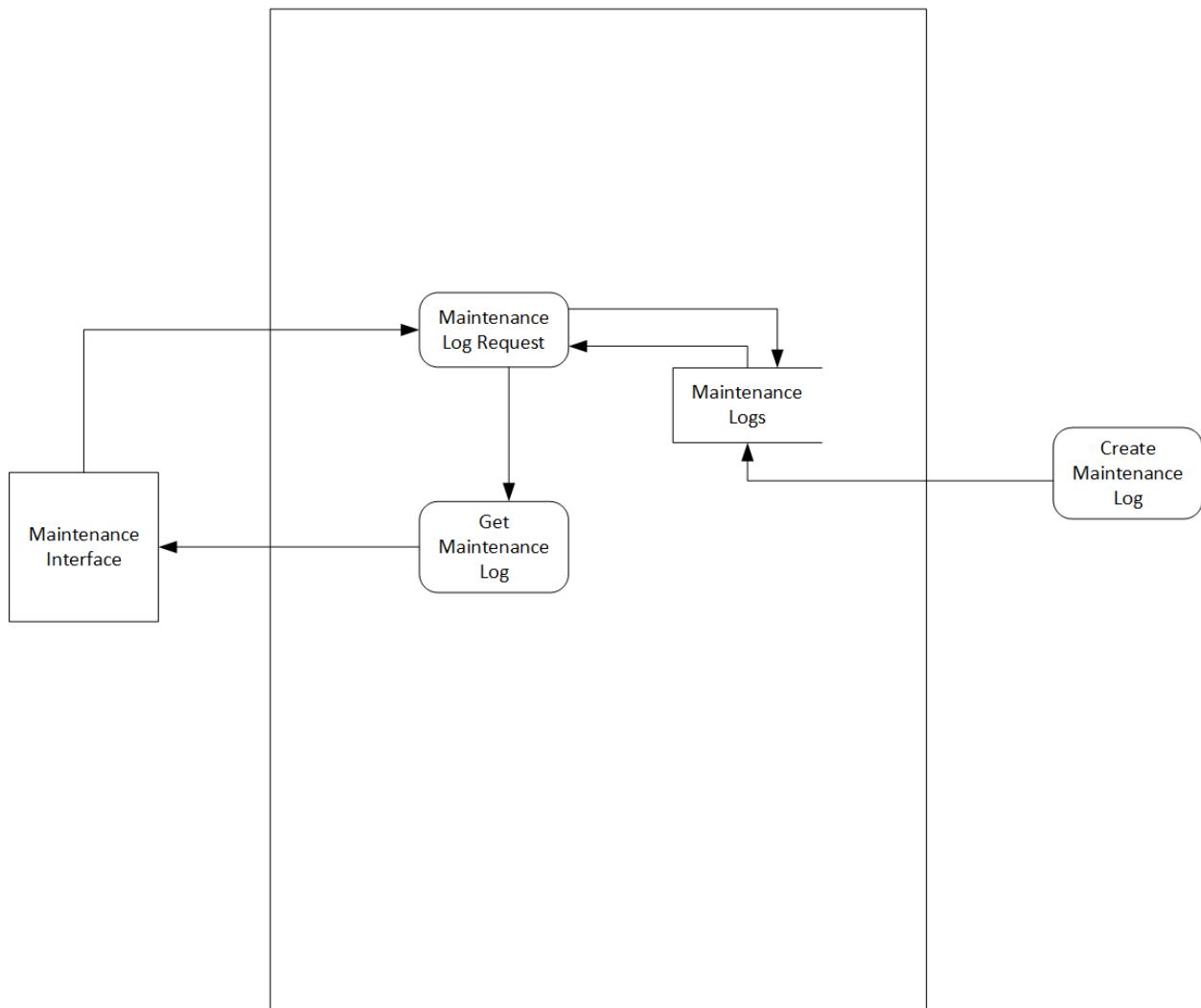
# Maintenance Subsystem

## Work Order Creation

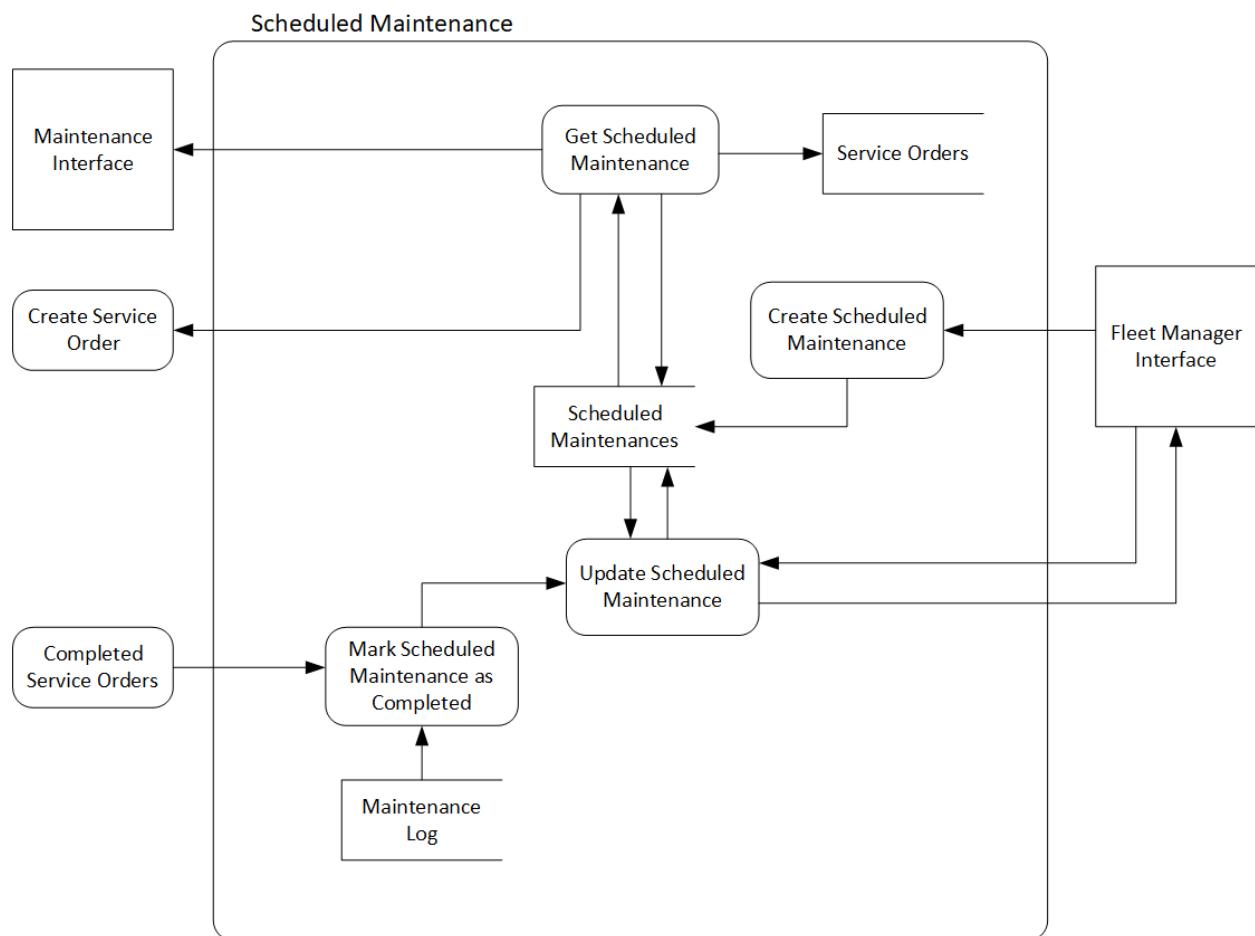


## View Maintenance Logs

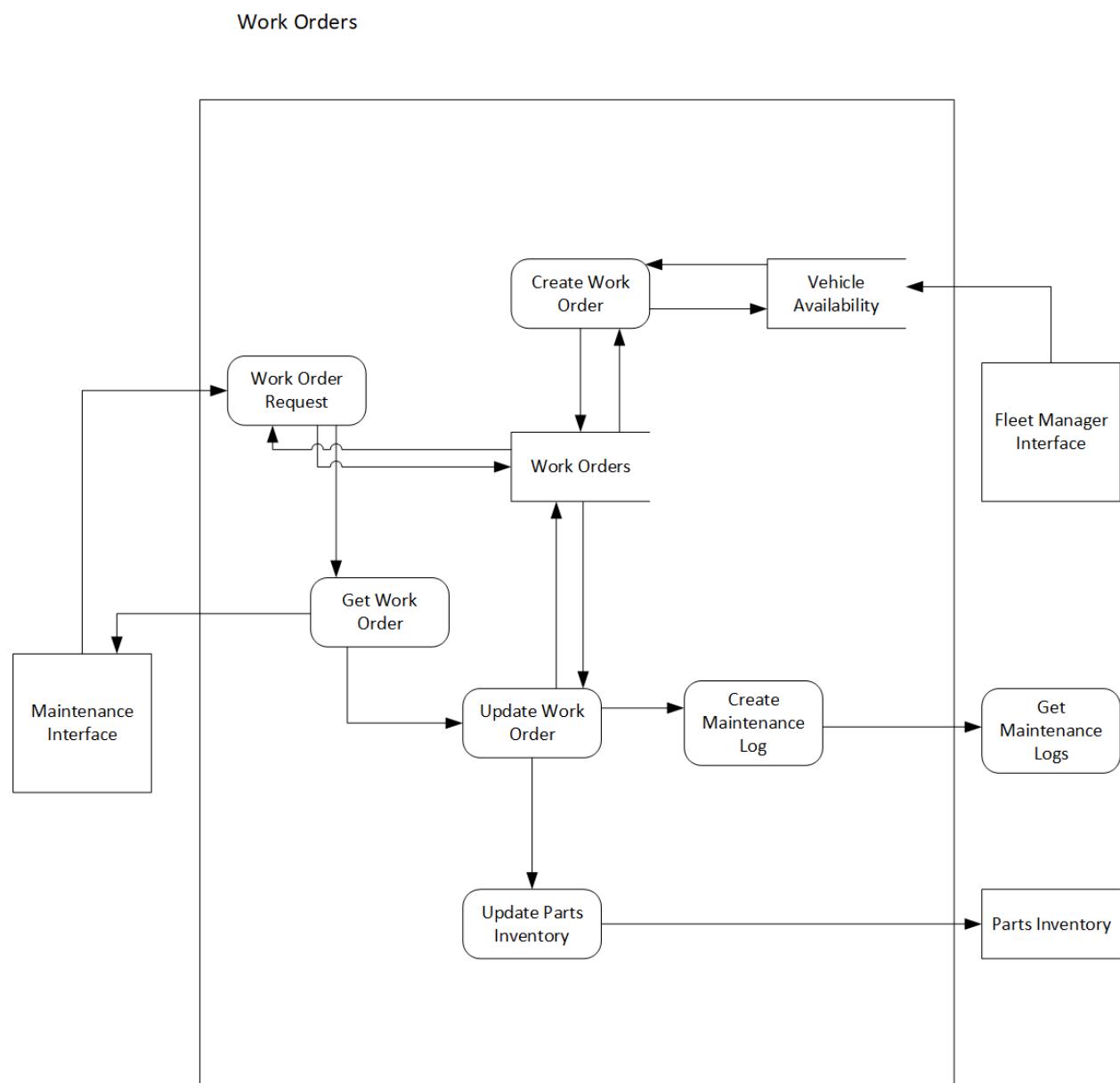
Maintenance Logs



## Scheduled Maintenance

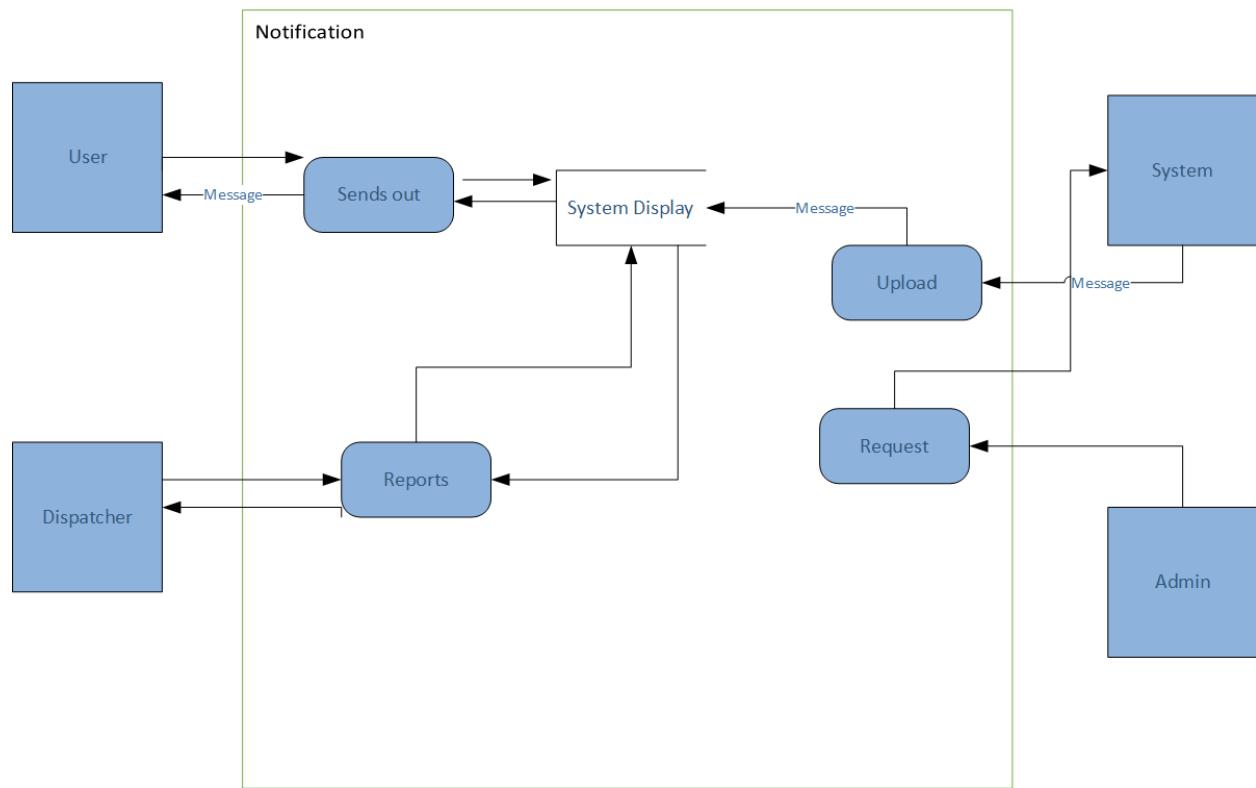


## Work Order



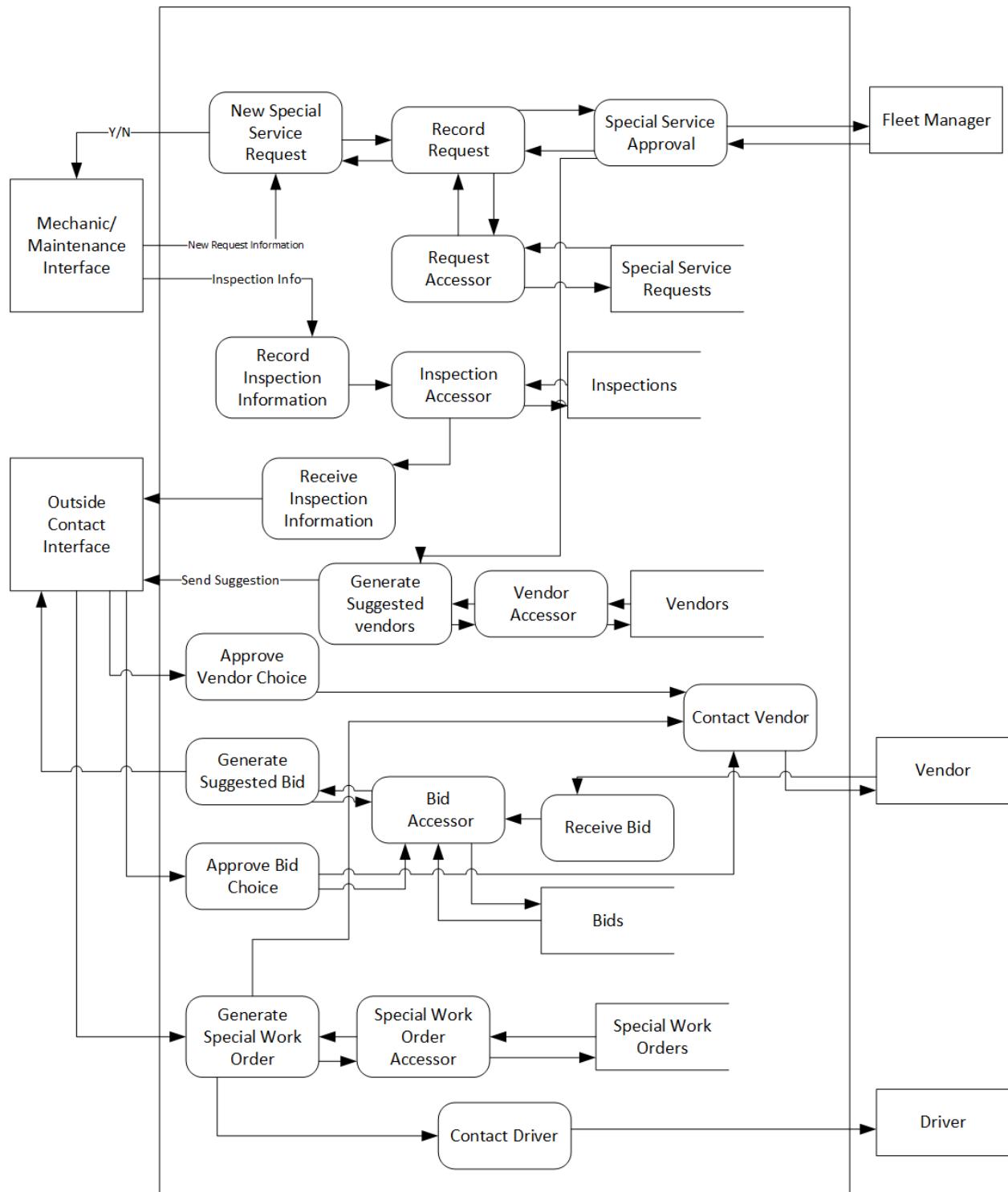
# Notification Subsystem

## Notification



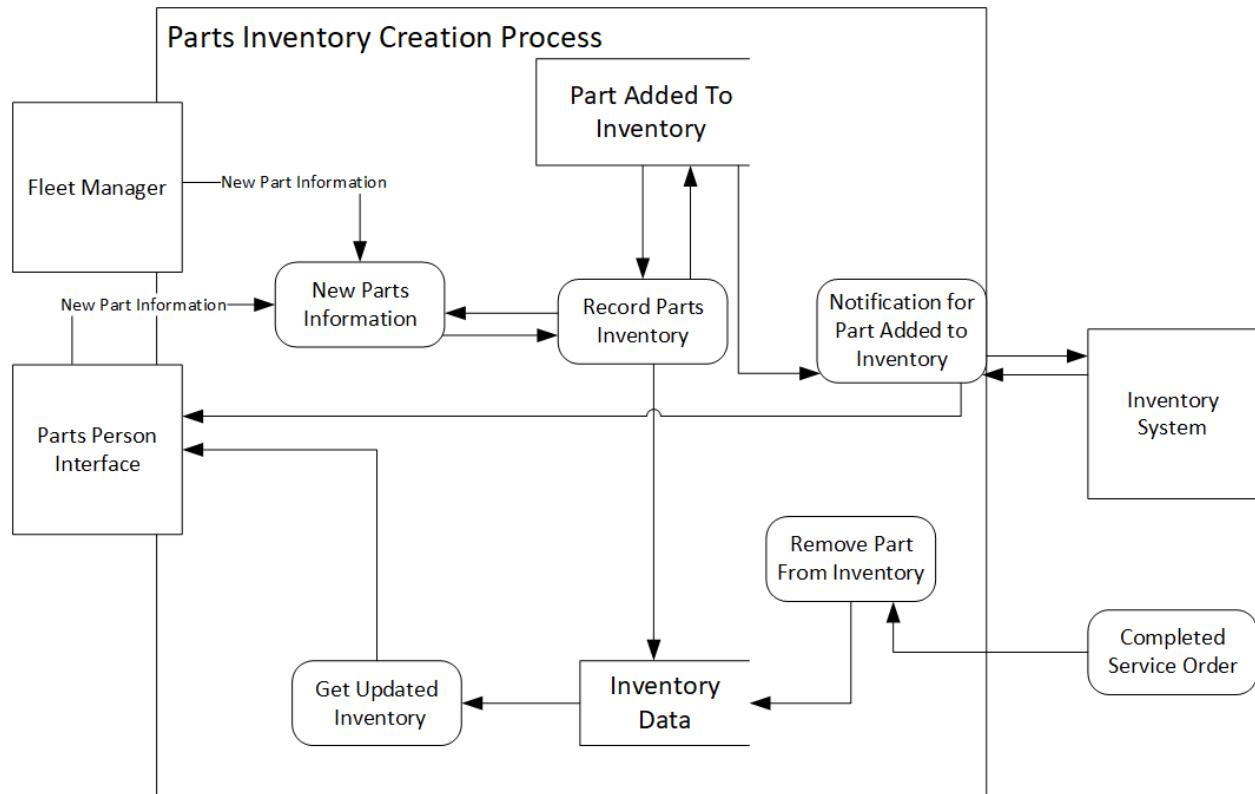
# Outside Contact Subsystem

## Outside Contact

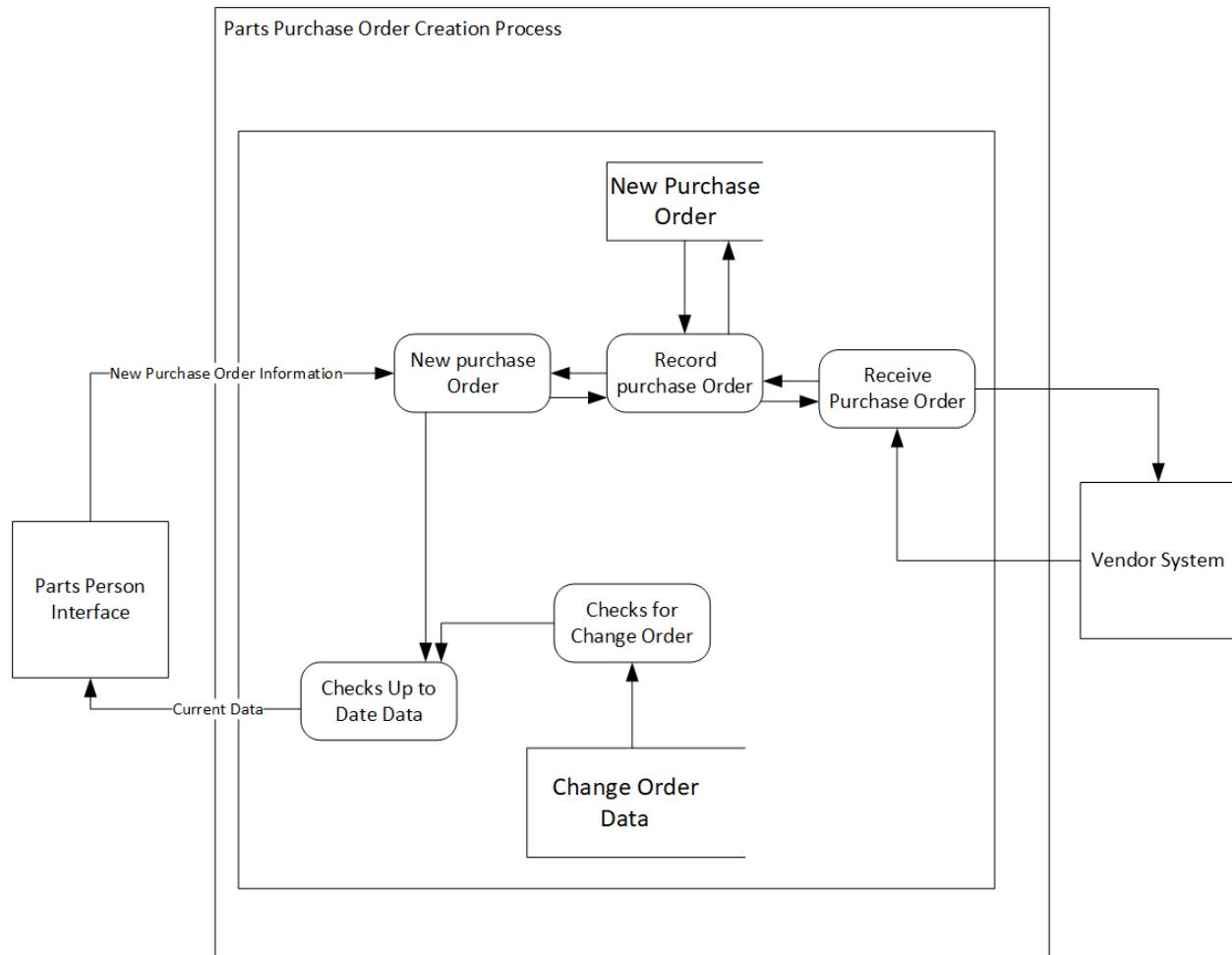


## Parts Subsystem

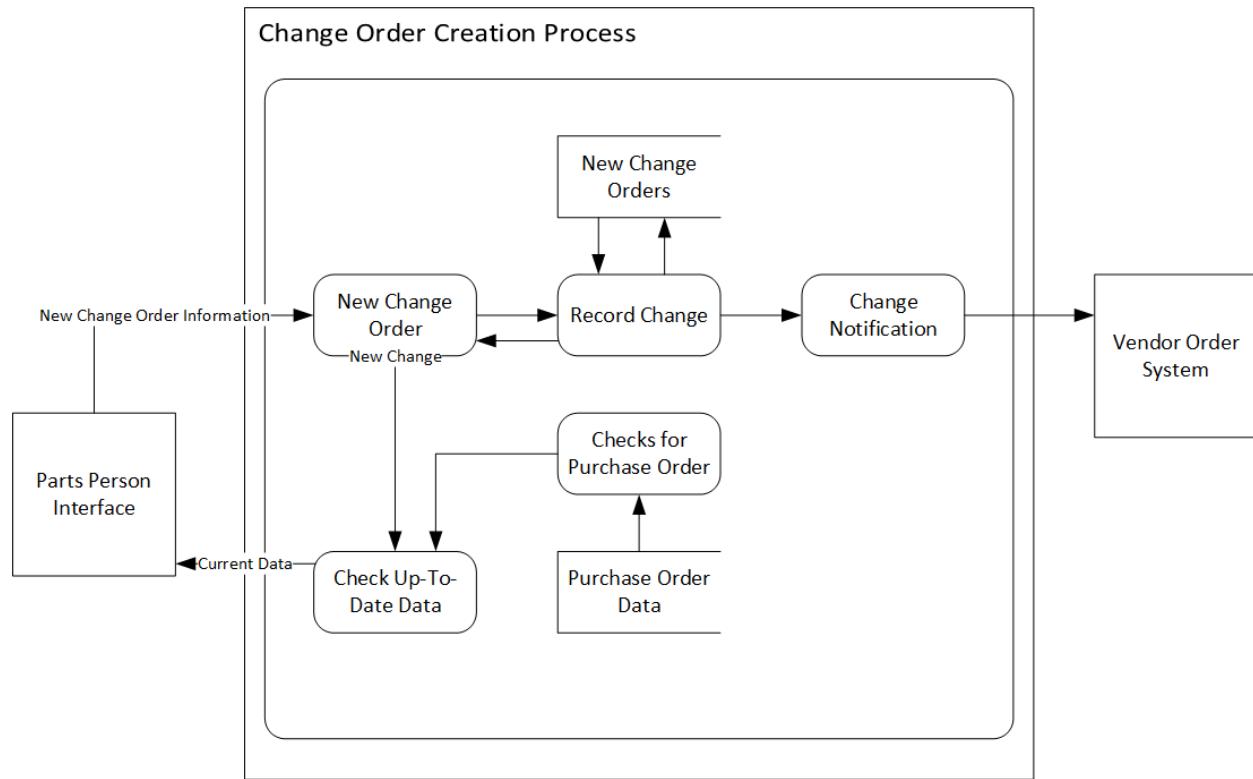
### Part Inventory



## Purchase Order

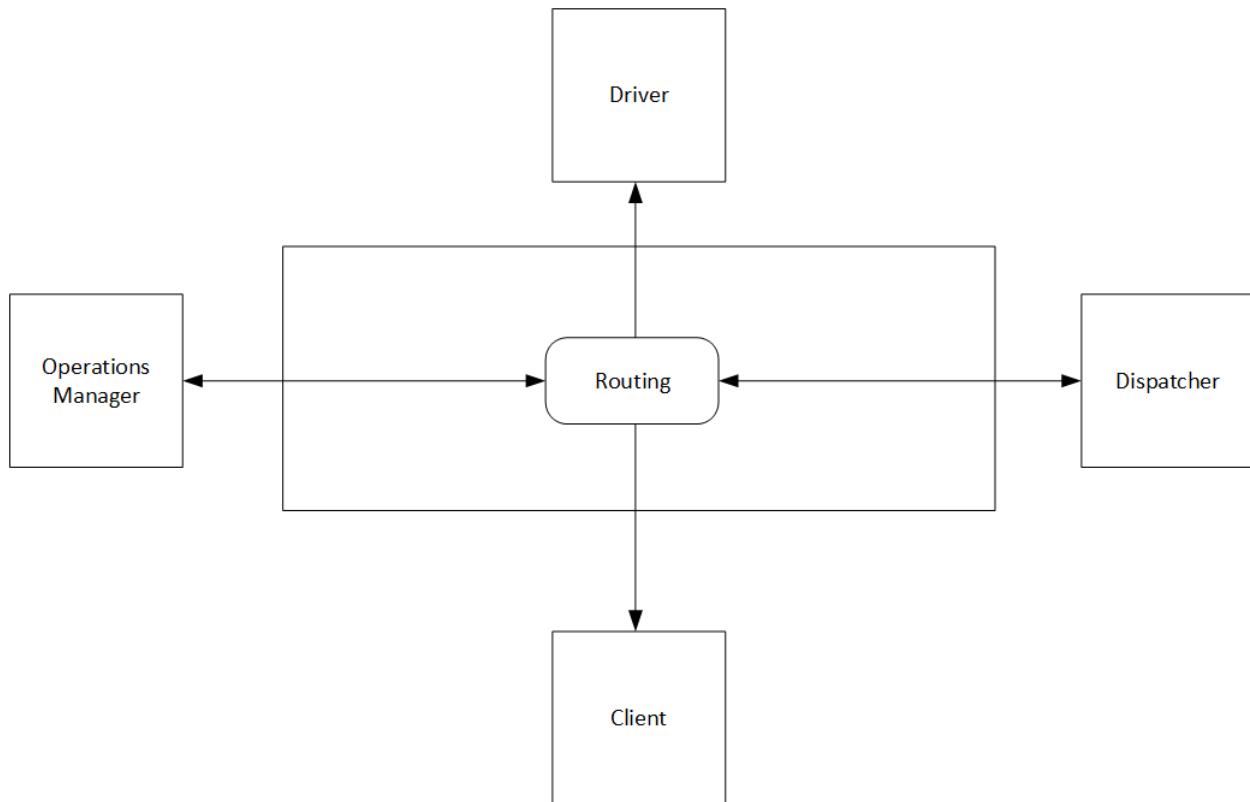


## Change Order

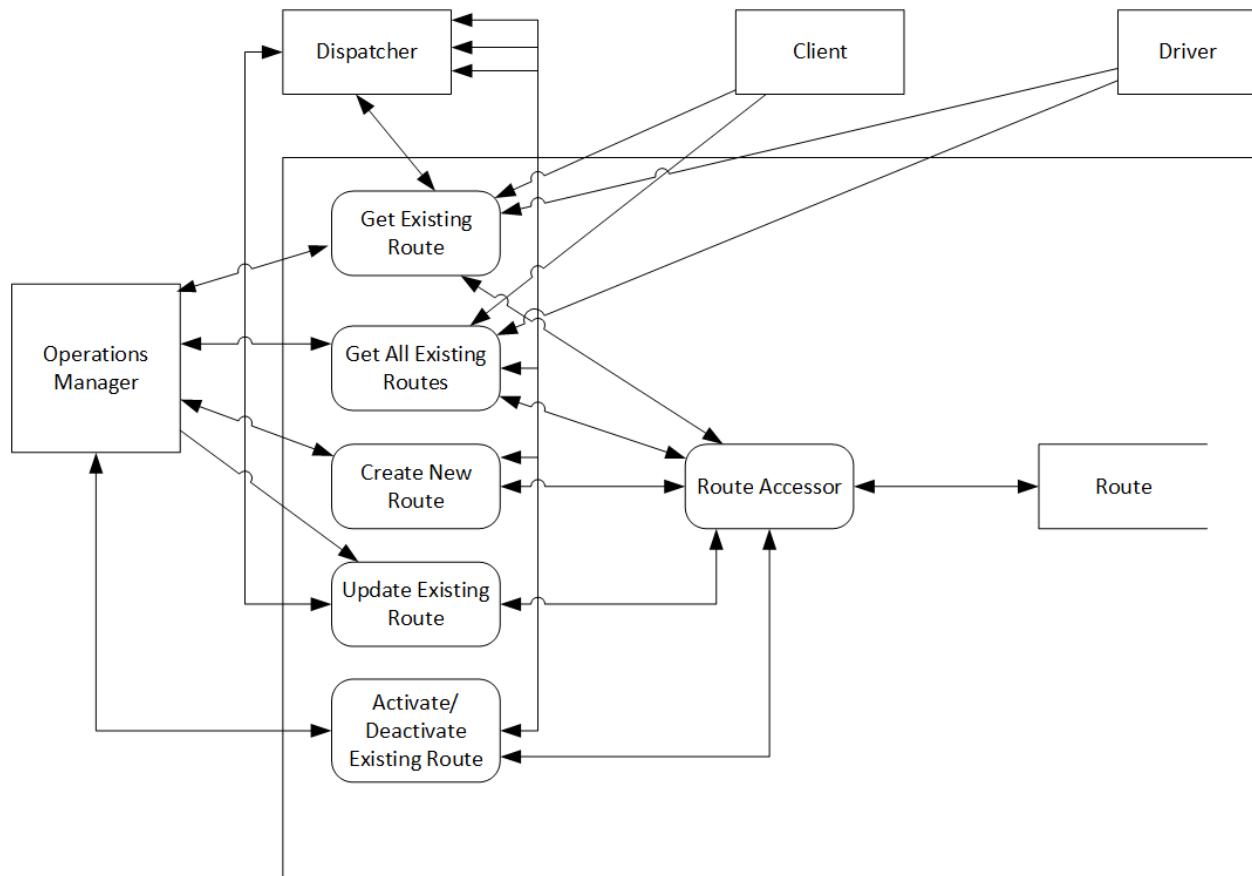


## Route Subsystem

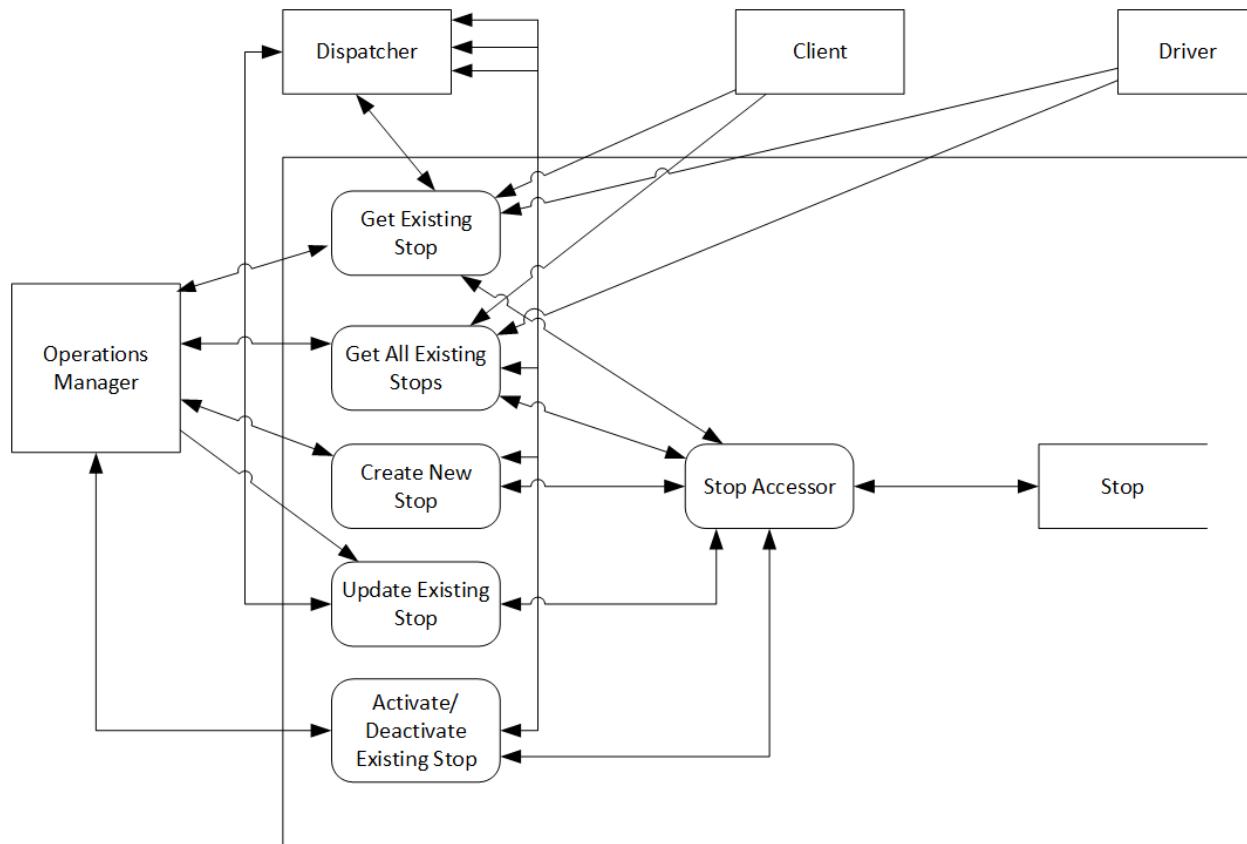
### Interaction of Roles



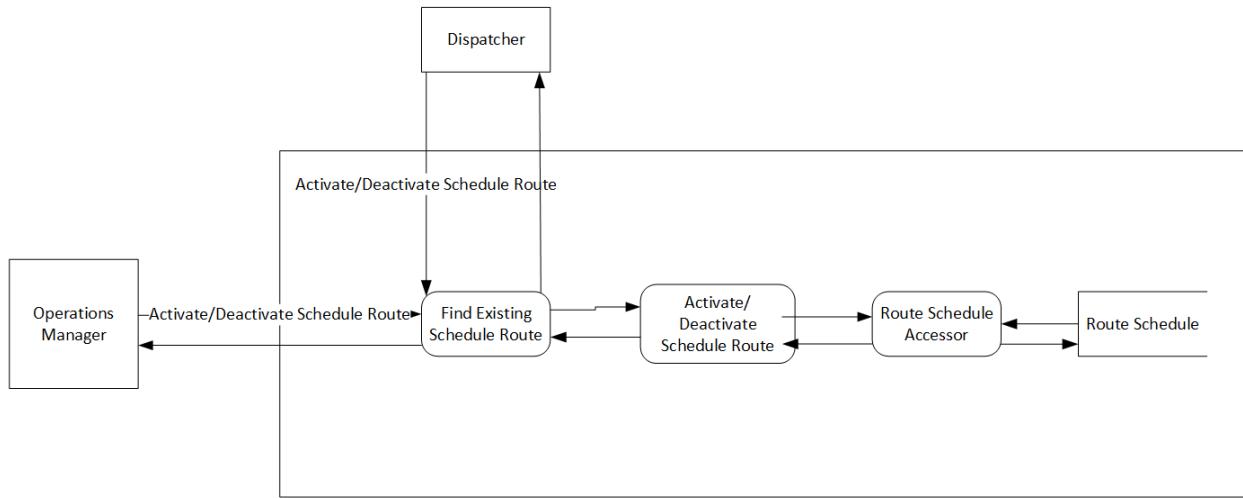
## General Route Detailed



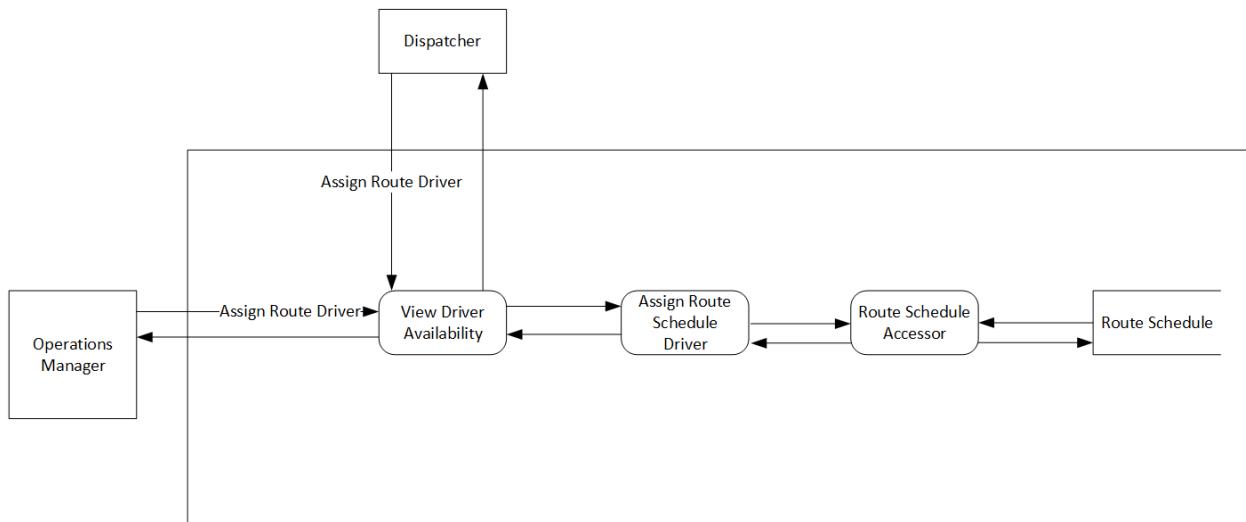
## General Stop Detailed



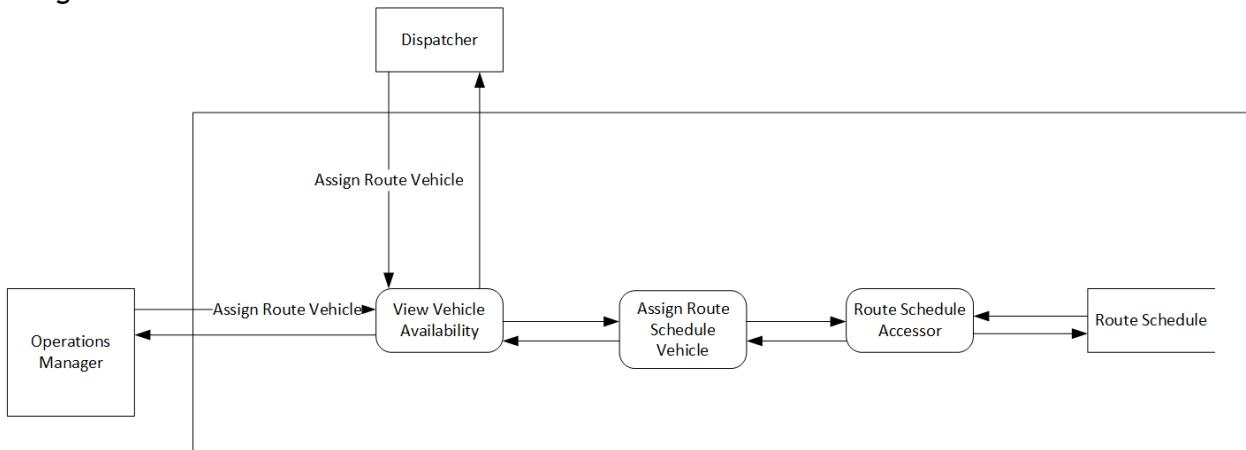
## Update Route Schedule Activation



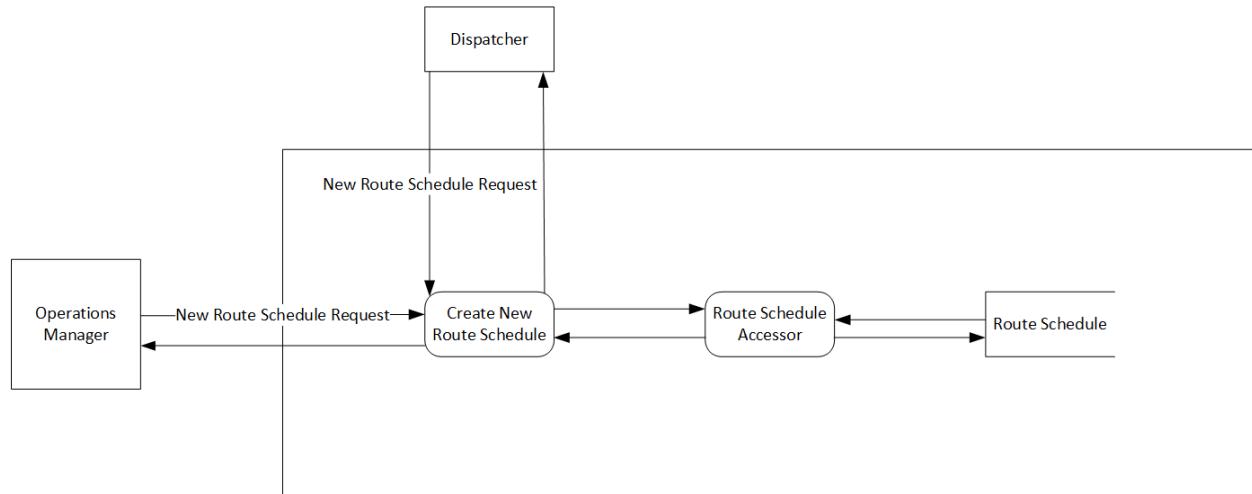
## Assign Driver to Route



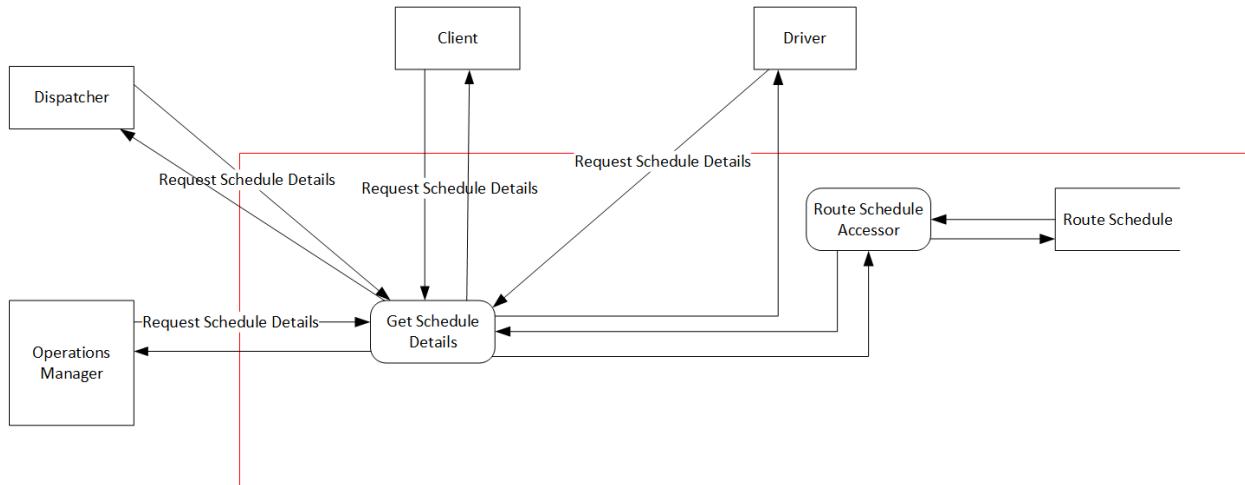
## Assign Vehicle to Route



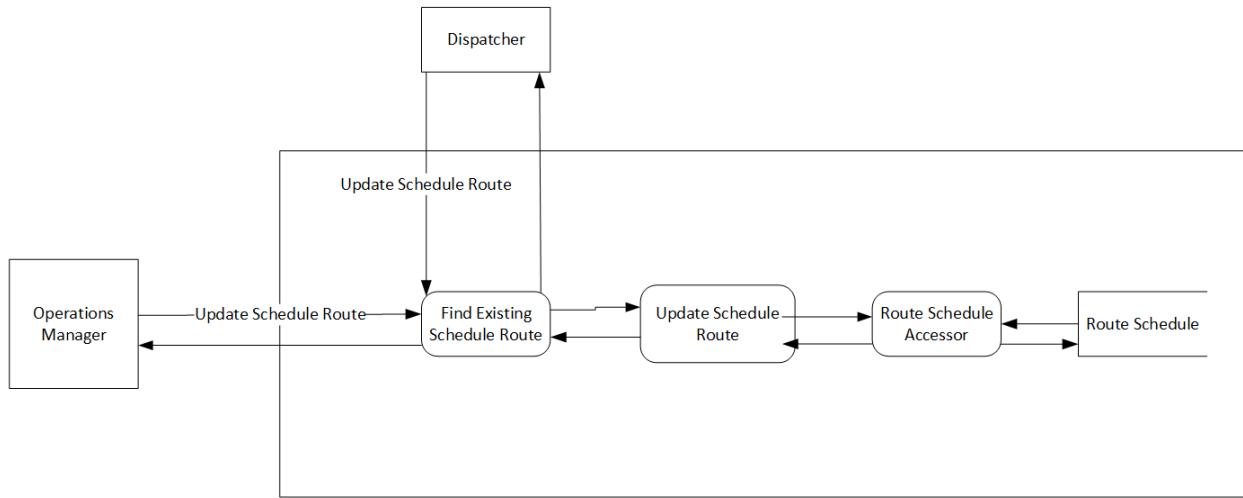
## Create Route Schedule



## View Route Schedule Details



## Update Route Schedule

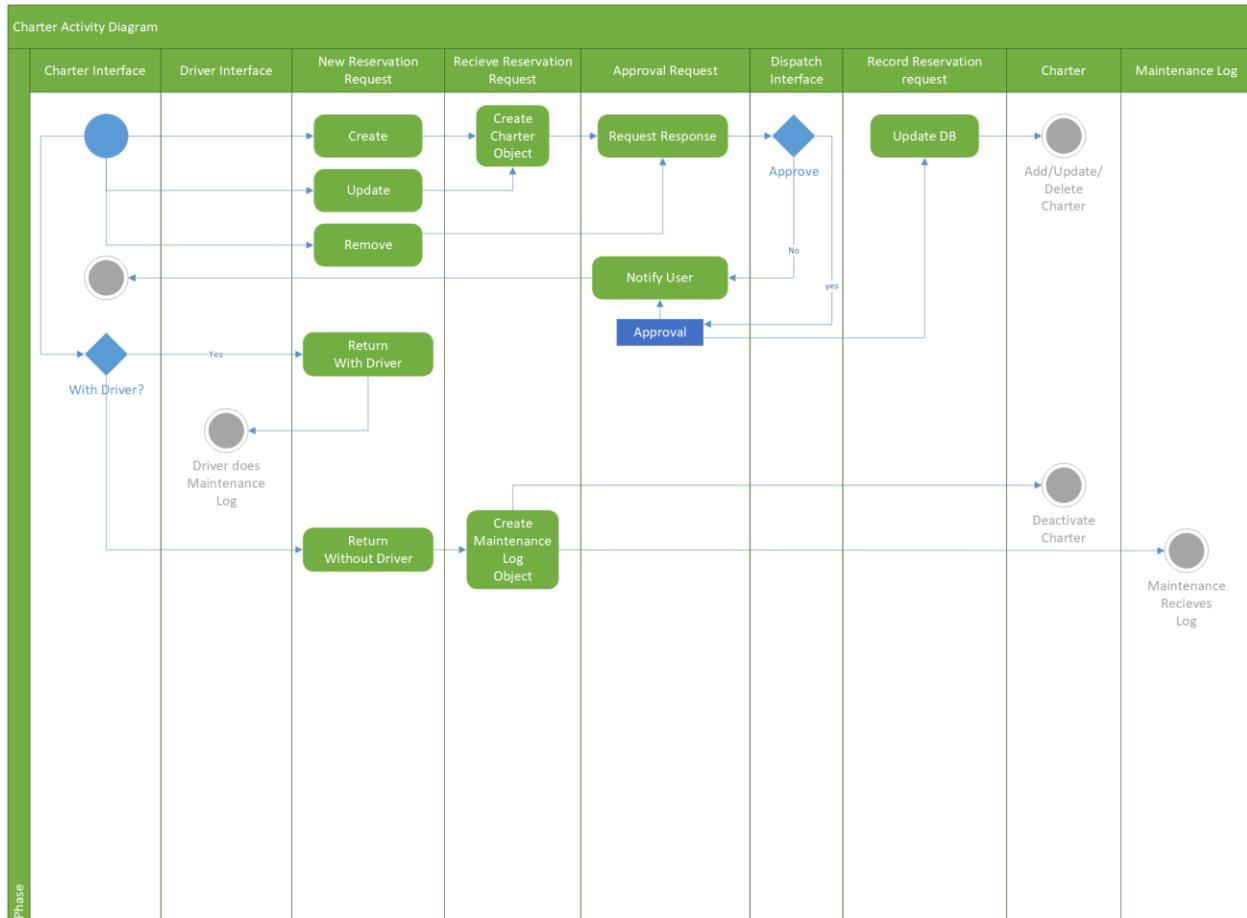


## Activity Diagrams by Subsystem

The Activity Diagrams describe the logical flows of functionality between the user interface, the application, and the database. Unlike the Data Flow diagram, the Activity Diagram describes in detail the branches of functionality and the conditions under which branches are taken in certain points during an application process.

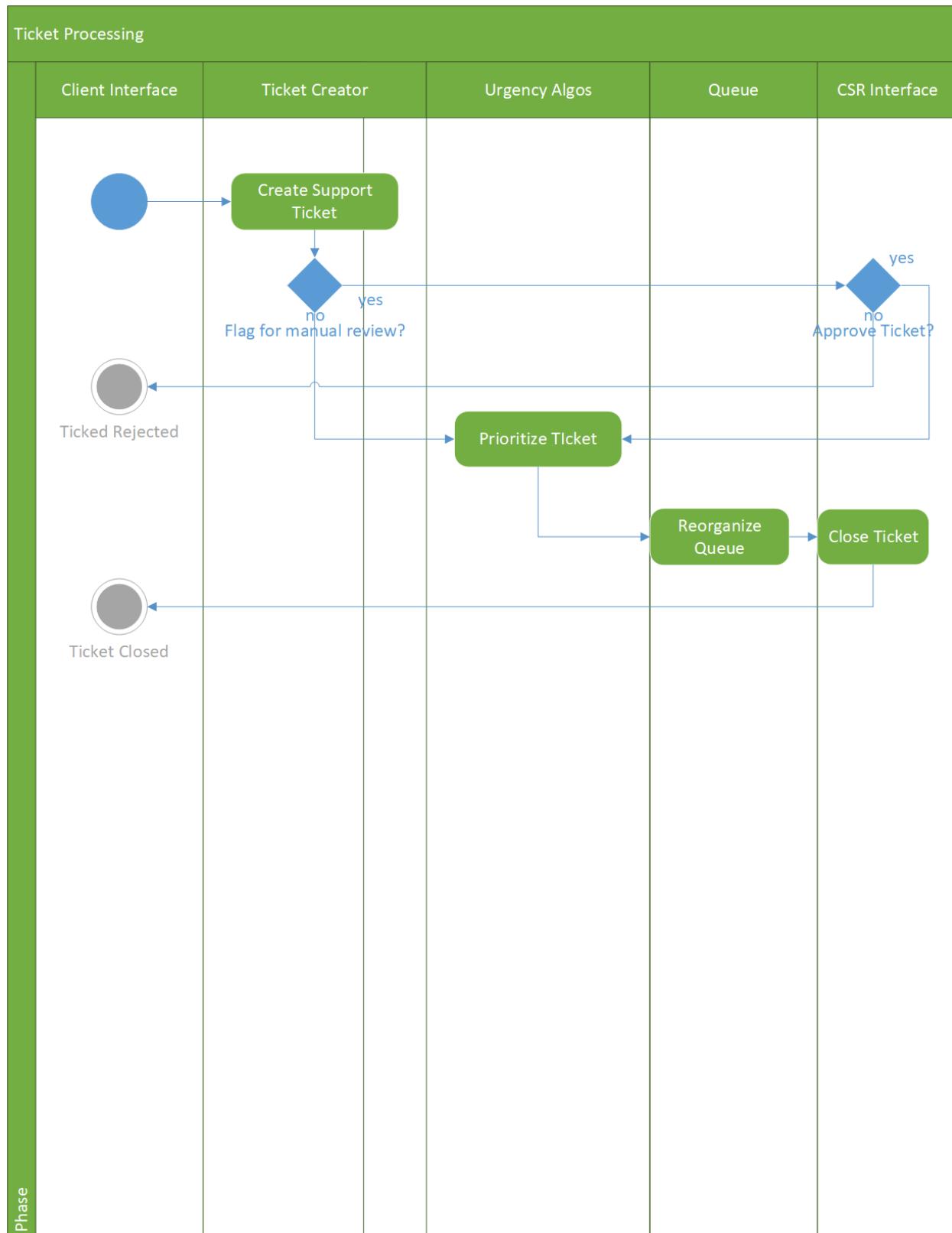
# Charter Subsystem

## Charter



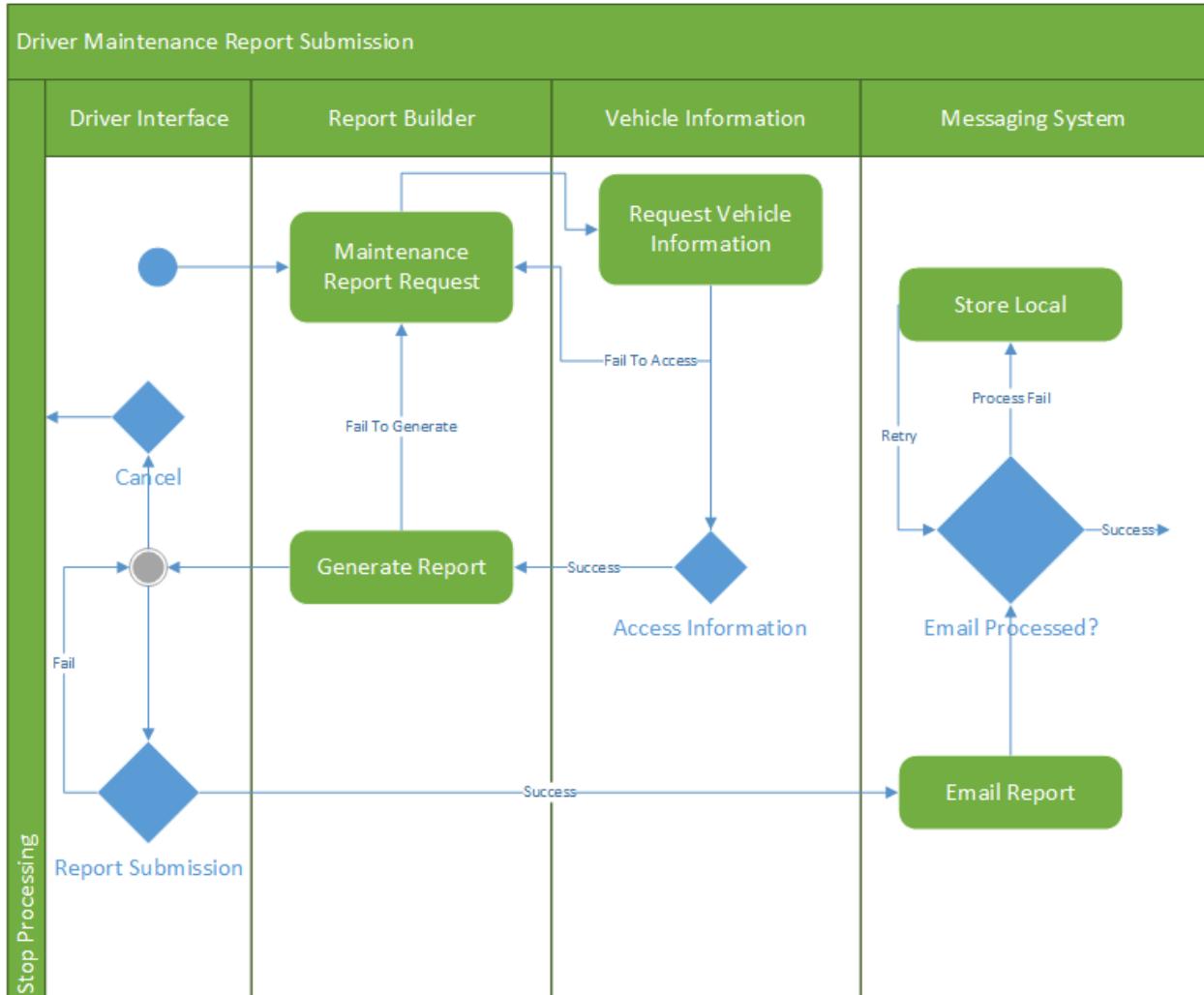
# Customer Service Subsystem

## Tickets

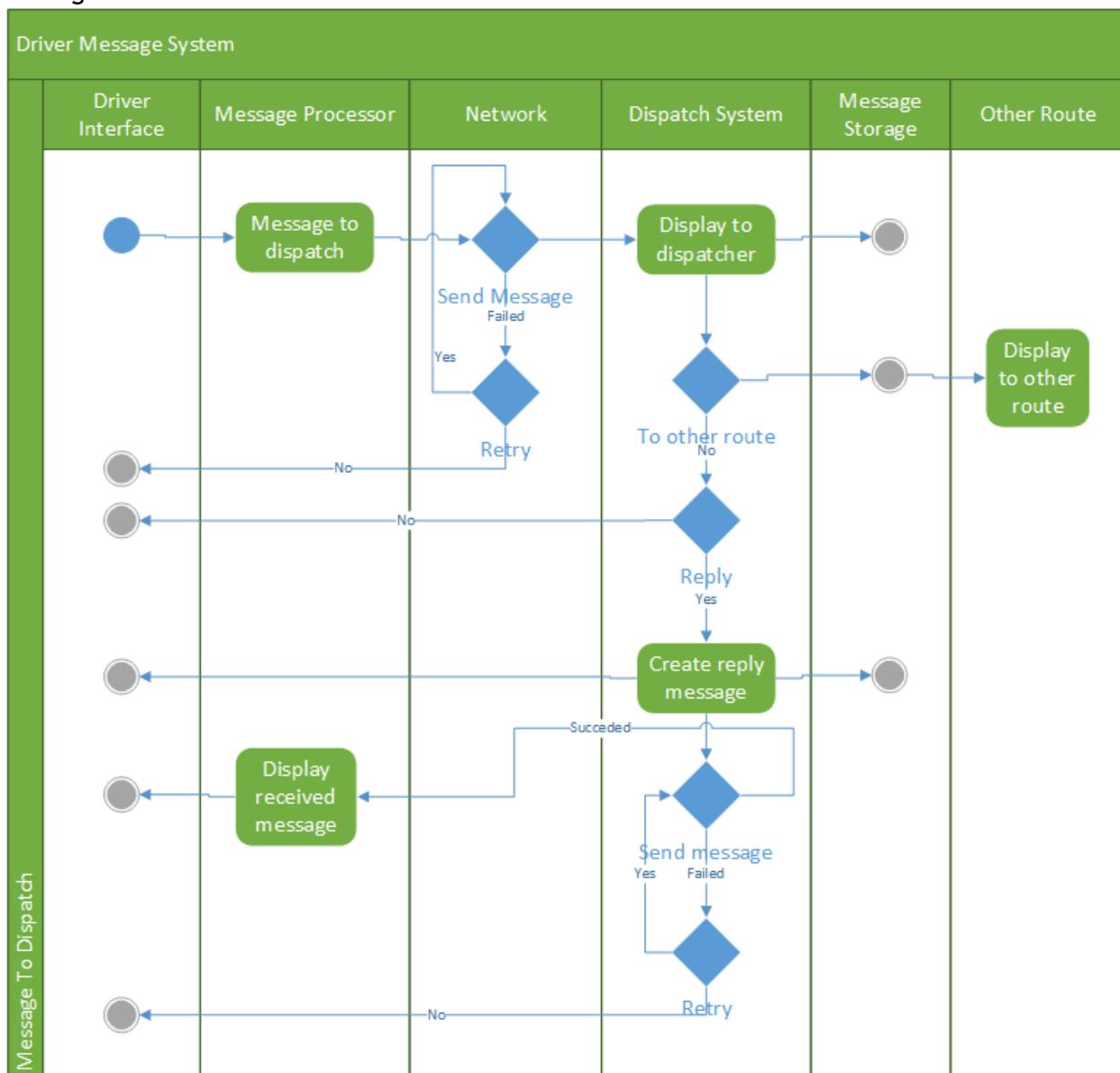


# Driver Subsystem

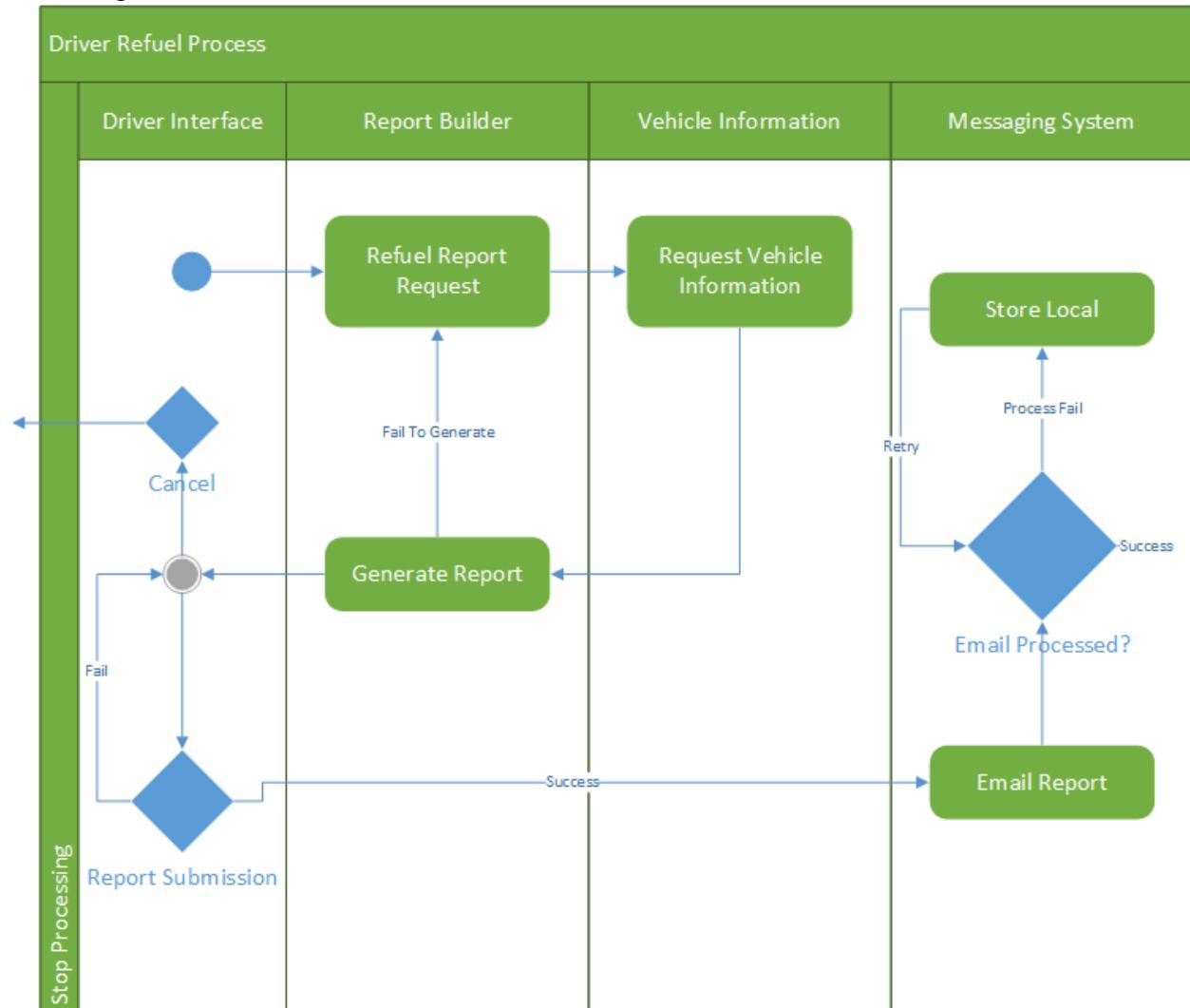
## Maintenance Reports



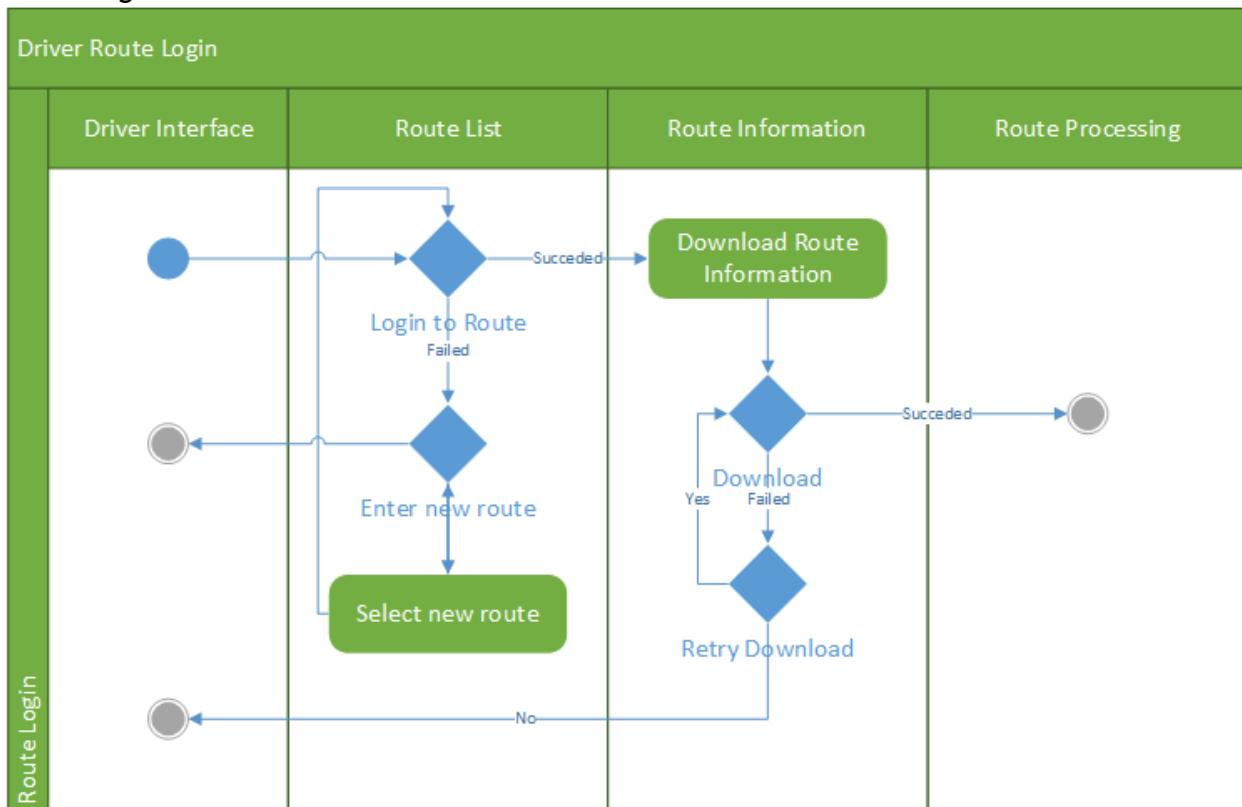
## Messages



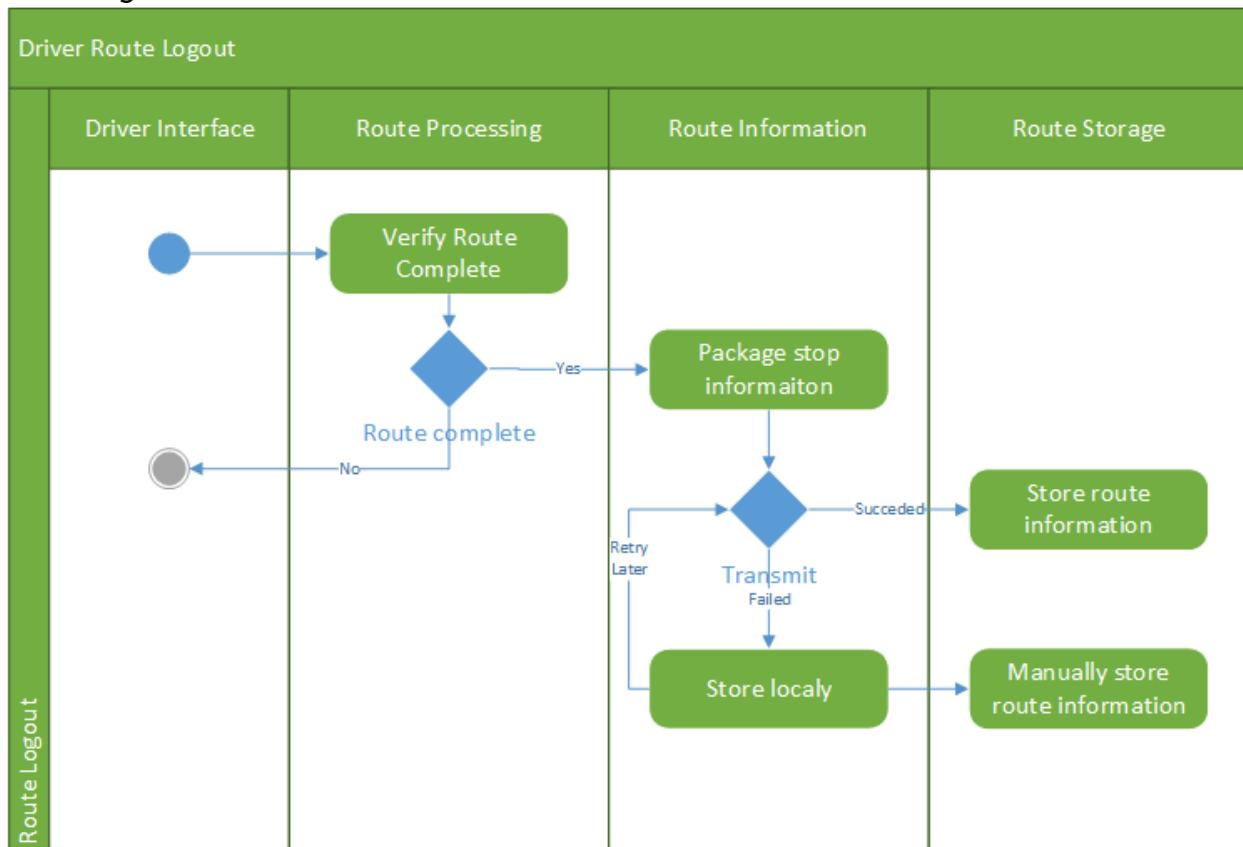
## Refueling Process



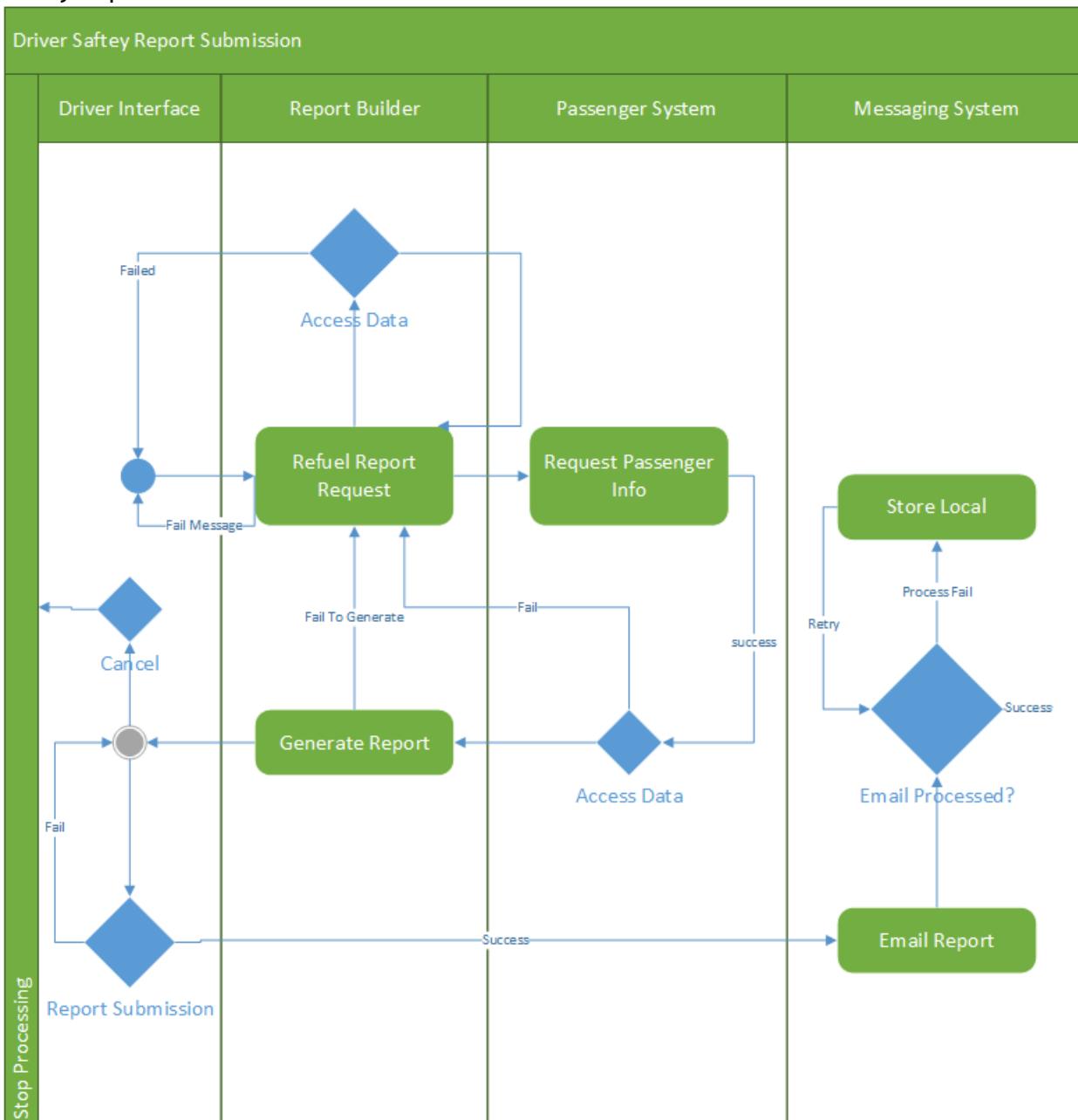
## Route Log-In



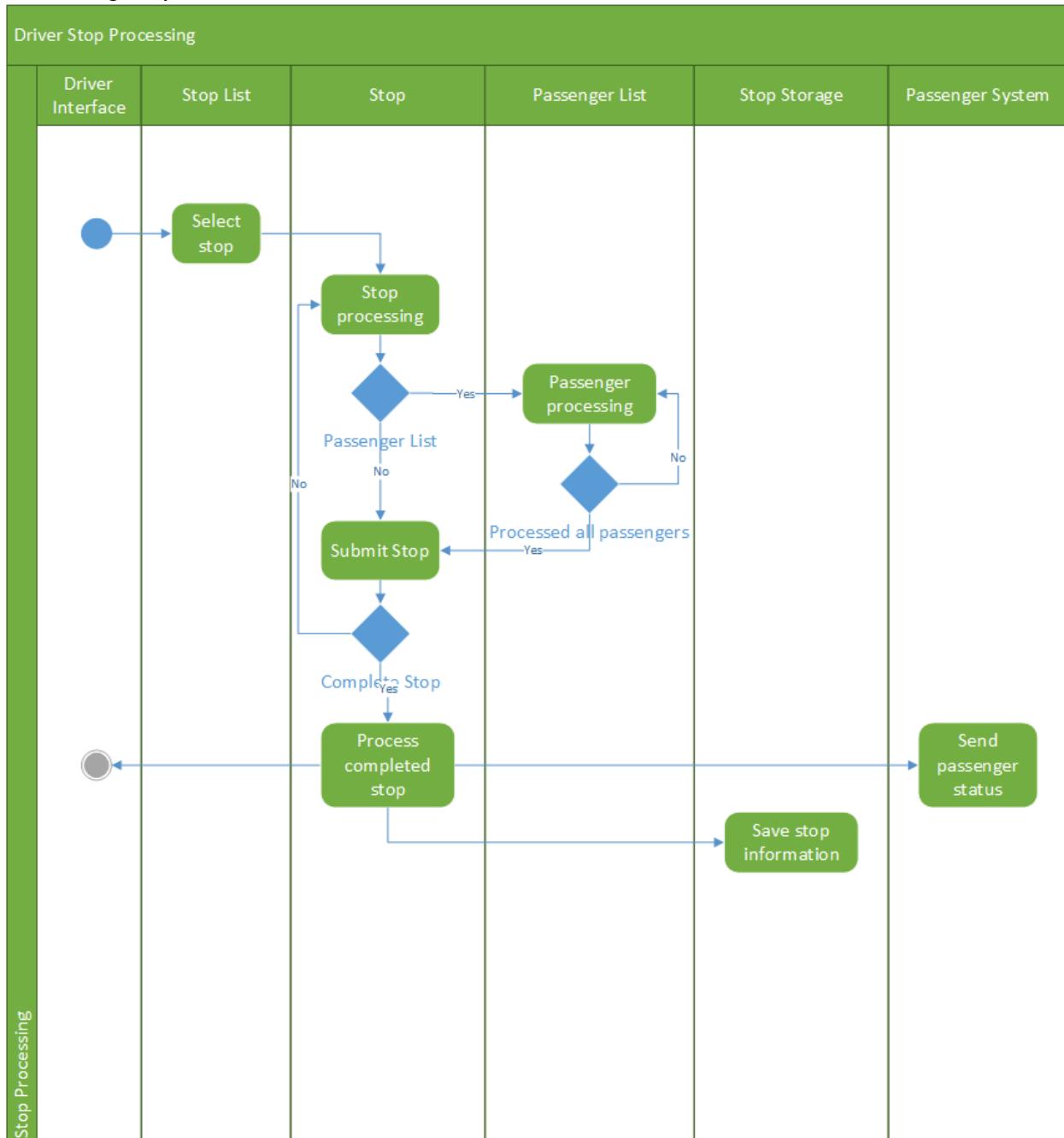
## Route Log-Out



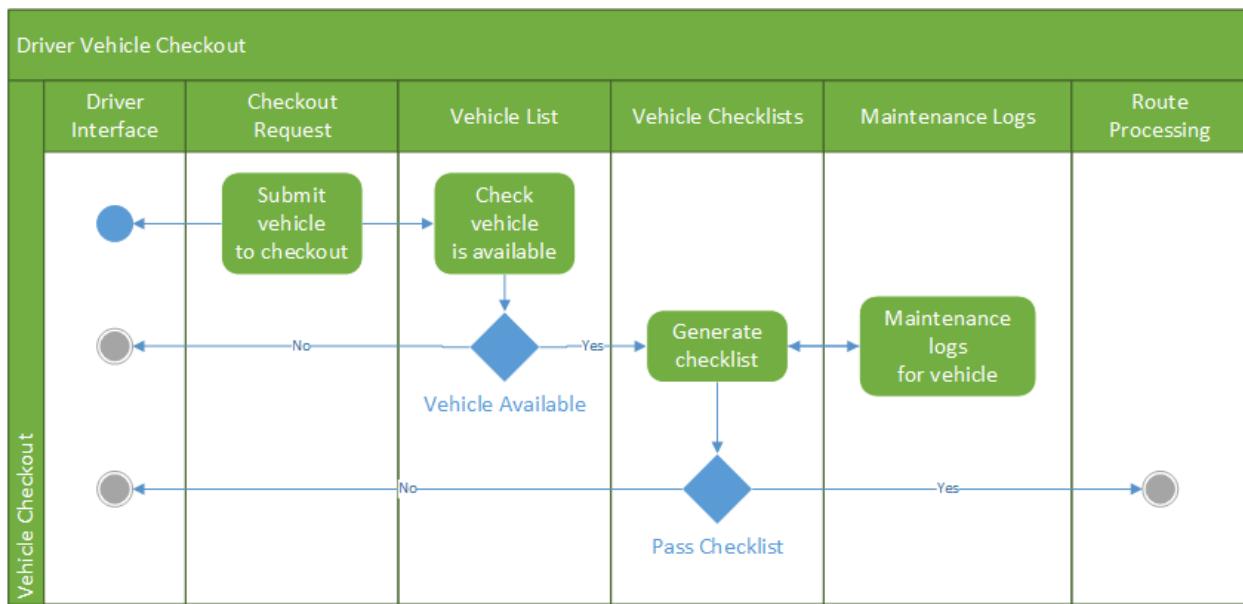
## Safety Reports



## Processing Stops

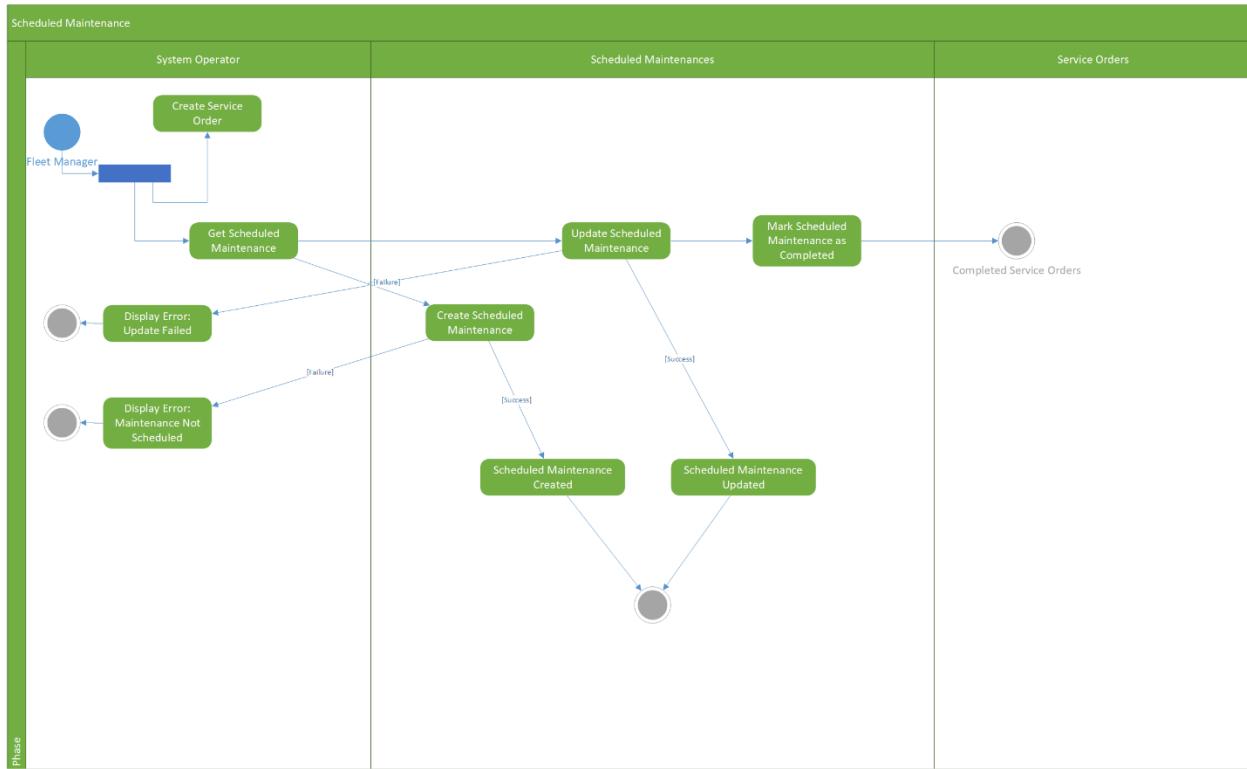


## Driver-Vehicle Check-Out



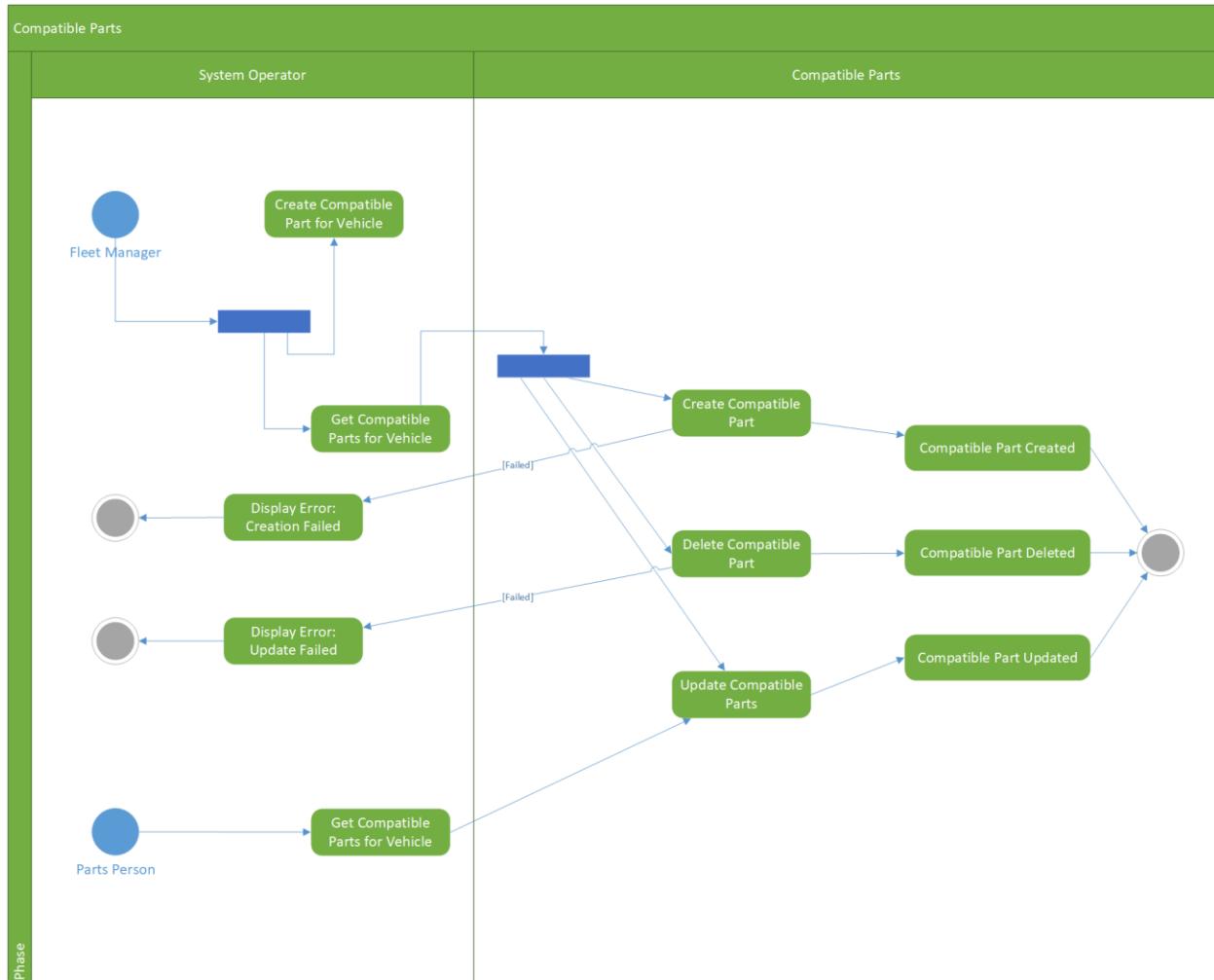
# Maintenance Subsystem

## Scheduled Maintenance

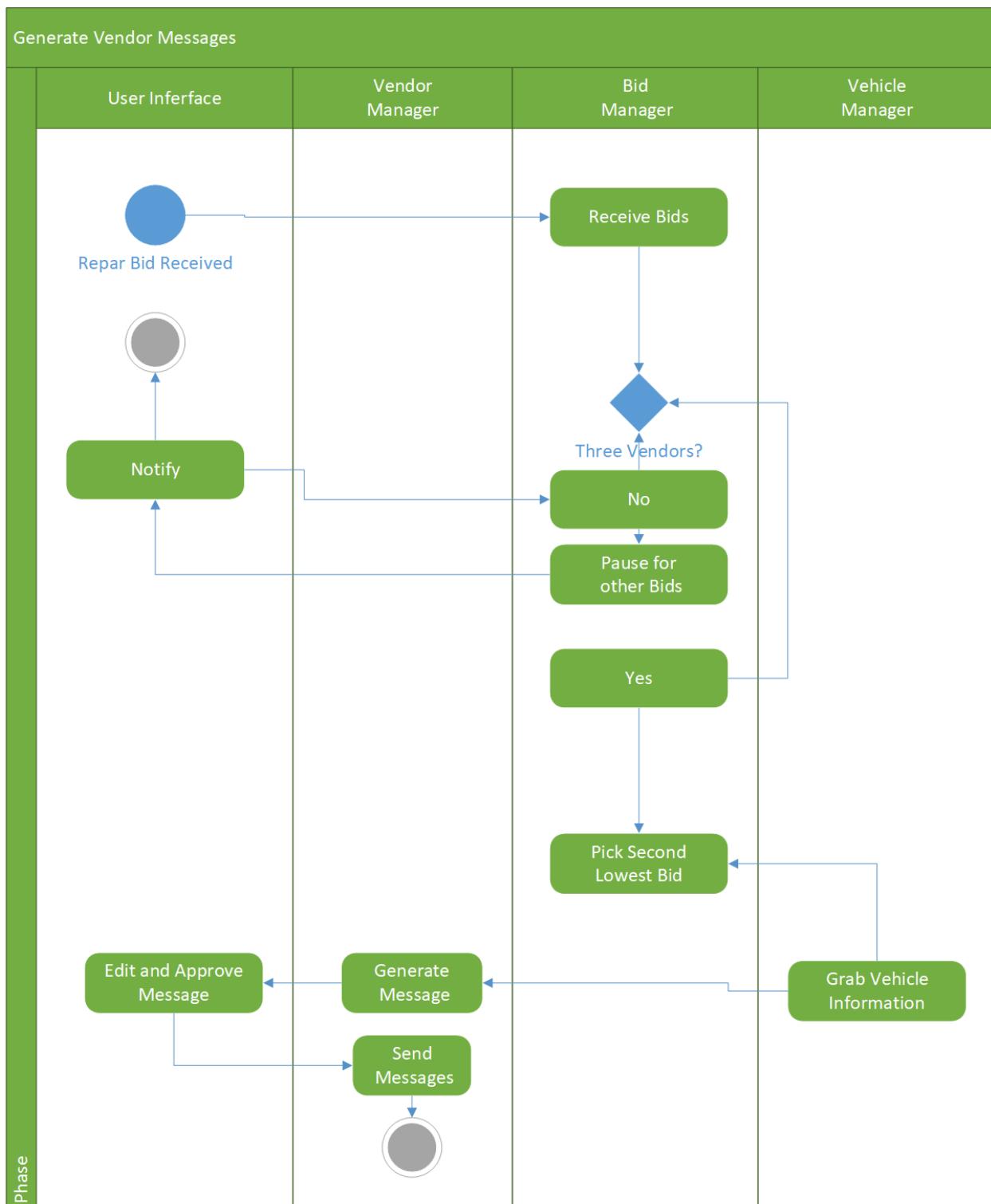


# Parts Subsystem

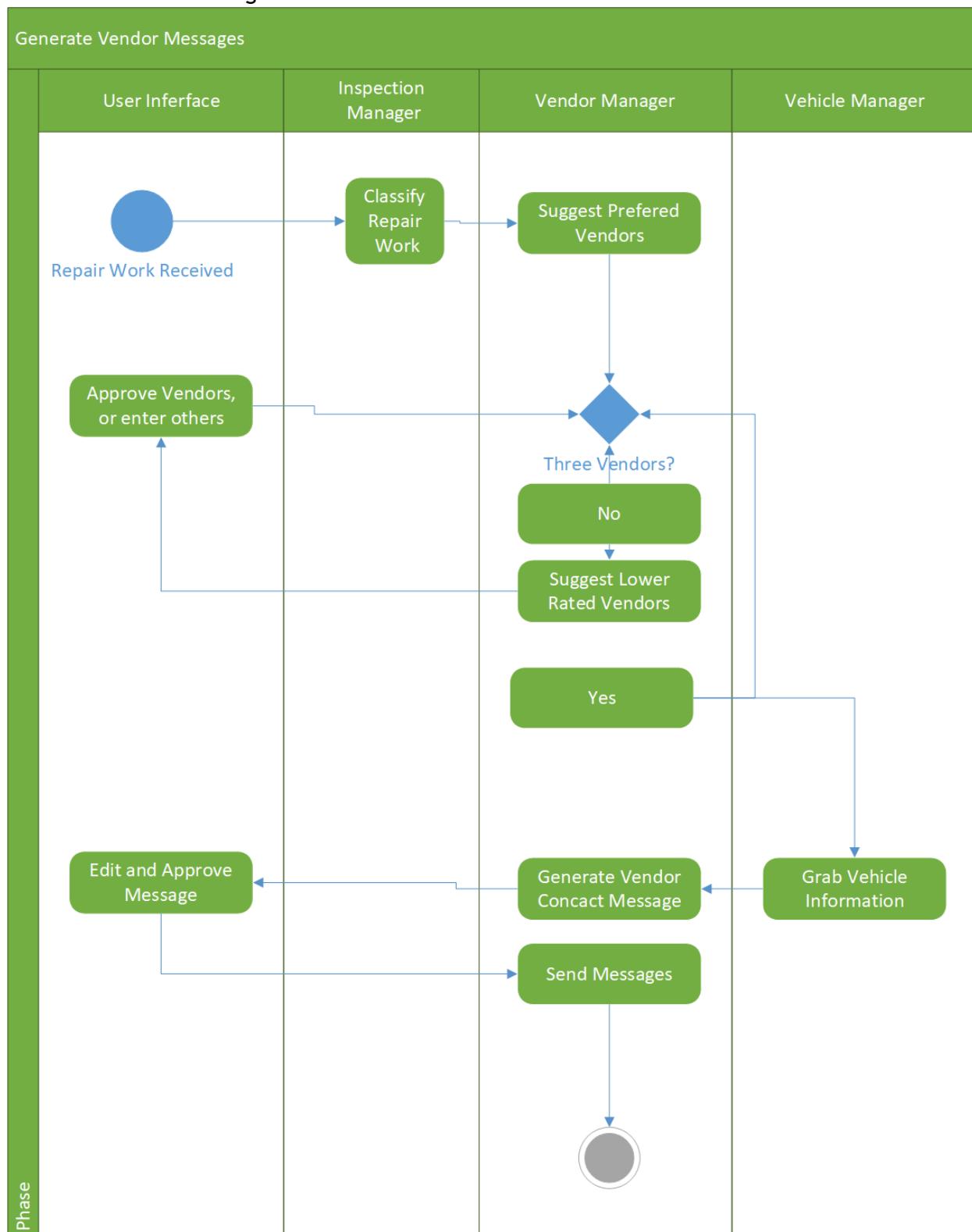
## Compatible Parts



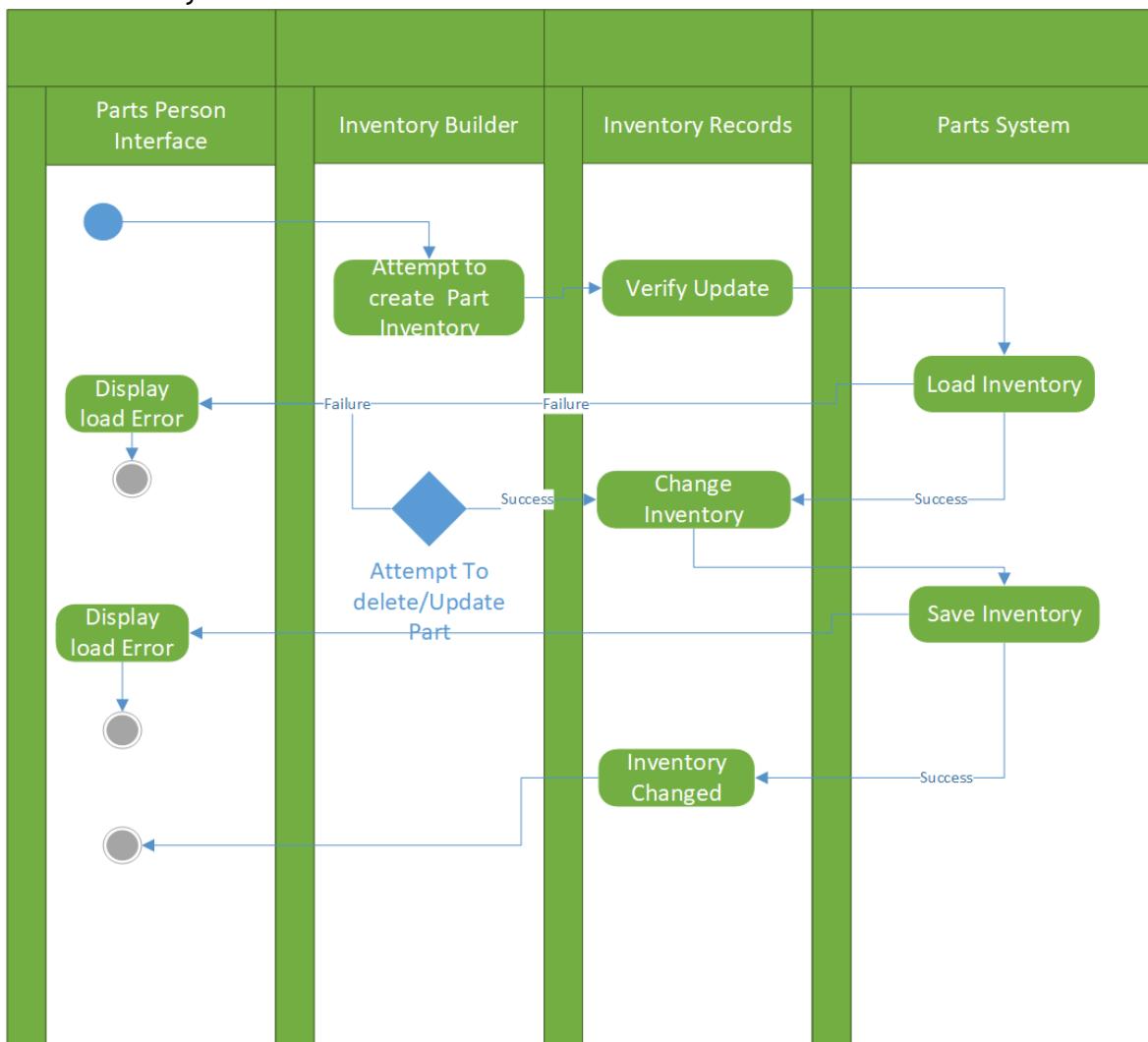
## Generate Vendor Bids



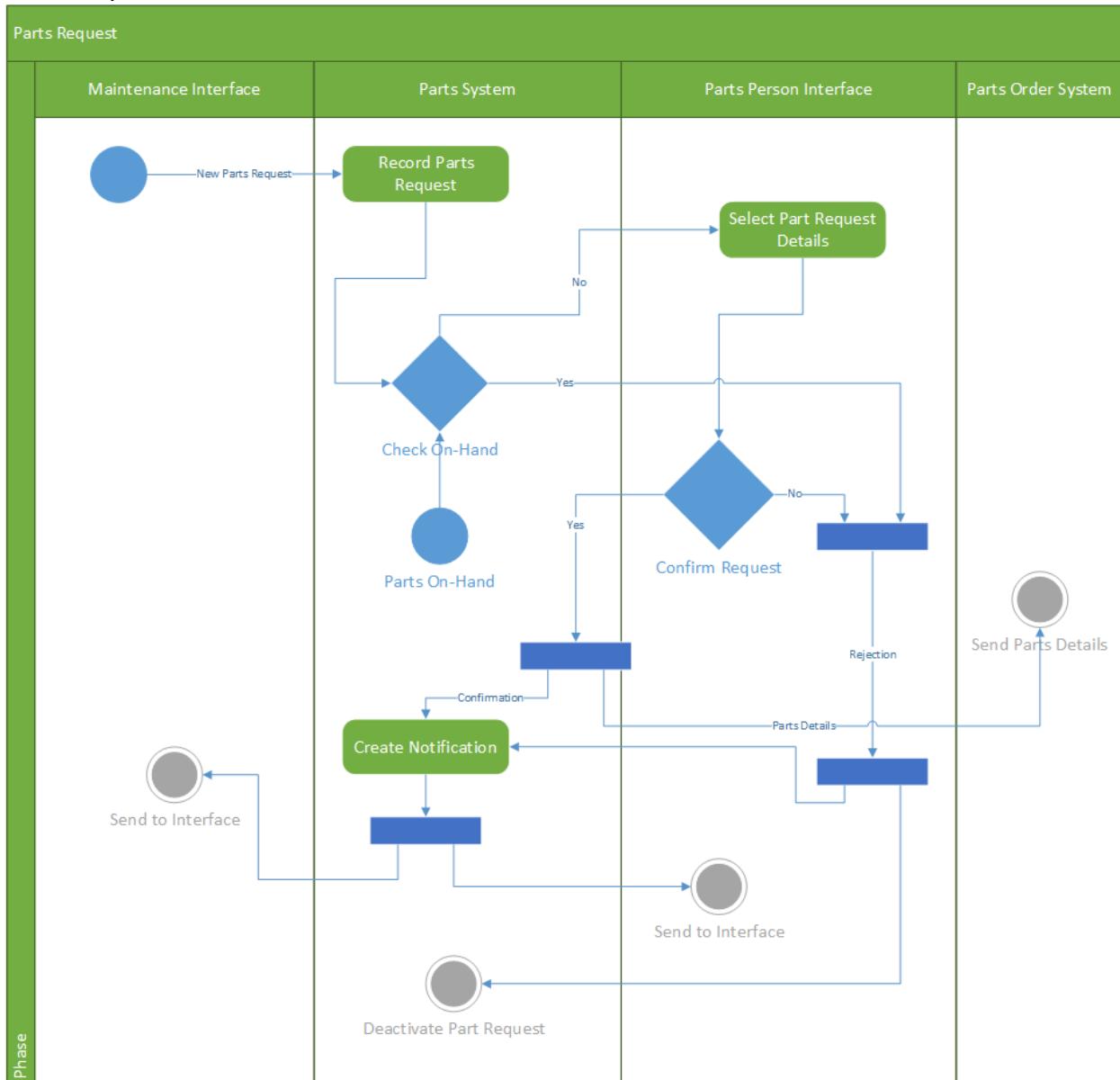
## Generate Vendor Messages



## Parts Inventory

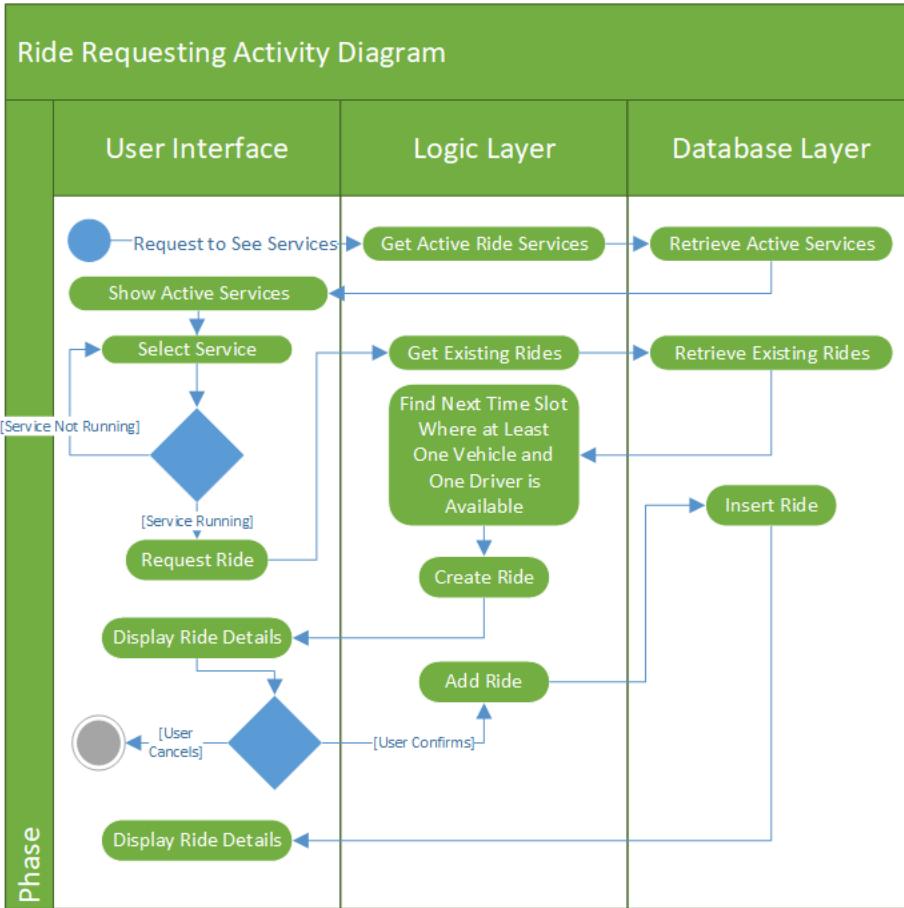


## Parts Request

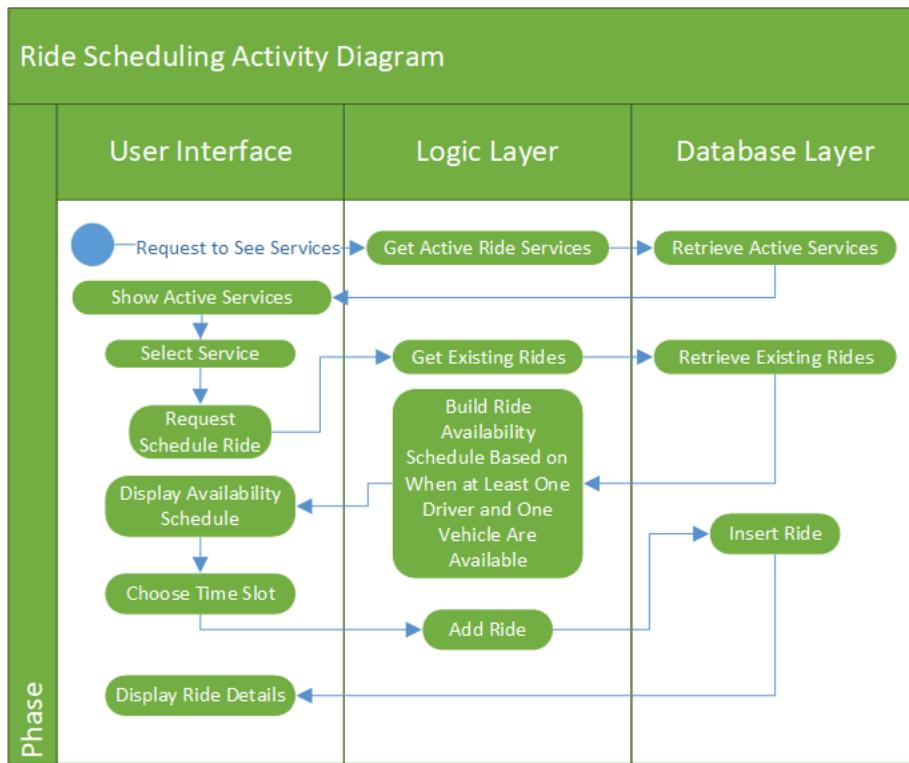


# Ride Subsystem

## Ride Requests

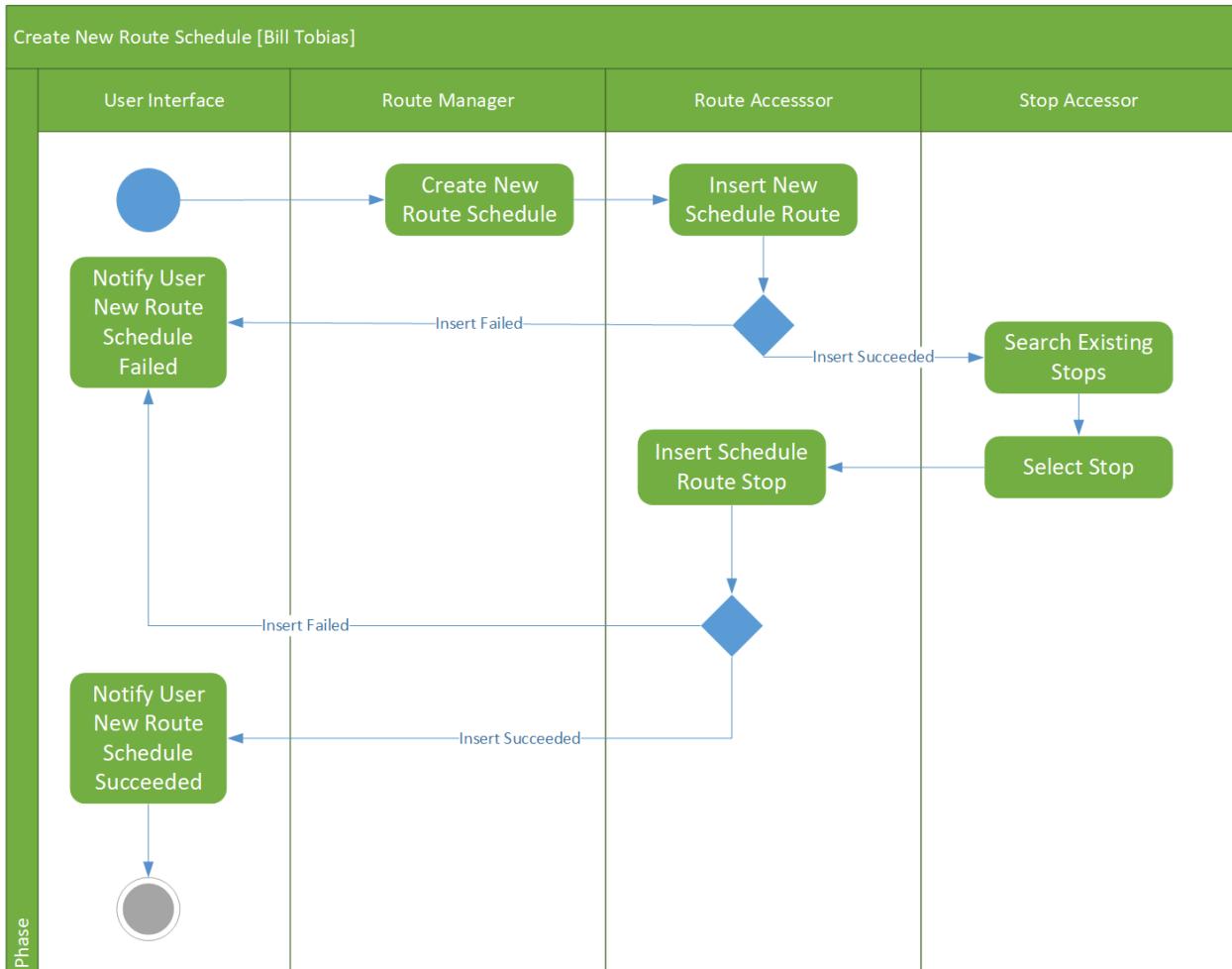


## Ride Schedules



# Route Subsystem

## Route Schedule

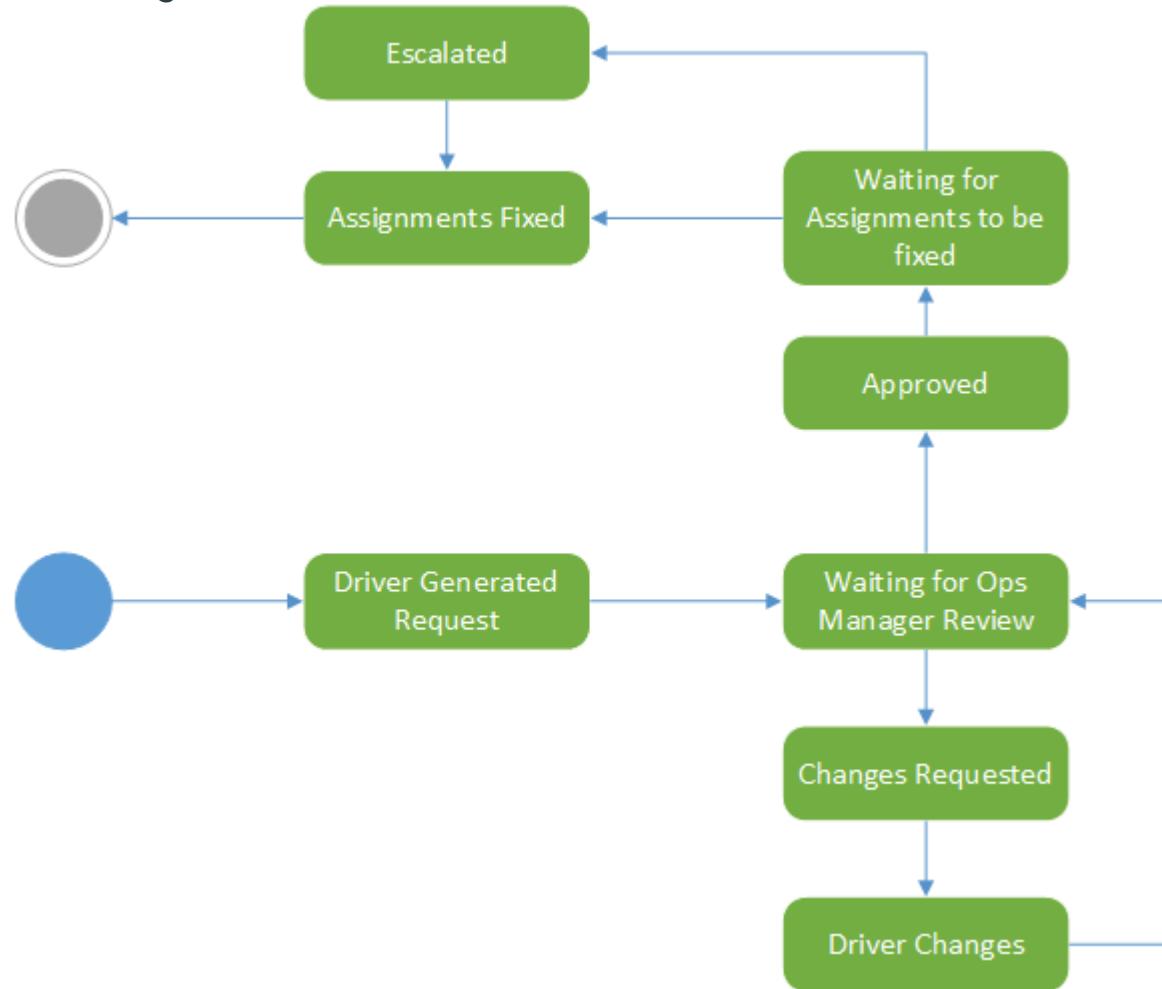


## State-Chart Diagrams by Subsystem

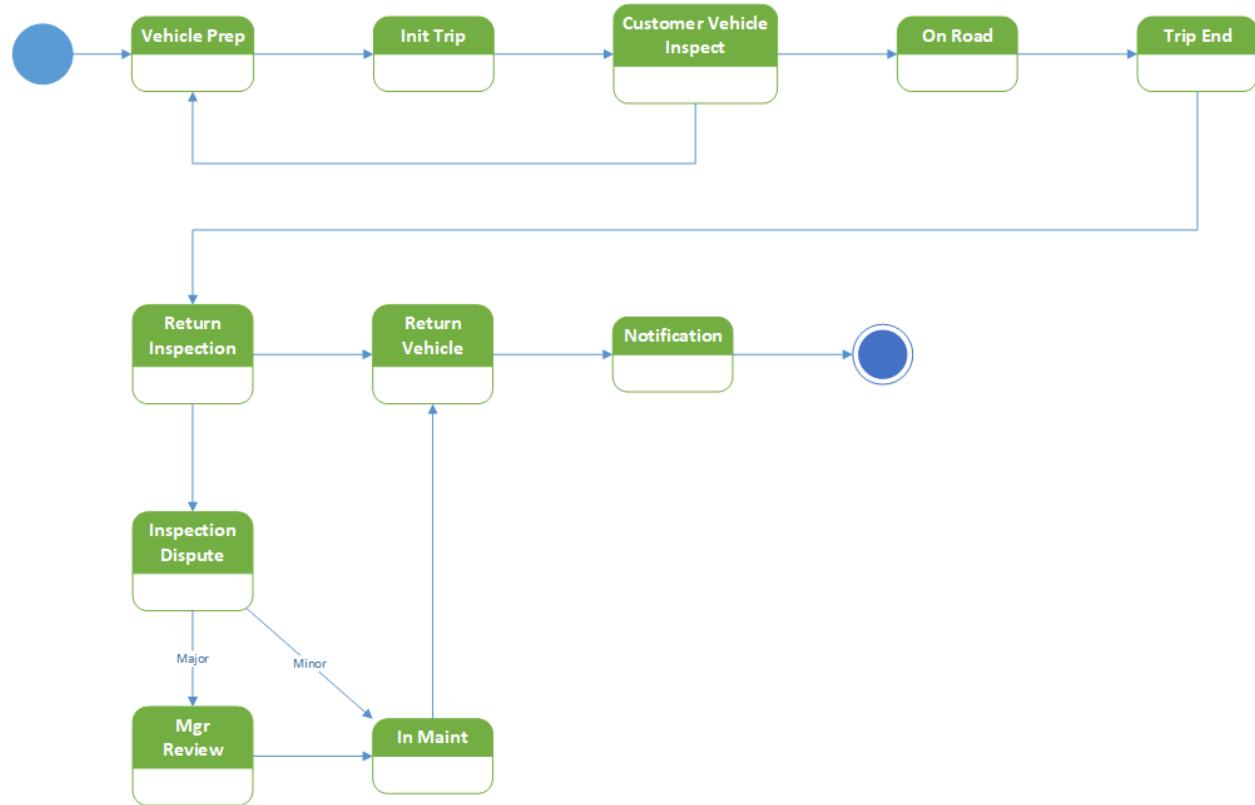
The Activity Diagrams describe the logical flows of functionality between the user interface, the application, and the database. Unlike the Data Flow diagram, the Activity Diagram describes in detail the branches of functionality and the conditions under which branches are taken in certain points during an application process.

## Schedule And Driver Subsystem

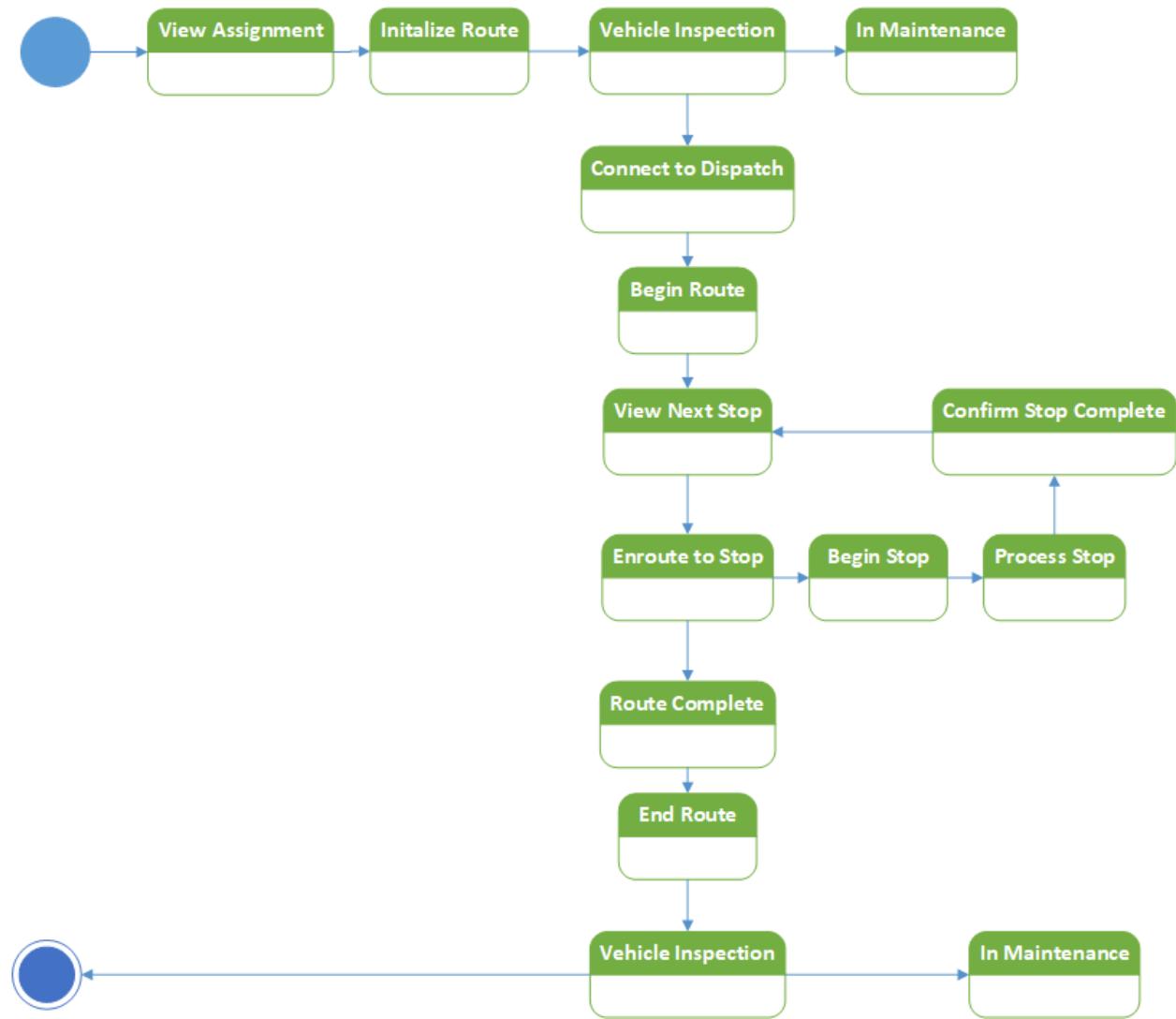
### Scheduling



## Charter Customer Trip

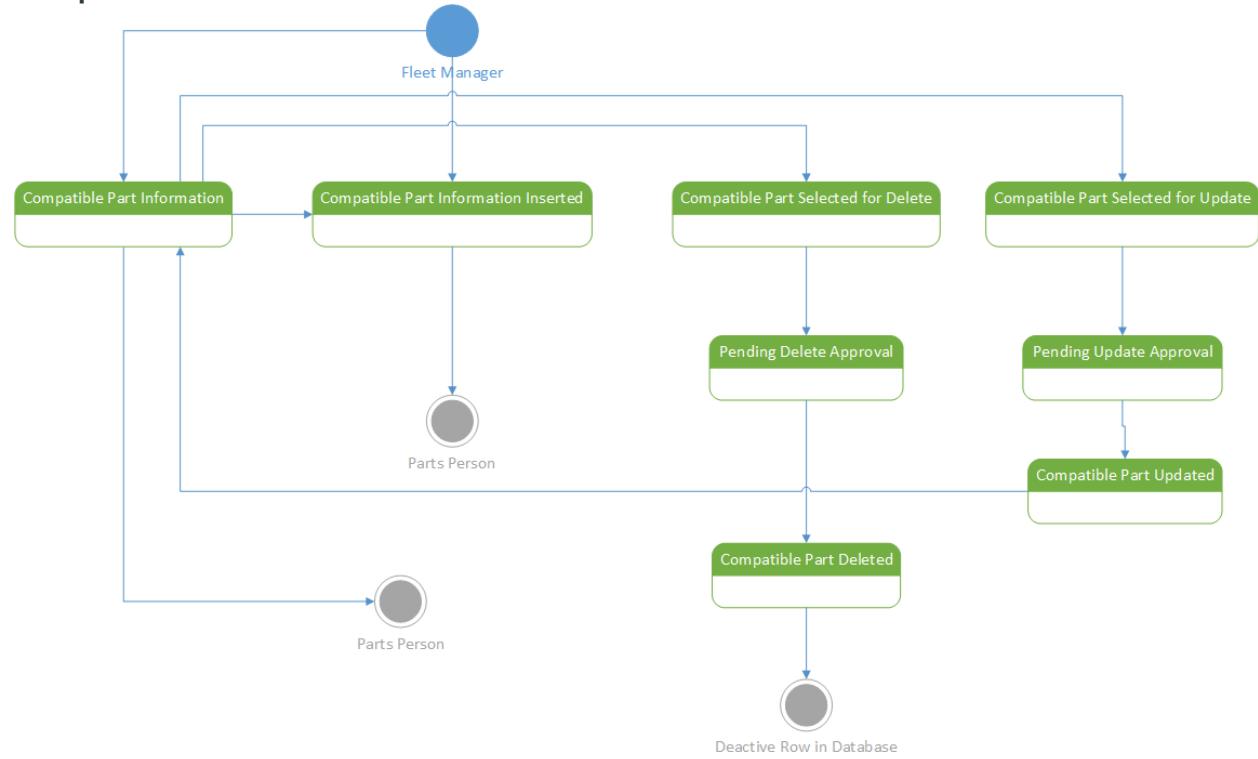


## Driver

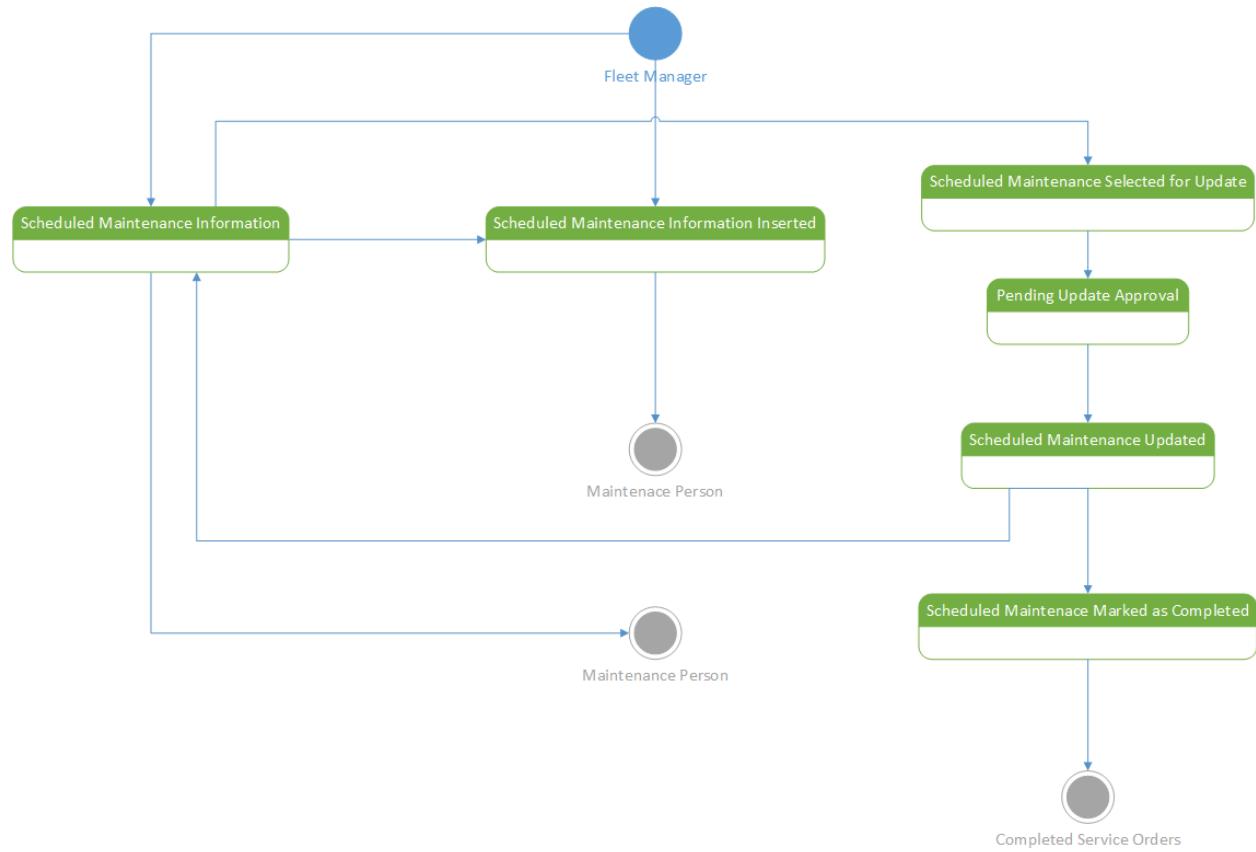


# Maintenance and Mechanic Subsystem

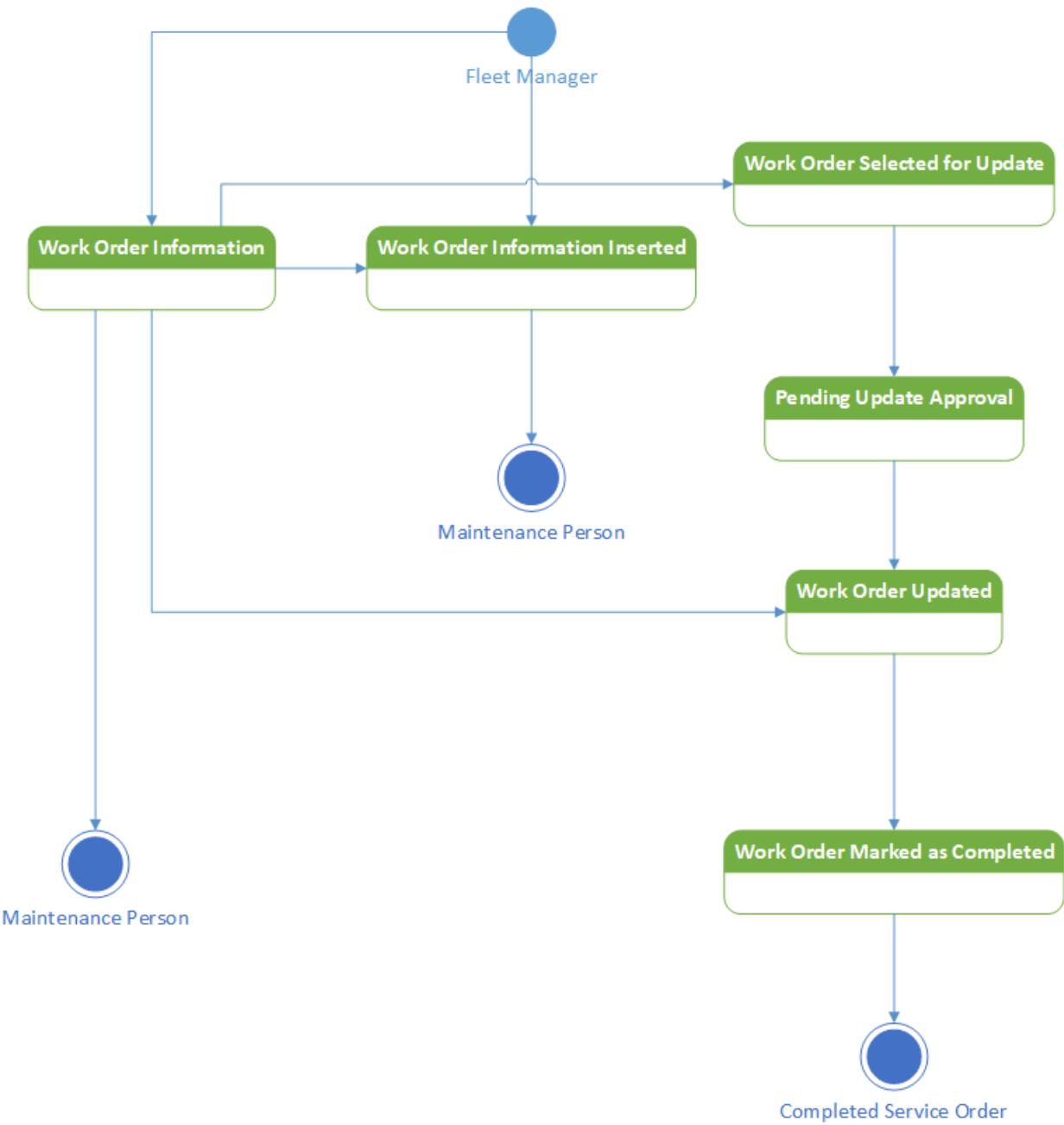
## Compatible Parts

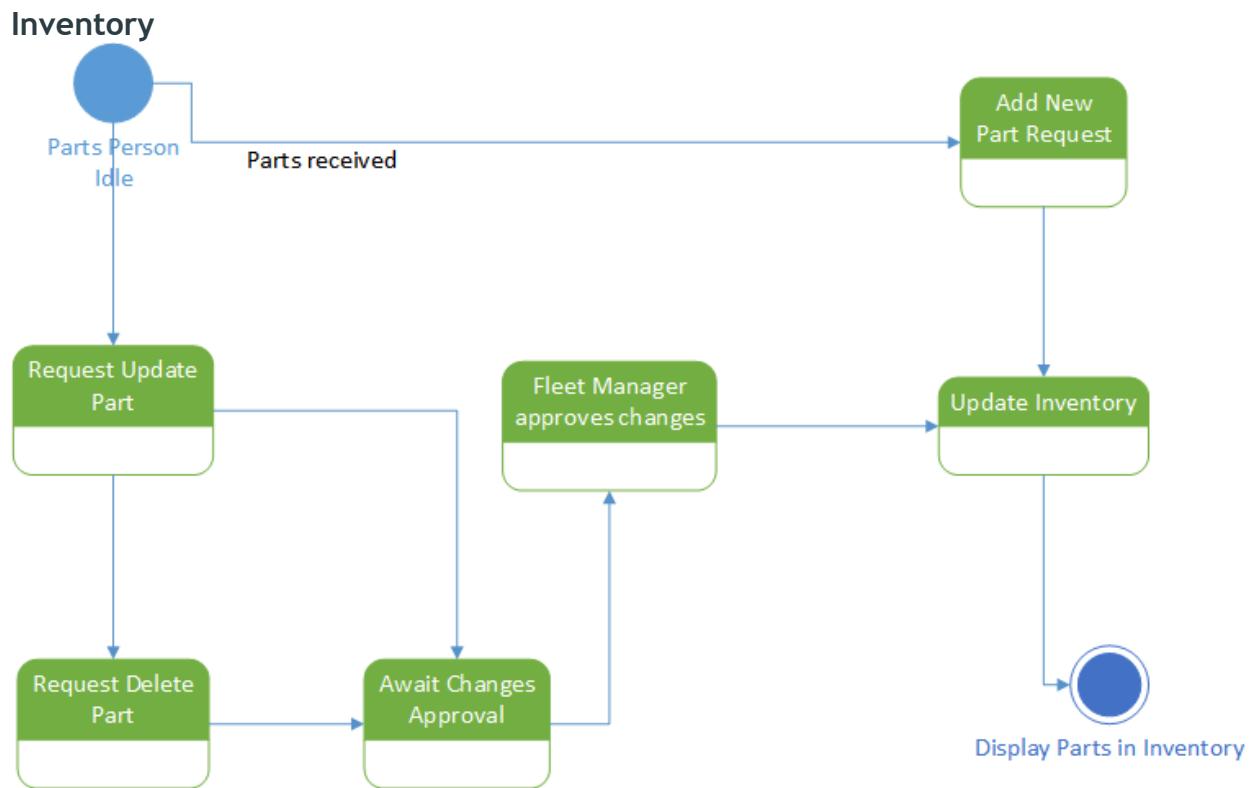


## Scheduled Maintenance

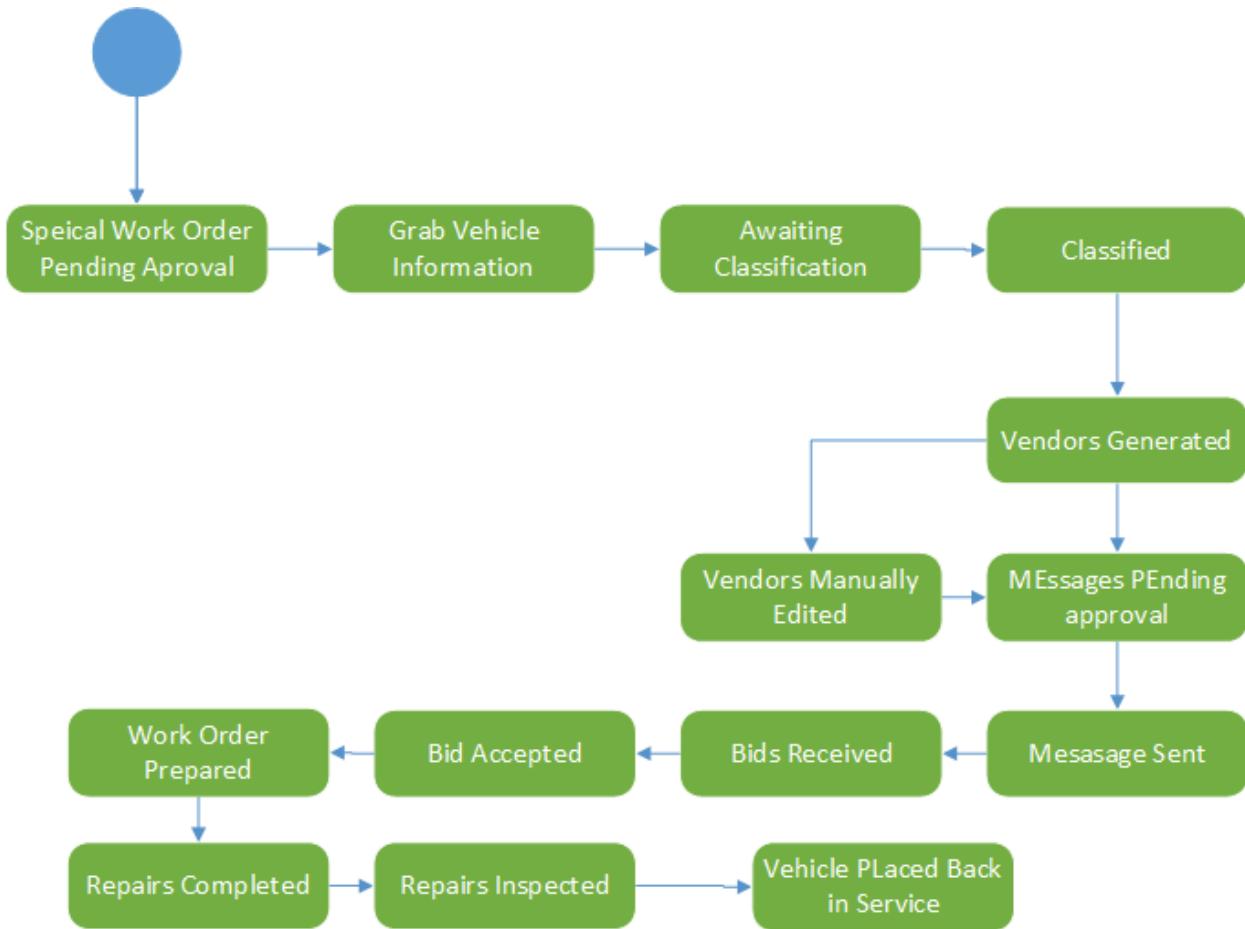


## Work Orders





## Special Work Orders



# Project Details

## Timeline

The project can be expected within a timeframe of between 14-16 weeks, divided into 4 agile sprints

## Technology and Tools

The technology stack for this project includes ASP.NET for the web frontend, WPF for the desktop frontend, ADO.NET as the database driver, Microsoft SQL Server as the database and C# as the programming language.

## Testing and QA

*Minimum for Customer release:*

To ensure a high-quality piece of software is delivered, various testing methodologies will be used. This includes validating logic layer functionality with unit testing, integration testing for component compatibility, an adherence to code integrity testing, and usability testing for a guaranteed user-friendly interface.

*Developer Cycle Integrity:*

The development process will focus on avoiding redundancy, encourage testable and maintainable code with interfaces, adherence to naming conventions by all developers, and clear and concise commented code.

*Maintains Dev's Relationship with Customer:*

For an optimal user experience, the software will be designed to display information that is relevant, labels and controls will be clearly labeled, UI controls will have a clear purpose and will be easy to use, and robust documentation will be provided.

## Project Management

The Agile software development methodology will be followed by the team. GitHub will be the employed version control system. To ensure communication, comprehensive documentation will be maintained throughout the project lifecycle.